# A New Coding Mode for Hybrid Video Coders Based on Quantized Motion Vectors

Marie Andrée Agostini-Vautard, Marco Cagnazzo, *Senior Member, IEEE,* Marc Antonini, *Member, IEEE,* Guillaume Laroche, and Joël Jung

***Abstract*—The rate allocation tradeoff between motion vectors and transform coefficients has a major importance when it comes to efficient video compression. This paper introduces a new coding mode for an H.264/AVC-like video coder, which improves the management of this resource allocation. The proposed technique can be used within any hybrid video encoder allowing a different coding mode for any macroblock. The key tool of the new mode is the lossy coding of motion vectors, obtained via quantization: while the transformed motion-compensated residual is computed with a high-precision motion vector, the motion vector itself is quantized before being sent to the decoder, in a rate/distortion optimized way. Several problems have to be faced with in order to get an efficient implementation of the coding mode, especially the coding and prediction of the quantized motion vectors, and the selection and encoding of the quantization steps. This new coding mode improves the performance of the hybrid video encoder over several sequences at different resolutions.**

***Index Terms*—Motion vector, quantization, video coding.**

## I. Introduction

An EFFICIENT resource allocation between motion vectors (MVs) and motion-compensated residuals is a key feature in any video coder aiming at good rate-distortion (RD) performance. However, in standard coders [1]–[3], it is only possible to indirectly choose how the bit-rate is shared between MVs and residuals by selecting one among the several available *coding modes* for each macroblock (MB). As a consequence, it has been noted that when a sequence is encoded at low bit-rate, a large share of resources is allocated to MVs. The paper by Laroche *et al.* [4] reports that, for H.264/AVC, at high bit-rates the residual coding amounts to the 70% of the bit budget, and the MVs only require a sheer 17% (the rest of the bit-rate being used for mode selection, headers, and other information); on the other hand, for low-to-medium bit-rates, the situation is reversed: the MVs can demand up to the 40% of the rate, the residual accounts for 20% to 30%, and the rest is mainly used for encoding the mode selection. These statistics suggest that, in the framework of an hybrid coder, there could be room for performance improvement if some new coding mode with less costly motion information is introduced.

This intuition is reinforced by the quantitative study we carried out in our work. In Fig. 1, we report the average macroblock rate and distortion for several coding modes in a H.264/AVC coder (JM v.11.0 KTA 1.4 [5]).[1] There is a significant gap between the low-cost, high-distortion SKIP mode and the relatively higher-cost, low-distortion INTER 16×16 mode (while INTRA and lossless IPCM modes are far more expensive and usually not suitable in low bit-rate context). Therefore, we want to introduce a new coding mode which should have an intermediate behavior between SKIP and INTER. To achieve this target we propose the *lossy coding* of MVs, obtained thanks to scalar quantization.

The quantization of MVs has attracted the attention of the video coding community since the mid 1990s [6]–[9], but the approach proposed in those works mainly amounts to a vector quantization (VQ) of MVs, with quantized vectors used at the encoder side to compute both the motion-compensated residual and prediction, that is to say in closed loop. Lee and Woods [6] simultaneously estimated and vector quantized the motion vectors by reinterpreting the block matching algorithm as a type of vector quantization. Joshi *et al.* [7] used an entropy-constrained quantization. In the paper by Regunathan and Rose [8], a RD-optimized codebook for VQ of MVs is designed, whereas Ribas-Corbera and Neuhoff proposed a model based optimization of the MV precision [10]. In order to reduce the cost of the motion information, Da Silva Cruz and Woods [9] presented an adaptive method based on vector quantization techniques with a dynamically updated codebook. Xiong *et al.* [11] used a motion layer decision algorithm to make the best tradeoff between motion and texture. Some approaches [12] use a precision limited coding for scalable motion vectors (vectors estimated at the highest resolution and then scaled

Manuscript received October 16, 2009; revised May 7, 2010 and December 9, 2010; accepted February 4, 2011. Date of publication March 28, 2011; date of current version July 7, 2011. This work was carried out in the framework of an "external research contract" supported by Orange Labs, Issy Les Moulineaux, France. This paper was recommended by Associate Editor S. Li.

M. A. Agostini-Vautard is with Nony, Patent and Trademark Attorneys, Paris 75008, France (e-mail: agostini.ma@gmail.com).

M. Cagnazzo is with TELECOM ParisTech, Paris 75634, France (e-mail: cagnazzo@telecom-paristech.fr).

M. Antonini is with the I3S Laboratory, CNRS, University of Nice-Sophia Antipolis, Sophia Antipolis 06903, France (e-mail: am@i3s.unice.fr).

G. Laroche is with the Canon Research and Development Center, Cesson-Sevigne 35517, France (e-mail: larocheguillaume@wanadoo.fr).

J. Jung is with Orange Labs, Issy Les Moulineaux 92130, France (e-mail: joelb.jung@orange-ftgroup.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2133590

[1]These operation points have been obtained on the sequence *City*; for other sequences similar results have been obtained.

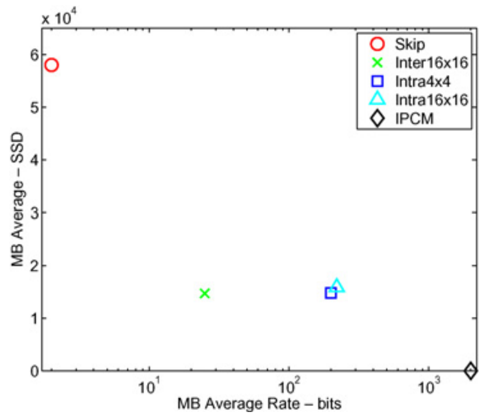


Fig. 1. Average operation points of H.264/AVC modes, sequence *City*.

down at the decoder). Barbarien *et al.* used [13] a quality-scalable motion vector coding algorithm using median-based motion vector prediction, and a heuristic technique for global rate-allocation. More recently, Wu and Woods [14] proposed an alphabet general partition of motion vectors to achieve accuracy or quality scalability of these vectors. A selective layered structure is used to reduce the number of motion vectors transmitted when appropriate. In a wavelet-based 2-D+t video coder, Tagliasacchi *et al.* [15] encoded the nodes of the quad-tree, resulting from the variable size block matching, from top to bottom starting from the most significant bit-plane. In the paper by Agostini *et al.* [16], an approach of quantization of MVs is used, in the framework of wavelet-based video coding, and the authors focus on a model-based analytical evaluation of optimal tradeoff between MV rate and coefficient rate.

Our technique is not based on VQ and provides a data-driven solution rather than a model-based one. The main difference between the proposed solution and those that can be found in the scientific literature is that we propose to use two different motion vectors for computing the residual and the prediction. The residual is sent (after transformation and quantization) to the decoder, so it can be computed using the most accurate vector. On the contrary, the prediction is obtained by using the encoded motion vector: this costs some bit-rate, but we can reduce this cost if the motion information is quantized. Of course, this can also increase the distortion. For this reason we consider a rate-distortion optimized framework: the encoder computes the reconstructed block as the decoder would do, and decides whether using a quantized motion vector after evaluating an RD cost function. In any case the encoder stores the decoded block and uses it as a reference for next frames encoding, thus assuring that no drift is introduced between encoder and decoder. This approach is novel in the literature (at the authors' best knowledge) and constitutes the main contribution of this paper. Unlike previous approaches [16], we refer to a H.264/AVC-like framework, where the proposed technique is implemented as a new coding mode in competition with the existing ones. Of course, the introduction of new coding modes does not assure an improvement of the coding efficiency. Indeed, if on the one hand it increases the signalling cost (i.e., the coding cost of *any* selected mode) for all the MBs, on the other hand, the new mode could achieve better performance than classical ones for some other MBs. One of the targets of this paper is to verify that the gains associated with the new mode surpass the losses. Of course, the proposed technique aims at improving the performance of an hybrid video encoder above all at low bit-rates, where the coding cost of MVs can be too high. Of course, many techniques for low bit-rate video coding exist, like the decoder-side motion vector derivation [17], or the motion vector competition described in [4], which has been introduced in many of the proposals for the new ISO/ITU video coding standard, as the one of France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor [18]. However, all these techniques are not alternative to the one we propose here, and on the contrary they can be used together with ours.

As the main new tool of this mode is the quantization of the MV, we call it quantized motion vector mode (QMV). Note that the term "mode" is borrowed from the H.264/AVC terminology, since the proposed technique naturally can be described as it was an additional mode for H.264/AVC; for this reason we use (with some abuse) the term mode in order to refer to the proposed technique. However, we emphasize that on one hand the proposed technique is not a part of the standard, and on the other that it can be used in any hybrid video coder. We implemented it into the H.264 JM/KTA, in order to take advantage from its excellent motion model and to compare the RD performance of the proposed technique with that of the standard modes.

The introduction of the QMV techniques requires to deal with several relevant problems, such as the encoding of quantized MVs, and the selection and encoding of the quantization step. One important issue is also the fact that the predictive coding of the MVs could not be a convenient strategy anymore. We describe these issues in this paper, as follows: the new coding mode is presented in Section II; Section III introduces the crucial question of the selection and encoding of the motion quantization steps; the quantization of the MVs and the resulting issues are described in Section IV; experimental setup and results are reported in Section V, while Section VI draws conclusions and ends this paper.

## II. New Coding Mode

In this section, we introduce the new mode in the framework of RD optimized video coding. This mode have been inserted into the JM-KTA [5] implementation of H.264/AVC, but it could be inserted in any hybrid coder allowing different modes. The encoder will compare the new mode to the standard modes to find the best one, i.e., the one optimizing the RD performance. This target is achieved by evaluating the Lagrangian cost function $J_i$ associated with the $i$th mode: $J_i = D_i + \lambda_{\text{mode}} R_i$. The basic idea is that we can tune the rate allocation by choosing among the non-QMV modes or a QMV mode with a suitable quantization step for MVs.

### A. General Description

The new coding mode is summarized in Fig. 2, where we describe the encoder and the decoder. The decoder performs a

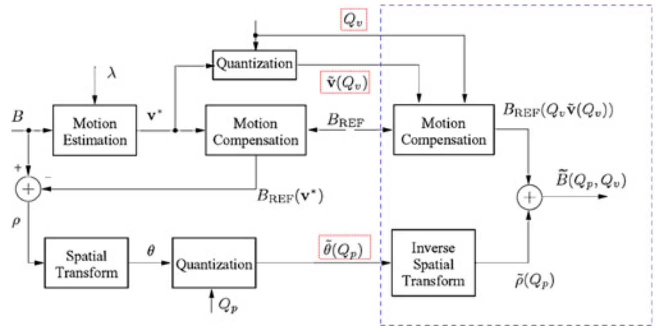


Fig. 2. New coding mode: quantization of motion vectors. In red: the information that the encoder sends to the decoder. The dashed box represents the decoder. The reference information $B_{\mathrm{REF}}$ is known both by the encoder and the decoder.

subset of the operations of the encoder, and it is highlighted by the blue dashed box. The quantities computed at the encoder and sent to the decoder are highlighted by a red dotted box.

The new coding mode works as follows: first, an accurate (i.e., non-quantized) motion vector $\mathbf{v}^*$ is computed by classical motion estimation (ME), and is used in order to compute the motion-compensated residual $\rho$, for each MB $B$. Such residual is then transformed, quantized by $Q_p$ and sent to the decoder (after lossless coding), like in all hybrid coders. The difference between this mode and the standard INTERmode is that the MV $\mathbf{v}^*$ is quantized by a simple uniform scalar quantization of its components with a step $Q_v$. We differ to Section III the problem of efficiently selecting and encoding $Q_v$. Knowing the quantization index $\tilde{\mathbf{v}}$ and the quantization step $Q_v$, the encoder can use them to perform motion compensation on the reference image, which is the same available at the decoder. Adding the residual to the motion-compensated prediction, the decoded block can be computed, exactly as the decoder would do. The encoder can therefore compute the associated distortion, and then the rate, by losslessly encoding $Q_v$, $\tilde{\mathbf{v}}$ and $\tilde{\theta}$. Hence, the encoder can take an RD-optimized decision about using the new mode or not. Further details about the RD cost computation are given in the next section.

Fig. 2 allows us to promptly catch the difference between the standard INTER mode and the new QMV: in the former, the same motion vector $\mathbf{v}^*$ is used to compute the motion-compensated prediction $B_{\mathrm{REF}}(\mathbf{v}^*)$ and the residual $\tilde{\rho}(Q_p)$. In the latter, the motion vector is quantized before being sent, but the encoder and the decoder still use the same prediction (computed with the quantized vector) and the same residual (computed with the original one $\mathbf{v}^*$). Finally, when the new mode is actually used to encode the current macroblock, the reference frame is updated accordingly. Therefore, there is no drift between the encoder and the decoder. In this sense, the proposed scheme is a closed-loop prediction.

### B. Rate-Distortion Optimization in Hybrid Video Coding

In a generic hybrid coder each MB can be encoded in different ways, called modes. Even if standards do not impose any constraint about which mode should be chosen for a MB, in order to achieve efficient RD performance, any decision (the choice of a mode, of the MV or of the spatial transform's size) should ideally be made according to its final effect on the decoded video. The goal of the encoder is to minimize the distortion $D$ subject to a constraint on the rate $R$ (rate-distortion optimization, RDO [19]). Let $Q_p$ be the residual quantization step; the cost function for the mode $i$ is

$$J_i(Q_p, \lambda_{\mathrm{mode}}) = \left[D_i(Q_p) + \lambda_{\mathrm{mode}} R_i(Q_p)\right]. \qquad (1)$$

The RDO problem is solved by computing the cost function for each mode $i$, and then selecting the mode with the minimal cost. This choice depends on the quantization step and on the Lagrangian parameter $\lambda_{\mathrm{mode}}$, which, in turn, is usually chosen as a function of $Q_p$. In H.264/AVC one can choose [20]

$$\lambda_{\mathrm{mode}} = 0.85 \cdot 2^{Q_p/3-4}. \qquad (2)$$

Similar formulas exist for other encoders. For MPEG-2 it has be shown that $\lambda_{\mathrm{mode}} = 0.85 Q_p^2$ [19]. In order to solve the RDO problem, we must compute the distortion, defined as a function of the error between the original MB $B$ and its reconstructed version $\widetilde{B}$; to this end, the SAD or the SSD are usually chosen. In particular, if we represent with $\mathbf{p} \in \mathbb{Z}^2$ the pixel location and with $\mathcal{S}$ the support of the macroblock, the distortion can be expressed as

$$D = \left\|B - \widetilde{B}\right\|_\ell = \sum_{\mathbf{p}\in\mathcal{S}} |B(\mathbf{p}) - \widetilde{B}(\mathbf{p})|^\ell \qquad (3)$$

where we indicate with $B(\mathbf{p})$ the luminance value of the block $B$ in the position $\mathbf{p}$. The parameter $\ell$ is equal to 1 for the SAD or to 2 for the SSD.[2] As far as the coding rate is concerned, for the non-motion compensated modes, it amounts to the cost of transmitting the quantized transform coefficients plus the signalling of the selected mode. For the motion-compensated modes, we should add the motion information cost.

For the INTRA mode, if we call $P$ the prediction of the current MB,[3] $\rho$ the residual, and $T[\cdot]$ the transform operator, we have

$$\begin{aligned} \rho &= B - P & \theta &= T[\rho] \\ \tilde{\theta}(Q_p) &= \mathcal{Q}(\theta, Q_p) & \widetilde{B}(Q_p) &= \left[T^{-1}\left(\mathcal{Q}^*(\tilde{\theta}, Q_p)\right)\right] + P \end{aligned}$$

where we introduced the transform coefficients $\theta$, the corresponding quantization index $\tilde{\theta}$, and the reconstructed macroblock $\widetilde{B}$. Moreover, we introduced the quantization function $\mathcal{Q}$ (associating a quantization index to the input, given a quantization step) and the function $\mathcal{Q}^*$ (returning the quantization level associated with an index $z$ and to a quantization step). Often a uniform quantization is used, such that

$$\mathcal{Q}(x, Q_p) = \mathrm{round}\left(\frac{x}{Q_p}\right)$$
$$\mathcal{Q}^*(z, Q_p) = Q_p z$$

[2]Note that (3) contains a little abuse of notation, since we should have written $\left\|B - \widetilde{B}\right\|_\ell^\ell$. However, we will keep the simplified notation for the sake of readability.

[3]We remark that in H.264/AVC there exist two spatial transforms and several spatial prediction techniques for each of them, so there is a considerable number of INTRA modes. Moreover, $P$ is always computed with data available to the decoder as well. Finally, for encoders that do not perform spatial prediction (as MPEG-2), we can assume $P = 0$.

where the function round is the rounding toward the nearest integer. However, in more recent codecs a deadzone quantization, with a central cell larger than others, is used. When the central cell is twice as large as the others, the $\mathcal{Q}(\cdot,\cdot)$ and $\mathcal{Q}^*(\cdot,\cdot)$ function are described by

$$\begin{aligned}\mathcal{Q}(x, Q_p) &= \operatorname{fix}\left(\frac{x}{Q_p}\right)\\ \mathcal{Q}^*(z, Q_p) &= Q_p(z+\frac{1}{2}\operatorname{sign}(z))\end{aligned} \quad (4)$$

where the function $\operatorname{fix}(\cdot)$ is the rounding to the nearest integer toward zero, and $\operatorname{sign}(\cdot)$ is the usual *signum* function: $\operatorname{sign}(x) = -1$ if $x < 0$, $\operatorname{sign}(x) = 0$ if $x = 0$, and $\operatorname{sign}(x) = 1$ if $x > 0$. H.264/AVC employs dead-zone quantization as well, but the formulas are different from those reported here, in order to address some specific issue. In particular in H.264 quantization can be performed by only using bit-shifts, sums and subtractions, and the actual quantization process depends on the MB mode (INTER/INTRA) and on the type of data (luma/chroma). We refer the interested reader to the paper by Malvar *et al.* [21] or to the ITU-T recommendation [22] for further details.

Now we can compute the distortion and the rate for any INTRA mode

$$\begin{aligned}D(Q_p) &= \left\|B - \widetilde{B}(Q_p)\right\|_\ell\\ &= \left\|\rho - T^{-1}\left(\mathcal{Q}^*(\tilde{\theta}, Q_p)\right)\right\|_\ell\\ R(Q_p) &= R(\tilde{\theta}(Q_p)) + R_{\text{mode}}\\ J_{\text{INTRA}}(Q_p, \lambda_{\text{mode}}) &= D(Q_p) + \lambda_{\text{mode}} R(Q_p).\end{aligned}$$

We note that the $R_{\text{mode}}$ term accounts for the signalling of the chosen transform and prediction scheme.

The computation of the cost function for the motion compensated modes is more complex than the one for the INTRA mode, since it depends on the MV as well. Therefore, let $B_{\text{REF}}(\mathbf{v})$ be the motion compensated prediction of the MB, computed using the motion vector $\mathbf{v}$.

For each candidate motion vector $\mathbf{v}$ in the search set $V$, we should compute the resulting decoded block, through the motion compensated residual, its quantized transform and finally the reconstructed residual. The reconstructed residual is used by the decoder to produce the reconstructed image, by adding it to the motion compensated prediction. In conclusion, we would have a cost function depending on $\mathbf{v}$

$$J_{\text{INTER}}(Q_p, \lambda_{\text{mode}}, \mathbf{v}) = D(Q_p, \mathbf{v}) + \lambda_{\text{mode}} R(Q_p, \mathbf{v}). \quad (5)$$

The ME should performed by searching the argument minimizing (5)

$$\mathbf{v}^*(Q_p, \lambda_{\text{mode}}) = \arg\min_{\mathbf{v}} J_{\text{INTER}}(Q_p, \lambda_{\text{mode}}, \mathbf{v}). \quad (6)$$

This way to compute $\mathbf{v}^*$, even though optimal, is hardly, if ever, used in practice, because it is far too complex [19]: for each *candidate vector* in the search set, the whole coding/decoding process should be carried out in order to find the best possible vector. Since the number of candidate vectors can be very large, this approach is unfeasible, all the more because there exist suboptimal ME techniques which allow us to largely reduce the computational complexity without affecting too much the final performance. These techniques split the cost function evaluation in two phases: first, a simplified *motion estimation* is performed, in order to find a "good"[4] motion vector. Then, the actual impact of this vector (and only of it) on final rate and distortion is computed.

In details, the ME is performed by using some function of the so-called displaced frame difference (DFD) as distortion measure, without considering the quantized coefficients of the residual. Then, the best vector is computed as the solution of the Lagrangian problem

$$\mathbf{v}^*(\lambda_{\text{ME}}) = \arg\min_{\mathbf{v}\in V} D_{\text{DFD}}(\mathbf{v}) + \lambda_{\text{ME}} R(\mathbf{v}) \quad (7)$$

where $R(\mathbf{v})$ is the rate for encoding the motion vector $\mathbf{v}$. As far as $\lambda_{\text{ME}}$ is concerned, it has been shown in [20] that a reasonable choice is

$$\lambda_{\text{ME}} = \lambda_{\text{mode}}^{\frac{1}{\ell}}. \quad (8)$$

The cost function computation proceeds as follows:

$$\begin{aligned}\rho(\mathbf{v}^*(\lambda_{\text{ME}})) &= B - B_{\text{REF}}(\mathbf{v}^*(\lambda_{\text{ME}}))\\ \theta(\mathbf{v}^*(\lambda_{\text{ME}})) &= T\left[\rho(\mathbf{v}^*(\lambda_{\text{ME}}))\right]\\ \tilde{\theta}(Q_p, \lambda_{\text{ME}}) &= \mathcal{Q}\left[\theta(\mathbf{v}^*(\lambda_{\text{ME}})), Q_p\right]\\ \tilde{\rho}(Q_p, \lambda_{\text{ME}}) &= T^{-1}\left[\mathcal{Q}^*\left(\tilde{\theta}(Q_p, \lambda_{\text{ME}}), Q_p\right)\right].\end{aligned}$$

Like in the case of INTRA modes, a deadzone or a uniform quantizer can be used. In conclusion, the distortion associated to this mode is

$$\begin{aligned}D(Q_p, \lambda_{\text{ME}}) &= \left\|B - B_{\text{REF}}(\mathbf{v}^*(\lambda_{\text{ME}})) - \tilde{\rho}(Q_p, \lambda_{\text{ME}})\right\|_\ell\\ &= \left\|\rho(\mathbf{v}^*(\lambda_{\text{ME}})) - \tilde{\rho}(Q_p, \lambda_{\text{ME}})\right\|_\ell\end{aligned}$$

while the final rate is

$$R(Q_p, \lambda_{\text{ME}}) = R(\mathbf{v}^*(\lambda_{\text{ME}})) + R(\tilde{\theta}(Q_p, \lambda_{\text{ME}})) + R_{\text{mode}}.$$

In conclusion, in sight of (8) the mode cost function is

$$J_{\text{INTER}}(Q_p, \lambda_{\text{mode}}) = D(Q_p, \lambda_{\text{ME}}) + \lambda_{\text{mode}} R(Q_p, \lambda_{\text{ME}}).$$

Finally, we observe that there exist several variants of the INTER16×16 mode. In these variants the MB is split in two or four blocks (in this case, each of the four blocks can be further split into two or four sub-blocks). According to the block size, these modes are called INTER16×8, INTER8×16, and so on. Of course, these divisions increase the coding cost, because a new motion vector is needed for each sub-block. However the distortion is hopefully reduced, so it makes sense to perform a Lagrangian competition among the INTER modes.

While these modes provide a smaller distortion at the cost of a higher coding rate, the SKIP mode explores the RD curve at the opposite side. When this mode is used, only the signalling information is sent, and the MB is reconstructed by copying the macroblock from the reference image at a position inferred from the motion vectors of the neighbors MBs. This mode has an extremely low coding cost, but the reconstructed quality cannot be very good. However, this mode is extremely effective for low-activity areas and can dramatically improve the coding efficiency at low bit-rates or for low motion videos.

[4]The "best" vector is the one in (6).

### C. *The Cost Function of the* QMV Mode

First of all, we perform a Lagrangian ME similar to the one used for the INTER mode and described for example in [20]. The only differences are that the search grid can be in principle finer[5] and the Lagrangian parameter (used to trade off between cost and accuracy of MV) is not necessarily the same used for INTER MV estimation. We use the estimated vector $\mathbf{v}^*$ to compute the residual. Let $B$ be the original MB, and $B_{\text{REF}}(\mathbf{v})$ the MB from the reference slice, compensated with a MV $\mathbf{v}$; then $\rho(\mathbf{v}^*) = B - B_{\text{REF}}(\mathbf{v}^*)$ is the motion-compensated residual, $\tilde{\theta} = \text{fix}\left(\frac{T[\rho]}{Q_p}\right)$ is the quantization index of the transformed residual, and $\tilde{\rho}(Q_p) = T^{-1}\left[\mathcal{Q}^*(\tilde{\theta}, Q_p)\right]$ is the reconstructed residual at the decoder.

Then, we send to the decoder the *quantized version* of the vector $\mathbf{v}^*$. For coherence with previous notation, we call $\tilde{\mathbf{v}}(Q_v)$ the quantization index. It can be obtained by uniform scalar quantization of the vector's components: in this case the reconstructed vector would be $Q_v\tilde{\mathbf{v}}(Q_v) = Q_v\text{round}(\mathbf{v}^*/Q_v)$, but any degraded version of $\mathbf{v}^*$ could be used, provided that its coding cost is less than the original vector, and the associated distortion is not much higher. As we will shown in Section IV this can be achieved in different ways. However, in the rest of this section for simplicity we will consider the case of uniform quantization. In conclusion, the reconstructed MB $\widetilde{B}$ is obtained by adding to the residual $\tilde{\rho}(Q_p)$, a motion-compensated prediction computed with $\tilde{\mathbf{v}}$

$$\widetilde{B}(Q_p, Q_v) = B_{\text{REF}}(Q_v\tilde{\mathbf{v}}(Q_v)) + \tilde{\rho}(Q_p). \tag{9}$$

In this equation we observe that the decoded block depends on both $Q_p$ and $Q_v$, while, in the classic INTER case, it would of course depend only on $Q_p$. This is a key point for our mode since, by tuning the two quantization steps, we can manage the RD tradeoff between coefficients and MVs representation. Coming to the QMV mode cost function, it is easy to show that

$$D(Q_p, Q_v) = \left\|\rho(Q_v\tilde{\mathbf{v}}(Q_v)) - \tilde{\rho}(Q_p)\right\|_\ell \tag{10}$$

where $\rho(Q_v\tilde{\mathbf{v}}(Q_v)) = B - B_{\text{REF}}(Q_v\tilde{\mathbf{v}}(Q_v))$ is the residual of the motion compensation with the quantized vector; the rate is given by

$$R(Q_p, Q_v) = R_{\text{mode}} + R[\tilde{\theta}(Q_p)] + R[\tilde{\mathbf{v}}(Q_v)] + R(Q_v). \tag{11}$$

The signalling rate $R_{\text{mode}}$ and the coefficient rate $R[\tilde{\theta}(Q_p)]$ are computed as usual via the entropy encoder; the MV rate $R[\tilde{\mathbf{v}}(Q_v)]$ depends on the MV coding technique, described in Section IV; moreover we observe that in principle a different $Q_v$ could be used in each MB, so we account for its coding cost in (11) with the term $R(Q_v)$. This issue is tackled in Section III. In conclusion, the resulting cost function for the QMV mode is

$$J_{\text{QMV}}(Q_p, Q_v, \lambda_{\text{mode}}) = D(Q_p, Q_v) + \lambda_{\text{mode}} R(Q_p, Q_v). \tag{12}$$

For some assigned $Q_v$, $Q_p$, and $\lambda_{\text{mode}}$, we find the cost function for the new mode. This value should be compared with the cost function of the other modes, and if it is the smallest, the QMV mode is selected.

## III. Selection and Encoding of the Motion Quantization Step

A central problem for an efficient implementation of the QMV mode is the selection and encoding of the quantization step $Q_v$ of the current MV. To this end, we consider the cost function $J(Q_p, Q_v, \lambda_{\text{mode}})$ as in (12). We consider that $Q_p$ has been chosen as input parameter and as a consequence, $\lambda_{\text{mode}}$ is determined by (2), so we drop the dependency of $J$ from these parameters. Thus for each MB, we would like to use the best $Q_v$, i.e., the one minimizing $J$: $Q_v^* = \arg\min_{Q_v\in\mathbb{R}} J(Q_v)$. However if we use too many bits to represent $Q_v$ for each macroblock, all the rate saving obtained by the quantization of MVs is lost. So we resort to a simpler solution by using a *double-pass* coding strategy. In a first scanning of the current slice, we gather the estimation of the cost function in $J(Q_v, k)$, where $k \in \{1, 2, \ldots, K\}$ is the MB index, $K$ is the number of MBs, and $Q_v$ is allowed to vary in a discrete set[6] $S_Q$. In this first step we do not encode any MB, since the actual $Q_v$ value has not been chosen yet. However at the end of this first step we have an estimation of the cost function values as function of $Q_v$ and of the MB index $k$. Then we try to represent in an efficient way the whole vector $\mathbf{Q}_v^* = \{Q_v^*(1), Q_v^*(2), \ldots Q_v^*(k), \ldots\}$, where $Q_v^*(k)$ is the best step for the $k$th MB. Many strategies can be envisaged for the representation of $\mathbf{Q}_v^*$, and the rate $R(Q_v)$ depends on the strategy and the set $S_Q$. We consider a couple of possible cases, one ideal, the other very simple, but effective.

### A. *"Oracle" Strategy*

The encoder uses the optimal vector $\mathbf{Q}_v^*$, but no bit is accounted for its coding cost. This gives us an upper bound on the achievable performance of the QMV mode, and it corresponds to the case of an ideal coding of $\mathbf{Q}_v^*$ (or, to the case of an "oracle" decoder, capable to know the $Q_v$ used for each MB). In other words, in this case we should have $R(Q_v) \approx \log_2|S_Q|$, but we set $R(Q_v) = 0$.

### B. *"Minsum" Strategy*

We use a single value of $Q_v$ for the whole slice, namely the one minimizing $\sum_k J(Q_v, k)$. In this way the coding cost of $Q_v$ (coded by CABAC or CAVCL) is practically negligible, since it is shared among all the MBs of the slice: $R(Q_v) \approx \log_2|S_Q|/K$.

The "Oracle" strategy allows us to evaluate the performance of the new mode in an ideal case. Even though one can expect that the "Oracle" strategy always performs better than the "Minsum" one, experiments show that sometimes this is

[5]In our experiments, we use a $\frac{1}{8}$-pixel grid for ME, because we want a good accuracy for the MVs before quantization. This is accurate enough, as we are targeting low-to-medium bit-rates. Moreover, the $\frac{1}{8}$-pel ME is already implemented in the KTA.

[6]The process of quantization consists in *a priori* subdividing the range of a signal in a finite number of intervals, so the choice of the intervals size has a sensible effect. In this case, the variation of the MVs has to be evaluated, in order to adapt the dynamic of the quantizer to the one of the data.

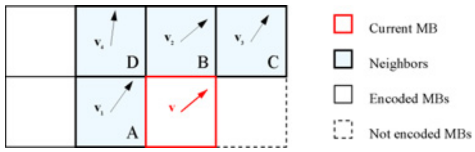


Fig. 3. Neighborhood used for coding MVs in the QMV mode D is used for prediction when C is unavailable.

not the case. The reason is that the choice of quantization step vector is performed minimizing a cost function which is only estimated: it is the cost function that we would have if all blocks were coded with QMV, so it is not always a exact estimation. However, we have to determine the vector $\mathbf{Q}_v^*$ in order to obtain the *real* cost function, so we are in a dead end. We decided to compute the cost function as described because this is a simple solution and, as shown in the experimental section, it gives good results. On the other hand, the Minsum strategy can be implemented in practice.

## IV. Coding the Quantized Motion Vectors

The new mode has to be competitive with the standard INTER modes, which benefit from a high-performance lossless coding of motion vectors. This coding exploits the median predictor based on neighboring vectors, and on CABAC coding of the prediction error. As a consequence, the coding cost for MVs can be quite small. Therefore, we need to use an efficient lossless coding of quantized motion vectors in order to really obtain a rate reduction. Our idea is to extend the H.264/AVC MV coding technique to cases where the QMV are included. We have then to define how to code the vectors for the QMV mode and how to code INTER MVs when some of the neighbors are quantized. We present in Section IV-A our method for coding the quantized vectors. In Section IV-B, we deal with the problem of the predictive coding of these vectors. Section IV-C presents an approach for further improving the prediction of the quantized MVs.

### A. Coding of the Quantized Motion Vectors

The motion-compensated modes in H.264/AVC perform a very efficient MV coding: with reference to Fig. 3, the current MV $\mathbf{v}$ is predicted as the component-by-component median of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ and the prediction error is encoded with the entropy coder. If the MV in C is not available, the vector of D is used instead, while if other vectors are unavailable yet, one of the available vectors is used according to the reference slice of the current and neighboring blocks.

For the QMV mode, the fact that the MBs can have vectors quantized with different steps is taken into account (an INTER MB is considered as a QMV MB with a quantization step equal to the ME resolution). This is managed by considering the *de-quantized* value $Q_v \cdot \tilde{\mathbf{v}}(Q_v)$ for all the available vectors, be them QMV MB or ordinary INTER MB. This de-quantization brings back all the vectors of the neighborhood to the same resolution, so that the median operator can be applied to them. If we call $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ the de-quantized vectors for MBs $A$, $B$, and $C$, respectively (supposing that all of them are available), we define the predictor $\hat{\mathbf{v}}$ of the current vector as the median of $\mathbf{v}_1$,$\mathbf{v}_2$, and $\mathbf{v}_3$. The vector prediction is quantized using the step $Q_v$, and the prediction error $\epsilon(\tilde{\mathbf{v}})$ is sent to the entropy coder

$$\epsilon(\tilde{\mathbf{v}}) = \tilde{\mathbf{v}} - \tilde{\hat{\mathbf{v}}} = \text{round}\left(\mathbf{v}^*/Q_v\right) - \text{round}\left(\hat{\mathbf{v}}/Q_v\right).$$

If some vectors of macroblocks $A$, $B$, and $C$ are not available, the neighborhood is formed according to the standard rules of H.264/AVC.

Even in the case when the current MB is an ordinary INTER, the MV coding technique must be adapted because the neighbors could be QMV. Since an INTER MB is considered as a QMV MB with a quantization step equal to the ME resolution, we can simply apply the technique we have just described: predictive coding of de-quantized vectors and possibly re-quantization with the appropriate quantization step. We explicitly remark that, when none of the neighbors of the current MV is a QMV MB, this technique becomes equivalent to the ordinary MV coding of H.264/AVC.

### B. Switch on the Prediction of the Quantized Vectors

The MVs can be quantized with different $Q_v$ but smaller dimension of MBs can lead to a wrong prediction. We know that MVs are predicted from a suitable neighborhood and the prediction error is losslessly encoded. The predictor is obtained from the median of the de-quantized vectors, but each one is de-quantized with its optimal step, and if the steps are very different, the reconstruction levels have a different precision. The more these levels are different, the more the error on prediction of the current block will increase because the predictor does not utilize the original version of MV but the de-quantized version. For example, in the QMV8×8 mode where the MBs are split into four blocks, this perturbation is not negligible, and it could even further increase for smaller block sizes. We denote the prediction error as $\epsilon(\widetilde{\mathbf{v}}) = \widetilde{\mathbf{v}} - \tilde{\hat{\mathbf{v}}}$, with $\widetilde{\mathbf{v}}$ the quantized motion vector, and $\tilde{\hat{\mathbf{v}}}$ its prediction. The energy of prediction error for a slice is then $\sigma_\epsilon^2 = \sum \|\epsilon(\widetilde{\mathbf{v}})\|^2$, and it is compared with the energy of the quantized vectors for a slice $\sigma_v^2 = \sum \|\widetilde{\mathbf{v}}\|^2$. The expected result is that residual energy is smaller than vector energy, so transmitting the prediction error is more convenient than transmitting the vectors. But, if $\sigma_\epsilon^2 \geq \sigma_v^2$, the prediction gives no gain, and it is better to directly transmit the vectors.

This problem can increase the total bit-rate. The distortion is not impacted, because it is computed from transform coefficients, but the increase of bit-rate will deteriorate the coding efficiency of the QMV mode. We have analyzed some possible solutions in order to take into account the influence of prediction coming from different quantization steps. A first one is to introduce a criterion in order to choose if the transmission of prediction error is more convenient than the transmission of original vectors.

In order to evaluate the importance of the prediction, two different values of $J_{\text{QMV}}$ have to be considered in the first pass. For each $Q_v$ in the candidate set, one can compute

$$J_{\text{QMV,pred}}(Q_v) = D_{\text{pred}}(Q_v) + \lambda_{\text{mode}} R_{\text{pred}}(Q_v)$$

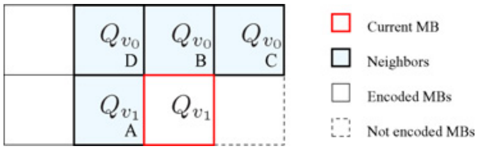


Fig. 4. Current MB (in red) and its neighborhood for quantized MVs prediction.

which gives the best quantization step $Q^*_{v,\text{pred}}$ in the case of predictive coding, and

$$J_{\text{QMV,vect}}(Q_v) = D_{\text{vect}}(Q_v) + \lambda_{\text{mode}} R_{\text{vect}}(Q_v)$$

which gives $Q^*_{v,\text{vect}}$. Then, the two different $J_{\text{QMV}}$, valuated with the respective best $Q^*_v$, are compared. At the second pass, if the minimum of the two $J_{\text{QMV}}$ corresponds to $J_{\text{QMV,pred}}$, computed with the rate of residual error, the second pass is not changed and $Q^*_{v,\text{pred}}$ is used. Otherwise, a non-predictive coding is done, i.e., in the second pass $J_{\text{QMV,vect}}$ is computed with $Q^*_{v,\text{vect}}$ and the rate obtained from the encoding of quantized MVs.

It should be remarked that, according to the $Q_v$ selection strategy of Section III, there exist two different ways to compare $J_{\text{QMV}}$. For the "Oracle" case, $J_{\text{QMV,pred}}(Q^*_{v,\text{pred}})$ and $J_{\text{QMV,vect}}(Q^*_{v,\text{vect}})$ are compared for each MB, because it can have different best $Q^*_v$ and the minimum between the predictive or not-predictive $J_{\text{QMV}}$ is chosen. For the "Minsum" case, the best $Q^*_v$ is chosen for the whole slice, so two global $J_{\text{QMV}}$ have to be estimated and then compared

$$J_{\text{QMV,pred}} = \sum_k J^k_{\text{QMV,pred}}(Q^*_{v,\text{pred}})$$

$$J_{\text{QMV,vect}} = \sum_k J^k_{\text{QMV,vect}}(Q^*_{v,\text{vect}})$$

where $k$ is the MB index. The choice of predictive or non-predictive coding is done for the whole slice.

### C. Adaptive Prediction Constrained on $Q_v$ Values

If the neighborhood is composed of quantized MVs with different $Q_v$ ("Oracle" case), the prediction error on current quantized motion vector could have a significant energy, when the accuracy of the motion vectors is very high. To reduce this energy, we only use for prediction the MVs that have been quantized with the same precision. In Fig. 4 for example, the current MB has quantization step $Q_{v_1}$. Normally, for its prediction, we compute a median with macroblocks A, B, C. If we consider the constraint on $Q_v$ values, we have to use only the MV of macroblock A, because its quantization step is equal to $Q_{v_1}$. If the current MB has quantization step $Q_{v_0}$, we can use the MVs of B and C, and so on.

## V. Experimental Setup and Results

The QMV mode has been integrated into the H.264/AVC JM-KTA[7] software (v.11.0 KTA 1.4 [5]), called H.264 in the following, with 1/8-pel ME enabled. In our experiments, we used the profile FREXT High, the following parameters for the ME: no fast ME, unrestricted search range, 32 pixels search range, no rate control, a GOP structure of IPP...P, and all the other parameters are set by default. Results for the 16×16 and 8×8 partitions, and results when both the 16×16 and 8×8 partitions are enabled, are presented. On the contrary, we do not consider the case where the 4×4 mode is enabled, since at low and low-to-medium rates this mode is selected very rarely, and therefore it does not affect significantly the RD performance. All of these results have been obtained at the decoder. Table I summarizes the sequences and the experimental setup. We consider the two variants of the proposed mode ("Oracle" and "Minsum") and we compare them to H.264/AVC (with 1/4 pel ME) and H.264/AVC KTA with 1/8 pel ME.

[7]The main difference between the JM and the JM-KTA is that the latter includes an 1/8-pel ME.

TABLE I
Test Sequences

| Sequence | Resol. | Frames | Experiments | |
|---|---|---|---|---|
| *Container* | CIF | 300 | RD chart | Fig. 7(a) |
| | | | Bjontegaard metric | Table II |
| *Foreman* | CIF | 300 | Bjontegaard metric | Table II |
| *Mobile* | CIF | 300 | Bjontegaard metric | Table II |
| *Tempete* | CIF | 300 | Mode distribution | Fig. 6 |
| | | | RD chart | Fig. 7(b) |
| | | | Bjontegaard metric | Table II |
| *City* | SD | 600 | Operational points, std modes | Fig. 1 |
| | | | Operational points, with new mode | Fig. 5 |
| | | | RD chart | Fig. 8(a) |
| | | | Bjontegaard metric | Table II |
| *Soccer* | SD | 600 | RD chart | Fig. 8(b) |
| | | | Bjontegaard metric | Table II |

### A. Operation Points

In a first test, the *City* sequence is coded with the modified encoder (but similar results have been obtained for other sequences and resolutions) and the average operation points of the encoding modes are computed. The results are shown in Fig. 5 for the "Minsum" mode. We observe a slight increase in distortion for the new mode (about 0.03 dB in the average) with respect to the INTER. On the other end, the average rate needed for the MBs encoded in QMV mode is about 19% smaller than the average rate of MBs encoded with the INTER. This does not mean that we should expect such a rate reduction when using the new mode in the coder, since it will not always be chosen. For this reason, it is interesting to evaluate the relative frequency of selected modes when the QMV is enabled: this is shown in the next section.

### B. Mode Distribution

In Fig. 6 we show the mode distributions for the four encoders (H.264/AVC at 1/4-pel and 1/8-pel motion estimations, and new coding mode with both strategies "Oracle" and "Minsum"), for the sequence *Tempete* at CIF resolution. The set of available $Q_v$ values is $S_Q = \{\frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$, weighted by the MVs dynamic range. In the "Oracle" case,

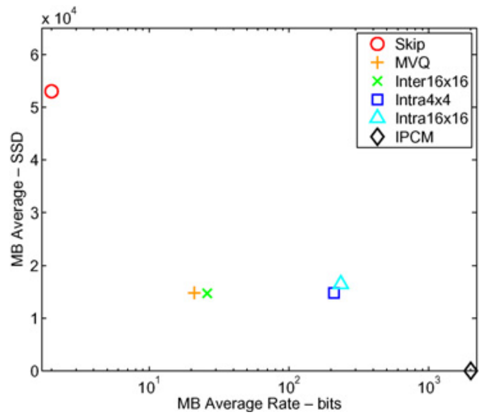


Fig. 5. Operation points with the new QMV 16×16 mode, sequence *City*.

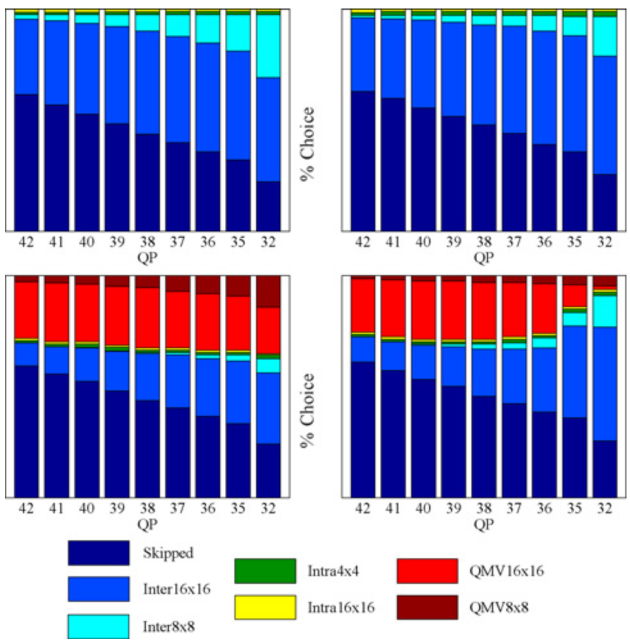


Fig. 6. Mode distribution, 16×16 and 8×8 enabled, for the sequence *Tempete*. First row: H.264 at 1/4-pel, H.264 at 1/8-pel, second row: QMV "Oracle," QMV "Minsum."

the QMV mode has replaced an important part of the INTER mode. This is reasonable since "Oracle" chooses the best $Q_v$ for each MB. When the more realistic "Minsum" strategy is used, the QMV mode is frequently chosen at low bit-rates (i.e., large $Q_p$). When the available bit-rate increases, the INTER mode is chosen more frequently. Once again, similar distributions have been observed for other sequences.

### C. Coding Performance

In order to assess the rate-distortion performance of the new mode, we considered all the six sequences shown in Table I, and we encode them using the four configurations, with 16×16 and 8×8 partitions enabled. Smallest partitions are not enabled since we are interested in low-to-medium bit-rate ranges, where in any case they would be selected very seldom and enabling them would not alter the RD performances. We also performed tests in which we enable one or both of the 16×16 and 8×8 partitions. This choice

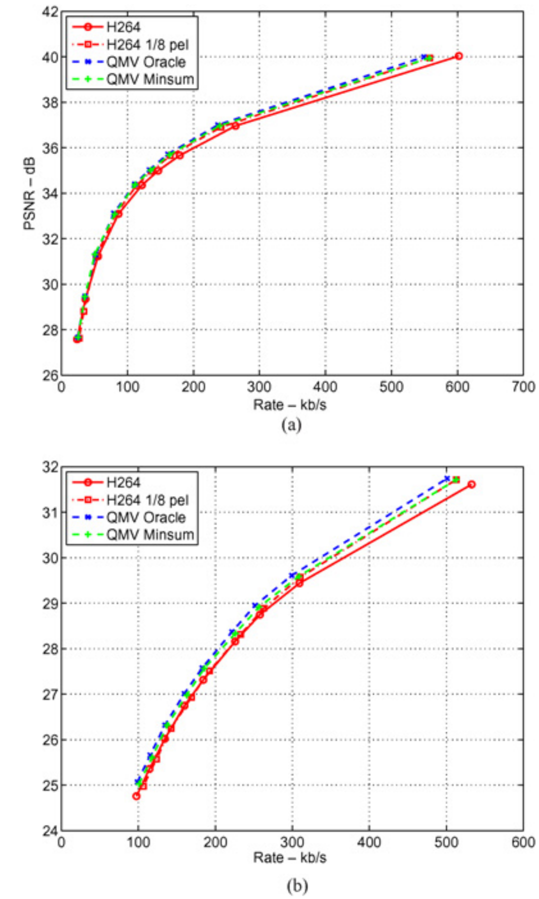


Fig. 7. Rate-distortion performance of the QMV coding mode, CIF sequences. (a) *Container* CIF at 30 frames/s. (b) *Tempete* CIF at 30 frames/s.

does not affect very much performance, so for the sake of simplicity we do not mention it in the following. However all the tests have been performed with the same partitions enabled for the four encoders. Complete results are reported in Table II, where we show the PSNR improvements and the percentual rate savings of the two QMV coders with respect to the two classic coders. We use the Bjontegaard metric [23], [24] (as recommended by the VCEG and JVT standardization groups), which provides the average bit-rate saving by computing the average difference between two RD curves, and also the difference between the distortion values expressed in PSNR. Two rate intervals are considered: low (corresponding to $Q_p$ ranging from 39 to 42) and medium ($Q_p$ from 36 to 39) rates. For these same experiments, we also report a few RD charts in Figs. 7 and 8. The first observation is that the proposed mode improves the RD performance in all the tests. The largest improvement is recorded w.r.t. H.264 with the 1/8-pel MV precision: this was expected since we are observing low-to-medium rates, where the bit-budget needed for high precision MVs is not affordable. When the QMV mode is enabled, the MVs rate and the MVs precision are adapted, and thanks to the quantization step this precision becomes variable and is optimized according to the RD cost function. On the contrary, with the classical implementation of H.264, the precision of the MVs is fixed. Second, we notice that the "Oracle" encoder has in general better performance than the "Minsum" one, but

TABLE II

PER CENT RATE SAVINGS (%R) AND DIFFERENCES BETWEEN PSNR VALUES (ΔPSNR) GIVEN BY THE BJONTEGAARD METRIC FOR THE QMV MODE AT DIFFERENT RATES ($Q_p$ BETWEEN 36 TO 39 FOR MEDIUM RATES, AND $Q_p$ BETWEEN 39 TO 42 FOR LOW ONES), COMPARED WITH H.264 AND H.264 WITH 1/8-PEL PRECISION (H264 $\frac{1}{8}$ PEL), FOR CIF (*Foreman*, *Tempete, Mobile, Container*) AND SD (*City, Soccer*) SEQUENCES

| Strategy | | "Oracle" | | | | "Minsum" | | | |
|---|---|---|---|---|---|---|---|---|---|
| Rate Range | | Low | | Medium | | Low | | Medium | |
| *Sequence* | Reference | *%R* | *ΔPSNR* | *%R* | *ΔPSNR* | *%R* | *ΔPSNR* | *%R* | *ΔPSNR* |
| *Container* | H264 | −5.82 | 0.26 | −11.83 | 0.41 | −6.51 | 0.31 | −9.93 | 0.34 |
| *Container* | H264 $\frac{1}{8}$pel | −5.22 | 0.24 | −4.23 | 0.15 | −5.94 | 0.29 | −2.33 | 0.08 |
| *Foreman* | H264 | −3.79 | 0.19 | −3.78 | 0.18 | −3.03 | 0.15 | −0.07 | 0.00 |
| *Foreman* | H264 $\frac{1}{8}$pel | −13.50 | 0.72 | −12.8 | 0.64 | −12.80 | 0.67 | −9.50 | 0.44 |
| *Mobile* | H264 | −7.14 | 0.33 | −8.49 | 0.40 | −4.22 | 0.19 | −3.38 | 0.15 |
| *Mobile* | H264 $\frac{1}{8}$pel | −8.03 | 0.38 | −6.00 | 0.30 | −5.15 | 0.24 | −0.87 | 0.04 |
| *Tempete* | H264 | −6.74 | 0.28 | −6.62 | 0.29 | −3.87 | 0.16 | −4.20 | 0.18 |
| *Tempete* | H264 $\frac{1}{8}$pel | −8.21 | 0.36 | −6.26 | 0.28 | −5.44 | 0.24 | −3.83 | 0.17 |
| *City* | H264 | −5.04 | 0.28 | −10.04 | 0.40 | −4.73 | 0.26 | −5.60 | 0.21 |
| *City* | H264 $\frac{1}{8}$pel | −11.39 | 0.65 | −8.55 | 0.36 | −10.98 | 0.65 | −4.08 | 0.15 |
| *Soccer* | H264 | −3.92 | 0.15 | −2.99 | 0.12 | −1.74 | 0.06 | −0.01 | 0.00 |
| *Soccer* | H264 $\frac{1}{8}$pel | −12.29 | 0.47 | −7.83 | 0.33 | −10.09 | 0.36 | −4.84 | 0.20 |

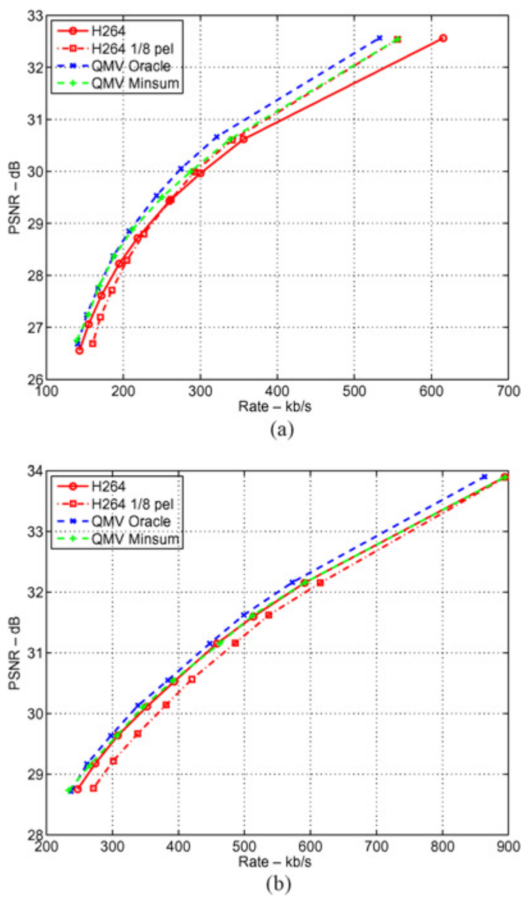


Fig. 8. Rate-distortion performance of the QMV coding mode, SD sequences. (a) *City* SD at 30 frames/s. (b) *Soccer* SD at 30 frames/s.

often the difference is small, and sometimes the "Minsum" encoder has better performance. These results are in line with our observations in Section III.

Finally we observe that the improvements are consistent when the sequence resolution is changed, and that proposed method takes better advantage from regular motion (e.g., *Container*, *City*) while the somewhat gains are limited when the sequence presents irregular movements (*Soccer*, *Foreman*).

## VI. CONCLUSION

Despite the excellent RD performances of the video coding standard H.264/AVC, some intuitions suggest that we can improve them using a more flexible motion coding. In this paper, we proposed a new coding mode based on motion vectors quantization. In order to insert this technique in the highly optimized H.264/AVC encoder, we solved some problems regarding the encoding of quantized MVs and the choice and encoding of the quantization step. The experimental results show that this new coding mode brings a non-negligible gain. In facts, the best performances are widely better than those of the H.264/AVC 1/4 or 1/8-pel coder.

Further improvements are expected when the proposed technique is extended to cover the cases where motion information is even more important, as it happens e.g., when sub-blocks of $4 \times 4$ pixels are enabled. We also think that an improvement of the coding efficiency could be achieved if any arbitrary real value can be chosen as MV quantization step. We are investigating whether a relationship can be found between the optimal $Q_v$ and the current quantization step for the coefficients. Another idea that we want to explore is to perform a layered coding of motion vectors, in order to have an efficient scalability in motion information as well. In this context, the work presented in this paper can help in tackling a major problem, that is the RD-optimized tradeoff between motion vector and residual rate for each layer, thanks to the proposed quantization strategy and the presented RDO framework.

## REFERENCES

[1] *Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14496-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: Apr. 1999, Version 3: May 2004.

[2] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 8: Consented in July 2007.

[3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[4] G. Laroche, J. Jung, and B. Pesquet-Popescu, "R-D optimized coding for motion vectors predictor selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1681–1691, Dec. 2008.

[5] *H.264 JM KTA Software Coordination, K. Suehring* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA

[6] Y. Y. Lee and J. W. Woods, "Motion vector quantization for video coding," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 378–382, Mar. 1995.

[7] R. L. Joshi, T. R. Fischer, and R. H. Bamberger, "Lossy coding of motion vector using entropy-constrained vector quantization," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3. Oct. 1995, pp. 3109–3112.

[8] S. L. Regunathan and K. Rose, "Motion vectors quantization in a rate-distortion framework," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Oct. 1997, pp. 21–24.

[9] L. A. Da Silva Cruz and J. W. Woods, "Adaptive motion vector quantization for video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Sep. 2000, pp. 867–870.

[10] J. Ribas-Corbera and D. L. Neuhoff, "Optimizing motion-vector accuracy in block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 497–511, Apr. 2001.

[11] R. Xiong, J. Xu, F. Wu, S. Li, and Y. Zhang, "Layered motion estimation and coding for fully scalable 3-D wavelet video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 4. Oct. 2004, pp. 2271–2274.

[12] M. Mrak, N. Sprljan, G. C. K. Abhayaratne, and E. Izquierdo, "Scalable generation and coding of motion vectors for highly scalable video coding," in *Proc. Picture Coding Symp.*, Dec. 2004, pp. 1–6.

[13] J. Barbarien, A. Munteanu, F. Verdicchio, Y. Andreopoulos, J. Cornelis, and P. Schelkens, "Motion and texture rate-allocation for prediction-based scalable motion vector coding," *Signal Process. Image Commun. (Elsevier Sci.)*, vol. 20, no. 4, pp. 315–342, Apr. 2005.

[14] Y. Wu and J. W. Woods, "Scalable motion vector coding based on CABAC for MC-EZBC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 790–795, Jun. 2007.

[15] M. Tagliasacchi, D. Maestroni, S. Tubaro, and A. Sarti, "Motion estimation and signaling techniques for 2-D+t scalable video coding," *EURASIP J. Appl. Signal Process.*, vol. 2006, no. 57308, p. 21, 2006.

[16] M. A. Agostini, M. Antonini, and M. Barlaud, "Model-based bit allocation between wavelet subbands and motion information in MCWT video coders," in *Proc. Eur. Signal Process. Conf.*, Sep. 2006, pp. 1–5.

[17] S. Kamp, J. Ballé, and M. Wien, "Multihypothesis prediction using decoder side-motion vector derivation in inter-frame video coding," *Proc. SPIE*, vol. 7257, no. 725704, Jan. 2009.

[18] I. Amonou, N. Cammas, G. Clare, J. Jung, L. Noblet, S. Pateux, S. Matsuo, S. Takamura, C. S. Boon, F. Bossen, A. Fujibayashi, S. Kanumuri, Y. Suzuki, J. Takiue, T. K. Tan, V. Drugeon, C. S. Lim, M. Narroschke, T. Nishi, H. Sasai, Y. Shibahara, K. Uchibayashi, T. Wedi, S. Wittmann, P. Bordes, C. Gomila, P. Guillotel, L. Guo, E. François, X. Lu, J. Sole, J. Vieron, Q. Xu, P. Yin, and Y. Zheng, *Description of Video Coding Technology Proposal by France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor*, document JCTVC-A114, Dresden, Germany, Apr. 2010.

[19] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

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

[21] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.

[22] T. Wiegand, G. Sullivan, and A. Luthra, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC)*, document JVT-G050r1, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Geneva, Switzerland, May 2003.

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves (VCEG-M33)," in *Proc. VCEG Meeting*, document ITU-T SG16 Q.6, Apr. 2001.

[24] S. Pateux and J. Jung, "An Excel add-in for computing Bjøntegaard metric and its evolution (VCEG-AE07)," in *Proc. VCEG Meeting*, document ITU-T SG16 Q.6, Jan. 2007.

**Marie Andrée Agostini-Vautard** received the Masters degree in image and signal processing from the University of Nice-Sophia Antipolis, Nice, France, in 2005, and the Ph.D. degree in information and communication technology from the University of Nice-Sophia Antipolis in 2009.

She was a Post-Doctoral Fellow with Queen Mary University, London, U.K., from 2009 to 2010. Since June 2010, she has been a Patent Engineer with Nony, Patent and Trademark Attorneys, Paris, France.

**Marco Cagnazzo** (M'05–SM'11) received the Laurea (equivalent to the M.S.) degree in telecommunication engineering from Federico II University, Napoli, Italy, in 2002, and the Ph.D. degree in information and communication technology from Federico II University and the University of Nice-Sophia Antipolis, Nice, France, in 2005.

He was a Post-Doctoral Fellow with the I3S Laboratory, Sophia Antipolis from 2006 to 2008. Since February 2008, he has been a Maître de Conférences (roughly equivalent to Associate Professor) with Institut TELECOM–TELECOM ParisTech, Paris, France, within the Multimedia Team. He is the author of more than 45 contributions in peer-reviewed journals and conferences proceedings. His current research interests include low-complexity, region-based, and content-adapted image coding, low complexity, scalable, robust, and distributed video coding, multiple description coding, and P2P video delivery.

Dr. Cagnazzo is the Area Editor for *Elsevier Signal Processing: Image Communication* and *Elsevier Signal Processing*. He is a reviewer for major international scientific reviews (IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS CIRCUITS SYSTEM VIDEO TECHNOLOGY, *Elsevier Signal Processing*, *Elsevier Signal Processing: Image Communication*, and others) and conferences (IEEE International Conference on Image Processing, IEEE MMSP, European Signal Processing Conference, and others).

**Marc Antonini** (M'05) received the Ph.D. degree in electrical engineering from the University of Nice-Sophia Antipolis, Nice, France, in 1991, and the Habilitation à Diriger des Recherches degree from the University of Nice-Sophia Antipolis in 2003.

He was a Post-Doctoral Fellow with Center National d'Etudes Spatiales (CNRS), Toulouse, France, in 1991 and 1992. He joined the CNRS in 1993 with the I3S Laboratory, both from the University of Nice-Sophia Antipolis and CNRS, and has been a Directeur de Recherche at CNRS since 2004. Since January 2008, he has been a Scientific Director of the CReATIVe Project and the Head of the IMAGES Research Group. He has been a Scientific Co-Director of French GDR-PRC ISIS since 2003 and is responsible in this GDR of the scientific organization and chairing of the research group: "Telecommunications: compression, transmission, protection." His current research interests include image coding, video coding, 3-D mesh coding, and 3-D+t mesh coding. Recently, he has also been interested in the analysis of the information contained by the neural code in the visual system, with applications in image compression.

Dr. Antonini was a member of the CNRS Expert Committee "Multidimensional and Multimodal Signals Processing" from 2005 to 2008. He is a regular reviewer for several journals (IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE INFORMATION THEORY, IEEE SIGNAL PROCESSING, and others), a member of the technical committees of several conferences (IEEE ICIP, IEEE ICASSP, EUSIPCO, IEEE MMSP, and others), and participated in the organization of the IEEE Workshop Multimedia and Signal Processing in 2001 in Cannes, France, and of the IEEE Workshop Multimedia and Signal Processing in 2010 in Saint-Malo, France. He participates in several national research and development projects with French industries and in several international academic collaborations.

**Guillaume Laroche** received the Ph.D. degree from Telecom ParisTech, Paris, France, in 2009. This thesis was conducted in partnership with Orange Labs, Issy Les Moulineaux, France.

His research was related to video-coding with competitive schemes for the next generation video codec and he integrated one new technology in the KTA software (international pre-study for high efficiency video coding). He joined Canon Research Center France, Paris, in October 2008 as a Research Scientist in the field of video coding.

**Joël Jung** received the Ph.D. degree from the University of Nice-Sophia Antipolis, Nice, France, in 2000.

From 1996 to 2000, he was with the I3S/CNRS Laboratory, University of Nice-Sophia Antipolis, where he was involved with the improvement of video decoders based on the correction of compression and transmission artifacts. He joined Philips Research France, Suresnes, France, in 2000 as a Research Scientist in video coding, post-processing, perceptual models, objective quality metrics, and low-power codecs. He is currently with Orange Labs, Issy Les Moulineaux, France. He is an active participant in standardization for video compression with more than 30 contributions to the video coding expert group ITU-T SG16-Q6, JVT, or JCT-VC. He holds more than 30 patents in the field of video coding.