Motion vector quantization for efficient low bit-rate video coding

M. Cagnazzo[a], M. A. Agostini [b], M. Antonini [b],G. Laroche [c], J. Jung [c]

[a]Département TSI , TELECOM ParisTech. 37-39 rue Dareau, 75014 Paris (France);
[b]Laboratoire I3S, UNSA/CNRS. 2000 route des Lucioles, 06903 Sophia Antipolis Cedex (France);
[c]France Télécom R&D, MAPS/MDG/SIA. 38-40 rue Leclerc, 92794 Issy Les Moulineaux (France)

ABSTRACT

The most recent video coding standard H.264 achieves excellent compression performances at many different bit-rates. However, it has been noted that, at very high compression ratios, a large part of the available coding resources is only used to code motion vectors. This can lead to a suboptimal coding performance. This paper introduces a new coding mode for a H.264-based video coder, using quantized motion vector (QMV) to improve the management of the resource allocation between motion information and transform coefficients. Several problems have to be faced with in order to get an efficient implementation of QMV techniques, yet encouraging results are reported in preliminary tests, allowing to improve the performances of H.264 at low bit-rates over several sequences.

Keywords: Video coding, motion vector, quantization, H.264

1. INTRODUCTION

An efficient resource allocation between motion vectors (MV) and motion-compensated residual is a key feature in any video coder aiming at good rate-distortion (RD) performances. In standard coders[1,2] 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). As a consequence, 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-like 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 some simple quantitative study. In Fig. 1 we report the average macroblock rate and distortion for several coding modes in a H.264 coder (JM v.11.0 KTA 1.4, Ref.[4]). These operation points have been obtained on the sequence *city*; for other sequences similar results have been obtained. We see that there is a significant gap between the low-cost, high-distortion SKIP mode and the relatively higher-cost, low-distortion INTER 16x16 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 16x16.

On one hand, the introduction of new coding modes increases the signalling cost (*i.e.* the coding cost of *any* selected mode), but on the other, it hopefully has a better RD performance for some MB. The main target of this paper is to verify that gains associated to the new mode surpass the losses. To achieve this target we propose the *lossy coding** of motion vectors, obtained via quantization. Moreover this lossy coding is performed in an open loop system so that, while the motion-compensated residual is computed with the motion vector $\mathbf{v}$, the vector $\mathbf{v}$ itself is quantized to $\tilde{\mathbf{v}}$ before being sent to the decoder. This will reduce the coding rate, but can also increase the distortion as the quantized vector $\tilde{\mathbf{v}}$ will be used to compute the motion-compensated (MCed) prediction

Further author information:
E-mail: cagnazzo@telecom-paristech.fr, {agostini,am}@i3s.unice.fr {guillaume.laroche,joelb.jung}@orange-ftgroup.com

*It is worth noting that in Ref.[3] authors achieved significant rate reduction by *lossless coding* of MVs.

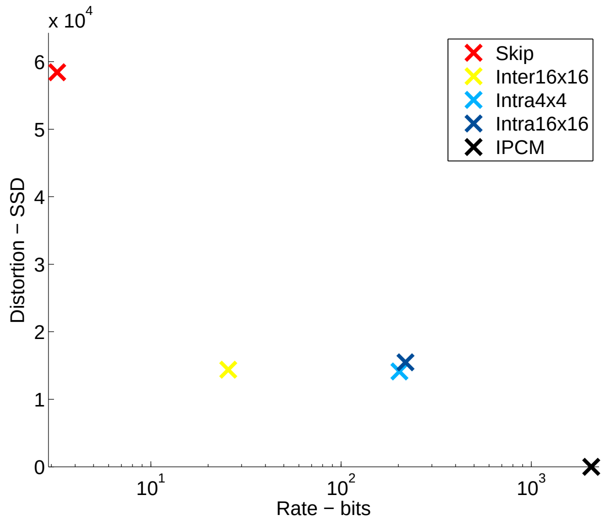

Figure 1. Average operation points of H.264 modes, sequence *city*

instead of $\mathbf{v}$. However, as explained in the following section, the amount of quantization for the motion vectors is chosen in a RD-optimized way, taking into account the final effect on the decoded image. As the main new tool of this mode is the quantization of the motion vector, we call it *quantized motion vector* mode (QMV).

The quantization of motion vectors has attracted the attention of the video coding community since the mid '90,[5,6] 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 (closed loop). For example in Ref.[7] a RD-optimized codebook for VQ of MVs is designed. In Ref.[8] a model based optimization of the MV precision is proposed. Our technique is not based on VQ and provides a data-driven solution rather than a model-based one. Moreover our solution is optimized for the open-loop scheme, which has better scalability properties (*e.g.* for MV scalability). In Ref.[9] an approach more similar to the one proposed here is 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. On the contrary, in this paper we refer to a H.264-like framework, and the rate allocation is performed via the direct evaluation of the RD cost function.

The introduction of the QMV techniques in this case requires to deal with several relevant theoretical and implementation issues, that constitute the main subject of this paper. The new coding mode, along with the main problems arisen with the implementation and the solution proposed, is shown in Section 2. Experimental setup and results are reported in Section 3, and Section 4 draws conclusions and ends the paper.

2. QUANTIZATION OF MOTION VECTORS

In this section we introduce the new coding mode in the framework of RD optimized video coding. We aim to insert this mode into the JM[4] implementation of H.264. 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 basic idea is that we can achieve a finer tuning of the rate allocation between vectors and residual coefficients because we can choose the coding mode not only among the non-QMV ones, but also by taking into account the QMV mode, with a suitable quantization step for MVs.

2.1 The QMV mode and its cost function

The new coding mode is quite simple: a relatively accurate (*i.e.* non-quantized) motion vector $\mathbf{v}^*$ is computed by classical motion estimation, and it is used in order to compute the MCed residual ρ, which is then transformed,

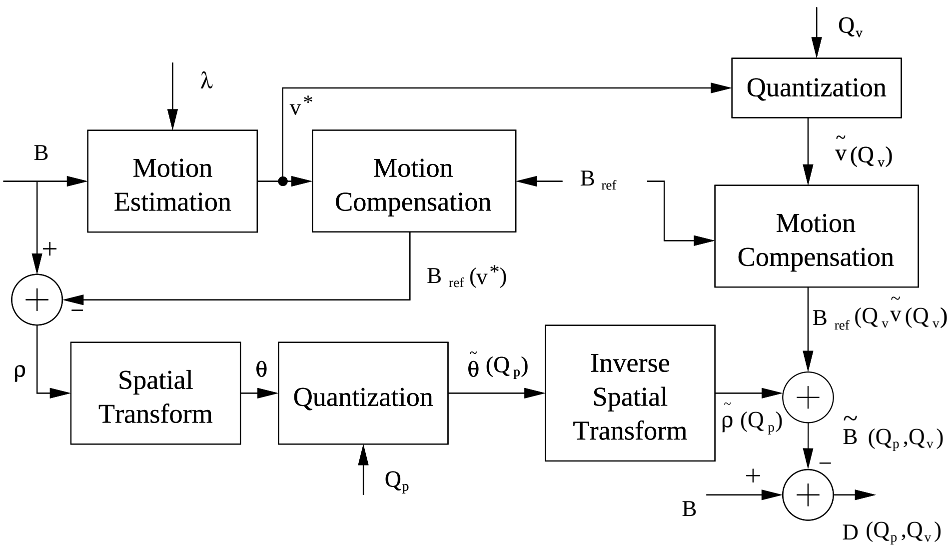

Figure 2. Open loop quantization of motion vectors

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 amounts to a simple scalar uniform quantization of its components with a quantization step Q_v. We differ to the section 2.3 the problem of efficiently selecting and encoding 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 depicted in Fig. 2.

First of all, we perform a lagrangian motion estimation (ME) similar to the one used for the INTER mode and described for example in Ref.[10] The only differences are that the search grid can be in principle finer[†] 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 macroblock, and $B_{\mathrm{REF}}(\mathbf{v})$ the MB from the reference slice, compensated with the MV $\mathbf{v}$; then $\rho(\mathbf{v}^*) = B - B_{\mathrm{REF}}(\mathbf{v}^*)$ is the MCed residual, $\tilde{\theta}(Q_p) = \mathrm{round}\left(\frac{T[\rho]}{Q_p}\right)$ is the transformed and quantized residual, and $\tilde{\rho}(Q_p) = T^{-1}\left[Q_p\tilde{\theta}(Q_p)\right]$ is the reconstructed residual at the decoder side.

Since we send to the decoder the *quantized vector* $\tilde{\mathbf{v}}(Q_v) = \mathrm{round}\left(\mathbf{v}^*/Q_v\right)$, the reconstructed MB $\widetilde{B}$ will be obtained by adding to the residual $\tilde{\rho}(Q_p)$, a MCed prediction computed with $\tilde{\mathbf{v}}$:

$$\widetilde{B}(Q_p, Q_v) = B_{\mathrm{REF}}(Q_v\tilde{\mathbf{v}}(Q_v)) + \tilde{\rho}(Q_p). \tag{1}$$

Equation 1 shows the difference between the QMV mode and classical MC-ed modes: the prediction B_{REF} depends on a quantized vector, and so the reconstructed MB depends on both the quantization steps Q_p and Q_v: tuning both of them we can manage the RD trade-off between coefficients and MV use of resources. Now, coming to the QMV mode cost function, it is easy to see that the distortion can be computed as follows:

$$\begin{aligned} D(Q_p, Q_v) &= \left\| B - \widetilde{B}(Q_p, Q_v) \right\|^p \\ &= \| B - B_{\mathrm{REF}}(Q_v\tilde{\mathbf{v}}(Q_v)) - \tilde{\rho}(Q_p) \|^p \\ &= \| \rho(Q_v\tilde{\mathbf{v}}(Q_v)) - \tilde{\rho}(Q_p) \|^p, \end{aligned} \tag{2}$$

where p is 1 for SAD and 2 for SSD and $\rho(Q_v\tilde{\mathbf{v}}(Q_v)) = B - B_{\mathrm{REF}}(Q_v\tilde{\mathbf{v}}(Q_v))$ is the residual with compensation via the quantized vector; the rate is given by

$$R(Q_p, Q_v) = R_{\mathrm{mode}} + R[\tilde{\theta}(Q_p)] + R[\tilde{\mathbf{v}}(Q_v)] + R(Q_v). \tag{3}$$

[†]We use a eighth-pixel grid for this motion estimation.

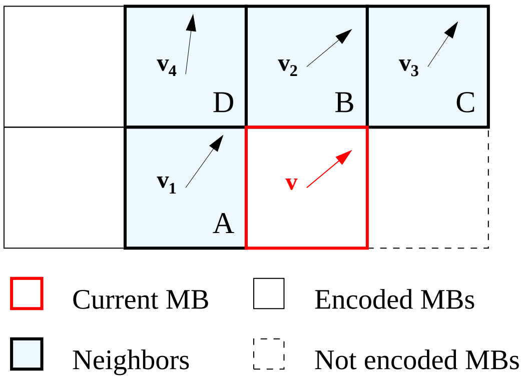

Figure 3. Neighborhood used for coding MVs in the QMV mode.

The signalling rate R_{mode} and the coefficient rate $R[\tilde{\theta}(Q_p)]$ are computed as usual via the entropic encoder; the MV rate $R[\tilde{\mathbf{v}}(Q_v)]$ depends on the MV coding technique, described in section 2.2; moreover we observe that in principle a different Q_v could be used in each MB, so we account for its coding cost in Eq. 3 with the term $R(Q_v)$. This issue is deawlt with in section 2.3.

In conclusion, the resulting cost function for the QMV mode is:

$$J_{\text{QMV}}(Q_p, Q_v, \lambda_{\text{mode}}) = D(Q_p, Q_v) + \lambda_{\text{mode}} R(Q_p, Q_v) \tag{4}$$

For some assigned Q_v, Q_p and λ_{mode}, we find the cost function for the new mode. This value should be compared with the cost function of the other modes. The QMV mode is selected if its cost function is less than others.

2.2 Coding of quantized motion vectors

The motion-compensated modes in H.264 perform a very efficient MV coding: with reference to Fig. 3, the current MV $\mathbf{v}$ is predicted as the 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 of it, while if other vectors are unavailable yet, one of the available vectors is used according to the reference images.

Since we want the new mode to be competitive with the existing one, we need an efficient way to encode quantized MV. The idea is to extend this coding technique to case where the QMV are included. So we have to define how to code the vectors for the QMV mode and how to code INTER MVs when some of the neighbors is QMV.

For the first case, the H.264 technique is extended to the QMV mode by taking into account that each MB can have vectors quantized with different steps (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 entropic 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 not all the vectors of macroblocks A, B and C are available, the neighborhood is formed according to the standard rules of H.264.

Even in the case that the current MB is an ordinary INTER, the MV coding technique must be updated because the neighbors could be QMV. Since an INTER MB can be considered as a QMV MB with a quantization step equal to the ME resolution, we can simply apply the technique we have just described: prediction from

[‡]The median is to be intended component by component.

de-quantized vectors and possibly re-quantization with the appropriate quantization step. We explicitly remark that, when none of the neighbor of the current MV is a QMV MB, this technique becomes equivalent to the ordinary MV coding of H.264.

2.3 Selection and encoding of the quantization step Q_v

A central problem for an efficient implementation of the QMV mode is the selection and encoding of the quantization step for the current MV. We consider the cost function $J(Q_p, Q_v, \lambda_{\text{mode}})$ as in equation (4). We can consider that Q_p has been chosen as input parameter and as a consequence, λ_{mode} is determined as well, since we assumed the usual relationship $\lambda_{\text{mode}} = 0.85 \cdot 2^{Q_p/3-4}$, see Ref.[10] Therefore, 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, and Q_v is allowed to vary in a discrete set 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 in 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 on the set S_Q. It is reasonable to start with some simple solutions, whose results will be useful to drive us in improving the Q_v selection strategy. So we consider:

"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 of the achievable performance of the QMV mode, and it corresponds to the case of an extremely efficient coding of $\mathbf{Q}_v^*$ (or, to the case of and "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 have $R(Q_v) = 0$.

"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 is practically negligible, since it is shared among all the MBs of the slice: $R(Q_V) \approx \log_2 |S_Q|/K$.

Many other strategies can be envisaged, with performance more realistic than the "oracle" and better than the "minsum". However, the results of these two techniques can prove extremely helpful in order to design more complex solutions.

3. EXPERIMENTAL SETUP AND RESULTS

The QMV mode has been implemented over the H.264 JM software (v.11.0 KTA 1.4 Ref.[4]), with eighth-pel motion estimation, but, for ease of implementation and of results interpretation, we use it only for the 16x16 partition. Other MB partitions are therefore disabled. Both strategies described in Section 2.3 have been considered.

In a first test, we encoded the sequence *city* with the new encoder, and we computed the average operation points of the encoding modes. The results are shown in Fig. 4 for the Minsum mode. We can see that, as expected, the new mode has a behavior intermediate between the SKIP and the INTER. Similar results have been obtained for other sequences.

In order to assess the RD performance of the new codec, we compared its two versions to the H.264 codec and to the H.264+$\frac{1}{8}$ pel ME (high profile with CABAC). With these 4 encoders we compressed several luminance-only CIF video sequences at 30 frame per second. We let Q_p assume the values 32 to 42. The set of available Q_v values was $S_Q = \{\frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}\}$.

In Fig. 5 we report the mode distribution for the 4 encoders. We see that in the Oracle case, the QMV mode has almost always replaced the INTER mode. This is reasonable since the Oracle choices 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

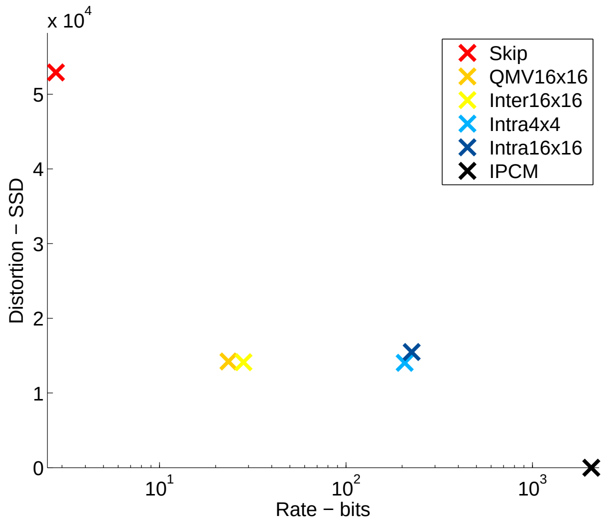

Figure 4. Operation points with the new mode

Total rate	Oracle Low	Oracle Medium	Oracle High	Minsum Low	Minsum Medium	Minsum High
city, H264	2.27	**-2.86**	**-5.33**	1.60	**-0.98**	**-3.80**
city, H264 $\frac{1}{8}$pel	**-2.09**	**-2.40**	**-2.71**	**-4.32**	**-0.12**	**-1.12**
tempete, H264	**-2.23**	**-3.88**	**-6.16**	**-0.77**	**-2.19**	**-5.49**
tempete, H264 $\frac{1}{8}$pel	**-3.67**	**-1.35**	0.02	**-2.24**	0.33	0.74
waterfall, H264	0.93	0.07	**-5.08**	0.28	**-0.80**	**-2.34**
waterfall, H264 $\frac{1}{8}$pel	**-7.28**	**-3.85**	**-3.08**	**-7.87**	**-4.58**	**-0.13**

Table 1. Per cent rate savings for the QMV mode at different 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 Tab. 1 we report the per cent rate savings of QMV modes with respect to the two H.264 coders over several CIF sequences, using the Bjontegaard metric[11](implementation: Ref.[12]), as recommended by the VCEG and JVT standardization groups. We considered three rate intervals: low (corresponding to Q_p ranging from 39 to 42), medium (Q_p from 35 to 38) and high (Q_p from 32 to 35) rates. We see that the QMV encoders generally improve with respect to H.264 at medium-to-high rates, and to H.264+$\frac{1}{8}$ at low-to-medium rates, since high-resolution MVs are not worth at low rates, where the standard encoder has the best performance, while when the available resources increase, we can afford more costly MVs. However, we see that with QMV mode we are able to adapt the MV rate, approaching or improving the 1/4 pel performances at low rates and the 1/8 pel performances at higher rates. We also observe that the Oracle coder has normally slightly better performances than the Minsum one and even better than those that we could obtain by switching the H.264 coder from the eighth pixel to the quarter pixel mode; the Minsum coder has performances comparable to and sometimes better than the switched H.264 coder.

Similar comments can be made about the the RD curve for the sequence *city*, shown in Fig. 6. We obtained the RD curves for the other sequences of our test set, and similar results were obtained.

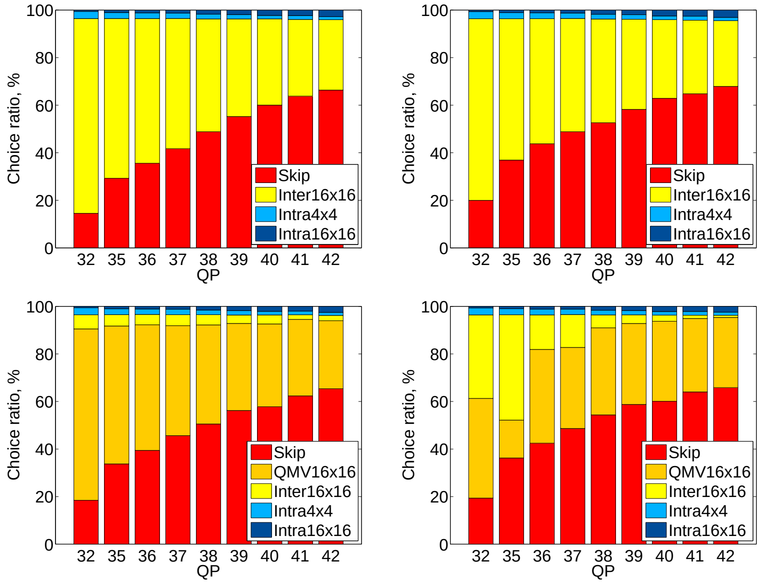

Figure 5. Mode distribution, *city*. First row: H.264, H.264 + 1/8-pel; second row: QMV Oracle, QMV Minsum

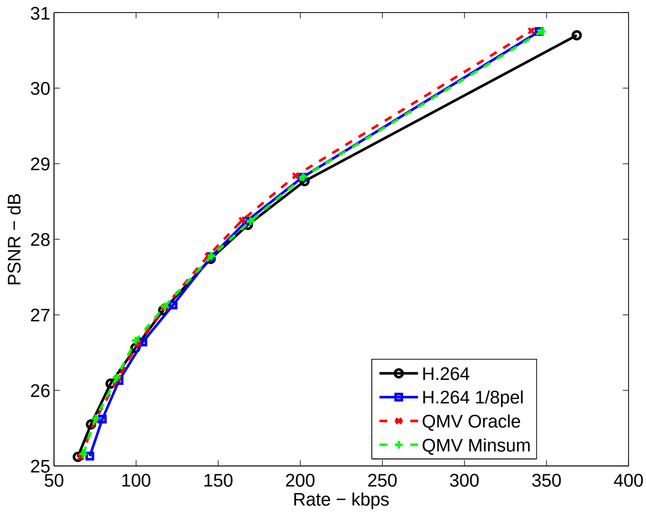

Figure 6. RD performance, *city*. CIF @ 30 fps

4. CONCLUSIONS AND FUTURE WORK

Even though H.264 has excellent RD performances, some intuitions suggest that we can improve them using a more flexible motion coding. In this paper we propose a new coding mode based on motion vectors quantization. In order to insert this technique in the highly optimized H.264 encoder, we must solve some problems regarding the choice of the quantization step and the encoding of quantized MVs. In this paper we showed how these problems can be solved and we report some experimental results, showing that this new mode promises a non-negligible gain. In facts, the upper bound performances are usually better than those of an hypothetical H.264 coder switching from quarter to eighth-pel resolution according to the global rate, while even with the simplistic Minsum strategy we achieve similar performance of the switched H.264 coder. More complex strategies will hopefully allow higher coding gains.

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 (4x4 pixels) are enabled. So our future work is oriented toward the use of quantized motion vectors for the 4x4 mode. 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.

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] 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] *H.264 JM KTA software coordination, K. Suehring*, http://iphome.hhi.de/suehring/tml/download/KTA/.

[5] R. L. Joshi, T. R. Fischer, and R. H. Bamberger, "Lossy coding of motion vector using entropy-constrained vector quantization," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Washinton D.C., USA, Oct. 1995, vol. 3, pp. 3109–3112.

[6] 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.

[7] 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.

[8] 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.

[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] 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, July 2003.

[11] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, Austin, USA, Apr. 2001.

[12] S. Pateux and J. Jung, "An Excel add-in for computing Bjontegaard metric and its evolution," in *VCEG Meeting*, Marrakech, MA, Jan. 2007.