

IMPROVING H.264 PERFORMANCES BY QUANTIZATION OF MOTION VECTORS

S. Corrado, M. A. Agostini, M. Cagnazzo*, M. Antonini

G. Laroche, J. Jung

Laboratoire I3S UNSA/CNRS
2000 route des Lucioles
06903 Sophia Antipolis Cedex France
{corrado, agostini, am}@i3s.unice.fr

Orange Labs
38-40 rue General Leclerc
92794 Issy Les Moulineaux France
joelb.jung@orange-ftgroup.com

ABSTRACT

The coding resources used for motion vectors (MVs) can attain quite high ratios even in the case of efficient video coders like H.264, and this can easily lead to suboptimal rate-distortion performance. In a previous paper, we proposed a new coding mode for H.264 based on the quantization of motion vectors (QMV). We only considered the case of 16x16 partitions for motion estimation and compensation. That method allowed us to obtain an improved trade-off in the resource allocation between vectors and coefficients, and to achieve better rate-distortion performances with respect to H.264. In this paper, we build on the proposed QMV coding mode, extending it to the case of macroblock partition into smaller blocks. This issue requires solving some problems mainly related to the motion vector coding. We show how this task can be performed efficiently in our framework, obtaining further improvements over the standard coding technique.

1. INTRODUCTION

The trade-off between allocation of coding resources to motion vectors (MV) or to transform coefficients has a major importance when it comes to efficient video coding techniques. Nevertheless, in video coding standards [1, 2], there is not much flexibility at this end: generally it is only possible to indirectly choose how the bit-rate is shared between motion and residual by selecting one among the several available coding modes for each macroblock (MB). Therefore, in several prior works, it has been noted that when a sequence is encoded at low and very low bit-rates, a large quota of resources is allocated to MVs [3]. This suggests that, in the framework of a H.264/AVC-like coder, there could be room for improvement if some new coding mode with less costly motion information is introduced, for example by quantizing the motion vectors.

The quantization of MVs has been an active research topic since the mid'90 [4, 5], but in these works the proposed approach was mainly a vector quantization (VQ) of MVs, with quantized vectors used both at the encoder and decoder sides (closed loop). For example in [6] a RD-optimized codebook for VQ of MVs is designed. In [7] a model based optimization of the MV precision is proposed, but here we are focused on a data-driven solution rather than a model-based one. In particular we are interested in the use of a new coding mode based on motion vector quantization.

However it should be noticed that any new mode introduced into a H.264-like coder causes an increase of the coding rate, because it is more costly to signal the chosen coding mode among

$N + 1$ than among N possible modes. On the other hand, if the coding mode is really adapted to some specific circumstances, it can have a less RD cost and be selected for some MB. In [8] we shown that in many cases the gain associated to the new mode are larger than the losses, so it is worth using this new mode, especially for sequences with an important motion. To this end, it was instrumental to use scalar quantization of motion vectors and a RD-optimized selection of the modes. More precisely, we proposed the lossy coding of MVs performed in an open loop system so that, while the motion-compensated residual is computed with the original MV, the latter is quantized before being sent to the decoder. This will reduce the coding rate, but can also increase the distortion as the motion-compensated prediction will be computed with the quantized vector instead of the original one. However the amount of quantization for the MVs is chosen in a RD-optimized way, taking into account the final effect on the decoded image. In [9] an approach more similar to the one proposed here was used; but in the framework of wavelet-based video coding, and the focus is on a model-based analytical evaluation of optimal trade-off between MV rate and coefficient rate.

The results in [8], however interesting they are, were limited by the fact that we only considered 16x16 MB in motion estimation. This was reasonable as first approach, above all when very low coding rates are considered. However, in order to fully enlighten the viability of the proposed QMV mode, we have to study its impact when smaller MB partitions are allowed. This involves some non trivial problems, mainly related to the coding of motion vectors. In this paper we show how these issues can be successfully dealt with, such that the QMV coding mode results even more competitive when smaller partitions are enabled.

The structure of the paper is the following: the QMV coding mode is briefly presented in Section 2. The main problems arisen with the implementation for the 8x8 partition and the solutions proposed are presented in Section 3. Experimental setup and results are reported in Section 4, and Section 5 draws conclusions.

2. QUANTIZATION OF MOTION VECTORS

In this section we recall the main properties of the QMV mode described in [8]. This mode was implemented within the JM implementation of H.264/AVC. The encoder will compare the new mode to the standard modes, in order to find the best one, *i.e.* the one minimizing the distortion for a given rate. This target is achieved by evaluating the lagrangian cost function J_i associated to the i -th mode: $J_i = D_i + \lambda_{\text{mode}} R_i$.

The new coding mode is quite simple: the MV \mathbf{v}^* is computed by classical motion estimation, and it is used in order to compute the motion-compensated residual ρ , which is then trans-

*This author performed the work while at Département TSI, TELECOM ParisTech, 37-39 rue Dareau, 75014 Paris, France; e-mail: cagnazzo@telecom-paristech.fr.

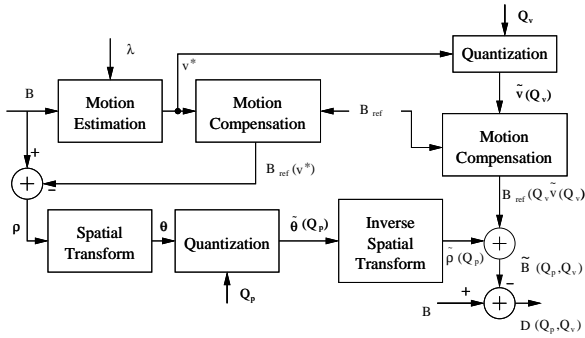


Fig. 1. Open loop quantization of motion vectors

formed, quantized and sent to the output (after lossless coding), like in all hybrid coders. The difference with a standard INTER mode is that the MV is quantized before being sent to the decoder. This process consists in a simple scalar uniform quantization of its components with a quantization step Q_v . Of course, the encoder must control the distortion caused by this quantization, which is accomplished by computing D at the encoder side, with a process described in figure 1. In the QMV mode, the prediction is computed using the quantized (using a step Q_v) vectors, and then the quantized (using a step Q_p) residual is added to it. Therefore the reconstructed MB, the associated rate and distortion depend on both the quantization steps: by tuning them, we can manage the RD trade-off between coefficients and MV use of resources.

A central problem for an efficient implementation of the QMV mode is the selection and encoding of the quantization step for the current MV. Indeed, if we use too many bits to represent Q_v for each MB, all the rate saving obtained by the quantization of MVs will be 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, with Q_v varying in a discrete set S_Q . At the end of this first step we have an estimation of the cost function values in function of Q_v for each MB. Then, we try to represent in an efficient way the whole vector $\mathbf{Q}_v^* = \{Q_v^*(1), Q_v^*(2), \dots, Q_v^*(k), \dots\}$, 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 chosen strategy. We consider here the:

“Oracle” strategy. The encoder uses the optimal vector \mathbf{Q}_v^* , but no bit is accounted for its coding cost. This corresponds to the case of an extremely efficient coding of \mathbf{Q}_v^* .

“Minsum” strategy. We use a single value of Q_v for the whole slice, namely the one minimizing $\sum_k J(Q_v, k)$, with k the number of the MB. In this way, the coding cost of Q_v is practically negligible, since it is shared among all the MBs of the slice.

We have presented in [8] some results for the QMV16x16 coding mode, which was reasonable for a first analysis. However, in order to better understand the usefulness of the 16x16 mode, it is necessary to test it when smaller partitions are enabled. We focus here on the QMV8x8 coding mode, which needs some adjustments of the vector coding technique. Two novel approaches for the motion prediction related to the motion quantization are proposed, and results with both the QMV16x16 and QMV8x8 coding modes enabled are presented in Section 4.

3. SOME THEORETICAL ISSUES RELATED TO THE QMV8x8 CODING MODE

For the QMV8x8 mode, the MBs are splitted in four sub-blocks 8x8. The MVs can be quantized with different Q_v (“Oracle” case) but the smaller dimension of MBs can introduce some disturbances on the prediction. We know that MVs are predicted from a suitable neighborhood and the prediction error is entropically encoded. The predictor is obtained from the median of the de-quantized vectors for the MBs of neighborhood. Each vector is de-quantized with his optimal step, so if the steps are very different, then the reconstruction levels have a different precision. The more these levels are different, the more the error on prediction of the current sub-macroblock will increase because the predictor does not utilize the original version of MV but the de-quantized version. In the QMV8x8 mode, this perturbation is not negligible, and it could increase when the size of sub-macroblocks is decreasing.

Potentially, the error on vector prediction could begin more significant than the original vector, so the prediction could not be a convenient strategy. We denote the prediction error as:

$$\epsilon(\tilde{\mathbf{v}}) = \tilde{\mathbf{v}} - \tilde{\tilde{\mathbf{v}}},$$

with $\tilde{\mathbf{v}}$ the quantized motion vector, and $\tilde{\tilde{\mathbf{v}}}$ its prediction. We also consider the energy of prediction error for a slice:

$$\sigma_\epsilon^2 = \|\sum \epsilon(\tilde{\mathbf{v}})\|^2,$$

and we compare it with the energy of the quantized vectors for a slice:

$$\sigma_v^2 = \|\sum \tilde{\mathbf{v}}\|^2.$$

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's better to directly transmit the vectors.

This problem can increase the total bit-rate, and can deteriorate the performances of the QMV encoder compared to those of the traditional H.264/AVC encoder. In the next sections, we analyze some possible solutions in order to take into account the influence of prediction coming from different quantization steps. A first solution is to introduce a criterion in order to choose if the transmission of prediction error is more convenient than the transmission of original vectors. Another solution is to adapt the prediction according to the quantization steps of the neighbor vectors.

3.1. Switch on the prediction of the quantized vectors

In order to evaluate the impact of the prediction, we have to consider two different values of J in the first pass. Indeed, for each Q_v in the testing set, we compute:

$$J_{\lambda, vect}(Q_v) = D + \lambda R_{pred},$$

and

$$J_{\lambda, pred}(Q_v) = D + \lambda R_{vect}.$$

Then, we compare the two different J , evaluated with the respective best Q_v^* . This comparison will determine which MV coding method will be used in the second pass. In other words, in the second pass we can do without the discarded method.

It should be remarked that, according to the Q_v selection strategy, we have two different ways to compare the J . For the “Oracle” case, we compare the $J_{pred}(Q_{v, pred}^*)$ and the $J_{vect}(Q_{v, vect}^*)$ for each MB, because it can have two different best Q_v^* and we

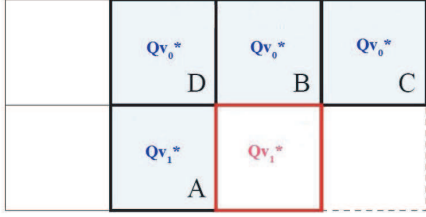


Fig. 2. The current MB (in red) and its neighborhood for quantized MVs prediction.

choose the minimum between the predictive or not-predictive J (with no additional information transmitted to the decoder). For the “Minsum” case, the best Q_v^* is chosen for the whole slice, so we have to estimate two global J and then to compare (with k the number of the MB):

$$J_{\lambda, vect} = \sum_k J_{k, \lambda, vect}(Q_{v, vect}^*),$$

and

$$J_{\lambda, pred} = \sum_k J_{k, \lambda, pred}(Q_{v, pred}^*).$$

3.2. Adaptive Prediction constrained on Qv 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 description of the motion is very precise. In order to reduce this energy, we only use for the prediction the MVs that have been quantized with the same precision. For an example, we consider figure 2, where 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 MB A. If the current MB has quantization step Q_{v_0} , we can use the MVs of B and C, and so on.

These improvements allow to obtain interesting results for the QMVcoding mode.

4. EXPERIMENTAL SETUP AND RESULTS

The QMV mode has been implemented over the H.264/AVC JM software (v.11.0 KTA 1.4 [10]), with 1/8-pel motion estimation enabled. We present results for the 8x8 partition, and results when both the 16x16 and 8x8 partitions are enabled. Both strategies described in Section 2 and approaches presented in Section 3 have been jointly considered. We studied the behavior of the proposed coder when smaller partitions are allowed, because this case allows up to a 0.5 dB improvement with respect to the case when only 16x16 partitions are allowed.

In order to assess the RD performance of the new codec, in a first test, we encoded the sequence *container* with the new encoder, with only the 8x8 partition enabled, and we computed the performances in terms of PSNR for the four configurations: the H.264/AVC codec at 1/4-pel and 1/8-pel motion estimations, and the new coding mode with both strategies “Oracle” and “Minsum”. The results are shown in figure 3. We can see that, as expected, the new mode has better results. Similar results have been obtained for other sequences.

Then, with the same 4 encoders and with both the 16x16 and 8x8 partitions enabled, we compressed several luminance-only CIF

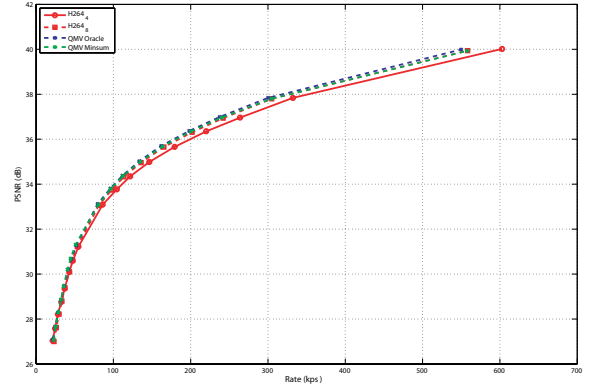


Fig. 3. RD performance, 8x8 coding mode, *container*. CIF @ 30 fps.

video sequences at 30 frames per second, *foreman*, *tempeste*, and *mobile*. We let Q_p assume the values 32 to 42, in order to check the behavior of the coders from low to medium bit-rates. 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.

In figure 4, we report the mode distribution for the 4 encoders for the sequence *tempeste*. We see that in the “Oracle” case, the QMV mode has almost always replaced 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.

In Table 1, we report the per cent rate savings of QMV modes with respect to the two H.264/AVC coders over several CIF sequences, using the Bjontegaard metric [11], as recommended by the VCEG and JVT standardization groups. We considered two rate intervals: low (corresponding to Q_p ranging from 39 to 42) and medium (Q_p from 36 to 39) rates. We see that the QMV encoders improve the performances with respect to H.264/AVC, and widely respect to H.264/AVC at the 1/8-pel precision. Indeed, high-resolution MVs are not worth at low rates, where the standard encoder has the best performance, while at high rates, we can afford high-resolution MVs. However, we see that with QMV mode we improve the 1/4 pel performances and the 1/8 pel performances, because we are able to adapt the MVs rate and the MVs precision. Indeed, thanks to the quantization step by slice or even by MB, this precision becomes variable and is optimized according to the complexity of the motion. On the contrary, with the classical implementation of H.264/AVC, the precision of the MVS is fixed (1/4- or 1/8-pel). We also observe that the “Oracle” coder has normally slightly better performances than the “Minsum” one, and even better than those obtained by the H.264/AVC coders.

Similar comments can be made about the RD curve for the sequence *tempeste*, shown in figure 5. We obtained the RD curves for the other sequences of our test set, and similar results were obtained.

5. CONCLUSIONS AND FUTURE WORK

Even though H.264/AVC has excellent RD performances, some intuitions suggest that we can improve them using a more flexible motion coding. In this paper, we propose improvements for

Strategy	Oracle		Minsum	
	Low	Medium	Low	Medium
foreman, H264 $\frac{1}{4}$ pel	-3,79	-3,78	-3,03	-0,07
foreman, H264 $\frac{1}{8}$ pel	-13,50	-12,84	-12,82	-9,50
tempete, H264 $\frac{1}{4}$ pel	-6,74	-6,62	-3,87	-4,20
tempete, H264 $\frac{1}{8}$ pel	-8,21	-6,26	-5,44	-3,83
mobile, H264 $\frac{1}{4}$ pel	-7,14	-8,49	-4,22	-3,38
mobile, H264 $\frac{1}{8}$ pel	-8,03	-6,00	-5,15	-0,87

Table 1. Per cent rate savings for the QMV mode at different rates (Q_p from 36 to 39), 16x16 and 8x8 enabled.

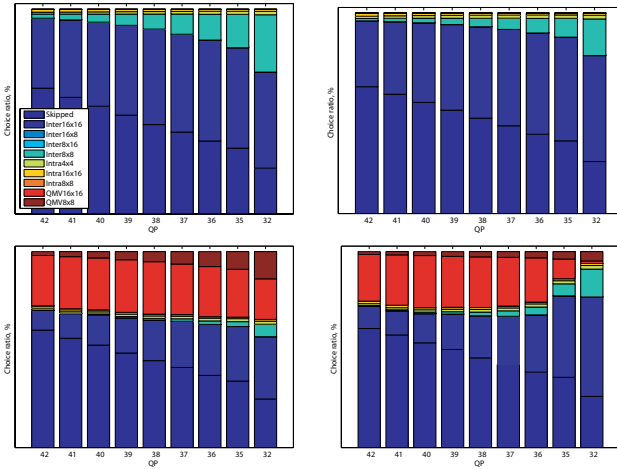


Fig. 4. Mode distribution, 16x16 and 8x8 enabled, *tempete*. First row: H.264 + 1/4-pel, H.264 + 1/8-pel; second row: QMV Oracle, QMV Minsum

a coding mode based on motion vectors quantization. In order to insert this technique in the highly optimized H.264/AVC encoder, we have solved some problems regarding the choice of the quantization step and the encoding of quantized MVs. In this paper, we focus on the 8x8 coding mode, and we resolve some issues due to the high precision of the motion. The experimental results show that this new coding mode brings a non-negligible gain. In fact, 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 will be extended to cover the cases where motion information is even more dominant, as it happens when sub-blocks of 4x4 pixels are enabled. We also think that a further improvement 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. Moreover we want to exploit the open loop structure to implement an efficient MV-scalable video coder.

6. REFERENCES

[1] *Coding of audio-visual objects—Part 2: Visual*, ISO/IEC

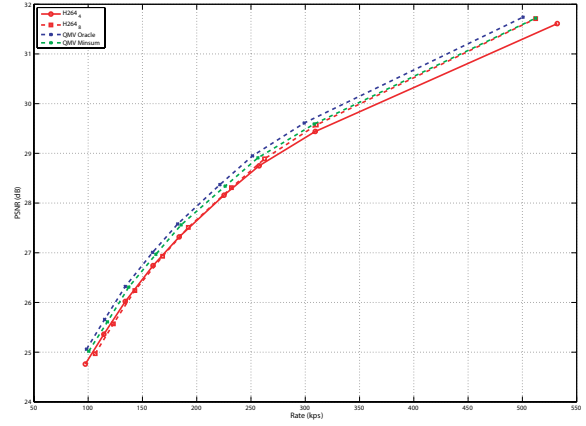


Fig. 5. RD performance, 16x16 and 8x8 enabled, *tempete*. CIF @ 30 fps.

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] G. Laroche, J. Jung, and B. Pesquet-Popescu, "A spatio-temporal competing scheme for the rate-distortion optimized selection and coding of motion vectors," in *Proceed. of Europ. Sign. Proc. Conf.*, Florence, Italy, Sept. 2006.
- [4] R. L. Joshi, T. R. Fischer, and R. H. Bamberg, "Lossy coding of motion vector using entropy-constrained vector quantization," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Washington D.C., USA, Oct. 1995, vol. 3, pp. 3109–3112.
- [5] Yoon Yung Lee and John W. Woods, "Motion vector quantization for video coding," *IEEE Trans. Image Processing*, vol. 4, no. 3, pp. 378–382, Mar. 1995.
- [6] S. L. Regunathan and K. Rose, "Motion vector quantization in a rate-distortion framework," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Santa Barbara, CA (USA), Oct. 1997, pp. 21–24.
- [7] 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.
- [8] M. Cagnazzo, M. A. Agostini, G. Laroche, J. Jung, and M. Antonini, "Motion vector quantization for efficient low bit-rate video coding," in *To appear in SPIE Electronical Imaging, Visual Communications and Image Processing conference*, San Jose, USA, 2009.
- [9] M. A. Agostini, M. Antonini, and M. Barlaud, "Model-based bit allocation between wavelet subbands and motion information in MCWT video coders," in *Proceed. of Europ. Sign. Proc. Conf.*, Florence, Italy, Sept. 2006.
- [10] *H.264 JM KTA software coordination*, K. Suehring, <http://iphome.hhi.de/suehring/tml/download/KTA/>.
- [11] S. Pateux and J. Jung, "An Excel add-in for computing Bjontegaard metric and its evolution," in *VCEG Meeting*, Marakech, MA, Jan. 2007.