Motion Vector Forecast and Mapping (MV-FMap) Method for Entropy Coding based Video Coders

Julien Le Tanou, Jean-Marc Thiesse, Joël Jung, Marc Antonini

▶ **To cite this version:**

HAL Id: hal-00531819
https://hal.science/hal-00531819

Submitted on 3 Nov 2010

Motion Vector Forecast and Mapping (MV-FMap) Method for Entropy Coding based Video Coders

Julien Le Tanou [#1], Jean-Marc Thiesse [#2], Joël Jung [#3], Marc Antonini [*4]

Orange Labs
38 rue du G. Leclerc, 92794 Issy les Moulineaux, France
[1] jletanou@gmail.com
{[2] jeanmarc.thiesse, [3] joelb.jung}@orange-ftgroup.com

* *I3S Lab. University of Nice-Sophia Antipolis/CNRS*
2000 route des Lucioles, 06903 Sophia Antipolis, France
[4] am@i3s.unice.fr

Abstract—Since the finalization of the H.264/AVC standard and in order to meet the target set by both ITU-T and MPEG to define a new standard that reaches 50% bit rate reduction compared to H.264/AVC, many tools have efficiently improved the texture coding and the motion compensation accuracy. These improvements have resulted in increasing the proportion of bit rate allocated to motion information. Thus, the bit rate reduction of this information becomes a key subject of research. This paper proposes a method for motion vector coding based on an adaptive redistribution of motion vector residuals before entropy coding. Motion information is gathered to forecast a list of motion vector residuals which are redistributed to unexpected residuals of lower coding cost. Compared to H.264/AVC, this scheme provides systematic gain on tested sequences, and 2.3% in average, reaching up to 4.9% for a given sequence.

I. INTRODUCTION

The H.264/AVC standard [1], which is the result of the joint work of the Video Coding Experts Group (VCEG) and Moving Picture Experts Group (MPEG), has achieved significant compression gain over its predecessor, H.263 and MPEG-4 part 2. This comes from the improvement of existing tools and the introduction of new ones, especially: various Intra predictors, variable block sizes, multiple reference frames and ¼ pel motion accuracy for motion compensation, deblocking filter and Context Adaptive Binary Arithmetic Coding (CABAC).

Today, new standardization activities in video coding are launched. VCEG and MPEG work collaboratively within the Joint Collaborative Team on Video Coding (JCT-VC) and have issued a Call for Proposals with answers expected by April 2010. The objective is a standard that reaches 50% bit rate reduction for the same subjective quality, with an allowed increase of the complexity by a factor 2 or 3. Several improvements are already known and gathered in the JM KTA (Key Technical Area) [2]. In particular, MV-Comp [3] efficiently acts on the motion coding. However, the coding cost of motion vectors remains high and its reduction is a major way to improve the next standard. Based on several observations that characterize the movement, such as the strong correlation

between motion vectors of neighboring frames and blocks, we propose in this paper a method for motion vector coding based on a motion vector residuals forecast followed by an adaptive mapping of residuals. This scheme is noted MV-FMap for Motion Vector Forecast and Mapping.

The remaining of this article is organized as follows: we first introduce a state of the art of motion vector coding in Section II, then we describe the proposed scheme in Section III. Finally, we present the experimental results in Section IV.

II. STATE OF THE ART

A. H.264/AVC Motion Vector Coding

The H.264/AVC reference codec is based on a competition between multiple Intra and Inter prediction coding modes, through the calculation of a rate-distortion (RD) criterion [4]. For Inter prediction, a block matching motion estimation is performed in order to exploit the temporal redundancies of video sequence. A predictive coding of the motion vector is then applied, with the motion vector residual r defined by $r = mv - mvp$, where mv is the motion vector, mvp the motion vector predictor defined by a median of the motion vectors of the spatially neighboring blocks. The accuracy of the motion vector prediction is sensitive to irregular motion and remains a key feature of the motion vector coding.

Finally, an entropy coding is independently applied to each component of the residual of the optimally selected motion vector. CABAC is currently the most powerful as it enables to adapt, for each block, the coding table applied to residual vectors. However, this table is defined from a finite number of probability models. These models are selected by a context criterion which can be insufficient. Furthermore, although the probabilities defined by these models are updated according to the occurrence statistics of each bit of the binarized residual, this coding is essentially a solution based model and can be supplemented by a method more driven by a motion vector forecast in addition to these observed data.

B. Recent Motion Vector Coding Improvements

Based on the coding scheme described above, several tools have been proposed to improve the efficiency of motion vector

coding. In [3], a proportion of the motion information in the total bit stream of nearly 35% is reported at low bit rate. To reduce the cost of this motion information, a method based on a spatio-temporal motion vector predictor competition (MV-Comp) is proposed. It takes advantage from the use of spatial and temporal redundancies in the motion vector fields, where the simple spatial median usually fails. Although an index transmission is required to signal the selected predictor, this coding scheme provides a very good compromise between coding efficiency and added complexity.

On top of the MV-Comp framework, several contributions on the motion vectors prediction (MVP) have been recently made. In [5], the set of predictor candidates is first adaptively determined based on the characteristics of neighboring motion vectors, secondly the optimal predictor is chosen from this set to minimize the number of bits used to encode motion vector difference (MVD). An index is then optimally transmitted to signal the selected predictor according to the rules defined in [6] and a boundary-matching which is applied at the decoder side in order to reduce predictor signalization cost. A similar scheme is proposed in [7] without consuming any additional bits to inform the choice of motion vector predictor to the decoder. It slightly improves MV-Comp yet suffers from a decrease of the motion vector prediction quality. However, none of these methods manage to significantly improve upon MV-Comp and to tackle its inherent signalization cost problem.

In [8], a new Inter coding mode, using quantized motion vectors, is introduced. Competing with other modes, it takes care of better distributing bits ressources between texture and residual motion vectors. However, this scheme is limited by the optimal choice of the quantization step used for the motion vector coding and its signalization.

Finally, [9] addresses both the MVP and the MVD coding. The predictor is firstly adaptively coded with additional bits based on the distribution of reference vectors. Secondly, an appropriate code table for the MVD is adaptively selected out of a couple of predefined tables. The latter approach seems promising as it tries to adapt the table for small or large MVDs.

III. PROPOSED SCHEME: MV-FMAP

A. Scheme Overview

The main idea of the scheme comes on the one hand from the observation of the high correlation of the motion vectors between successive frames and blocks, and on the other hand from the choice to correct a posteriori the error prediction of the motion vector without transmitting any signalization index.

Thanks to an additionnal step of motion information observation and forecast, the scheme proposes to efficiently map motion vector residuals before entropy coding in order to improve their coding. The scheme is divided in three steps presented in figure 1. Motion information is first gathered from previous and current frames; it is secondly processed in order to have a more acurate forecast of the information. A list of expected motion vector residuals and a list of unexpected residuals are infered for each block. Then, a mapping of the

motion vector residuals is performed: the forecasted residuals are distributed to residuals positions of lower coding cost.
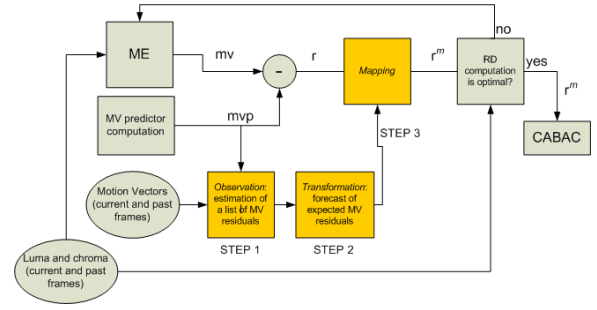

Figure 1.   Proposed scheme overview.

This mapping is performed during the motion estimation and is computed at the decoder side: no side information is necessary. The three steps are explained in the following parts.

B. Step 1 : Construction of the Motion Information

Ideally, we wish to know the list of the motion vectors selected for the prediction of the current frame. This infomation is of course not available: we consequently gather observed motion information. We define D_{mv} and D_r the two definition domains, respectively, of motion vectors and of residuals, such as $D_{mv} \subset \mathbb{Z}^2, D_r \subset \mathbb{Z}^2$.

A *histogram (h)* of the motion vectors previously encoded is computed for each block. h gives the frequency of each motion vector, $mv = (mv_x, mv_y) \in D_{mv}$. It is computed from the n previously coded frames and also the previously coded blocks of the current frame. For the ith block, the frequency of each motion vector is noted h_i, $h_i(mv) \geq 0$. The related frame and block positions of each motion vector are also stored.

h is initialized as follows:

$$h_0(mv) = 0, \forall mv \in D_{mv}. \tag{1}$$

The h update for the $i+1^{th}$ block is performed as follows:

$$h_{i+1}(mv^{\mathrm{i}}) = h_i(mv^{\mathrm{i}}) + 1, \tag{2}$$

where mv^{i} is the motion vector of the ith block.

For each current block to encode, a motion vector residual $r = (r_x, r_y) \in D_r$, is associated to each vector mv of h:

$$r = mv - mvp^i, \tag{3}$$

where mvp^i is the motion vector predictor of the ith block. The *histogram* of the residuals r are consequently defined by:

$$h_i(r) = h_i(mv). \tag{4}$$

Figure 2 gives an example of the residuals distribution for a given block. The red window is the motion estimation window centered on the motion vector predictor of the current block, the residuals with a non-null frequency are the green and red points (including the motion vectors of the neighbours

blocks). The red points, included in the search window, are the residuals which will be considered for the following steps of the scheme, we note $SW \subset D_r$ the search window.
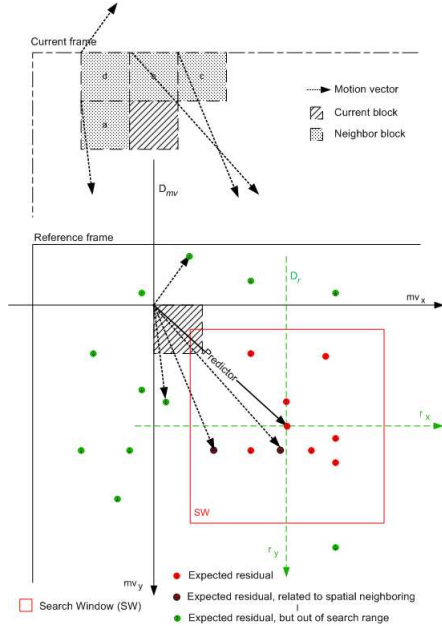

Figure 2. Example of a distribution of observed residuals for a given block.

At this stage of the process, only observed data (the motion vector residuals) has been merged. In order to make this information more representative of what can be expected, some modifications, detailed in the next section, are performed.

C. Step 2 : Transformation of the Motion Information

1) Derivation of the Confidence Interval: The motion information gathered in the first step is a raw information corresponding only to an observation. A refinement of this observation is needed to ensure the efficiency of the scheme, taking into account motion estimation characteristics. Consequently, the second step performs a forecast of the motion vector residuals which will be used in the third step. From the observation previously issued, a confidence degree (d) is associated to each motion vector residual of the search window, reflecting how much this residual is expected to be the final residual resulting from the prediction of the motion vector optimally selected by the encoder. The confidence degree for the ith block is defined by the function C:

$$C : SW \to [0,1] \tag{5}$$
$$r(r_x, r_y) \mapsto d$$

This confidence degree function is a combination of several functions C_k which allow to take into account various characteristics of each residual. They are defined below:

- utilization of a frequency threshold:

$$C_0 : SW \to \{0,1\}$$
$$r \mapsto \begin{cases} 1; & h_i(r) > 0 \\ 0; & h_i(r) = 0 \end{cases} \tag{6}$$

This first confidence degree is set to 1 for all the residuals which have been previously used for at least one block. This function can be seen as an initialization.

- utilization of the reference frame of each residual:

$$C_1 : SW \to [0,1]$$
$$r \mapsto \begin{cases} 1; & r\ related\ to\ the\ previous\ frame \\ \frac{1}{DistF_i(r)}; & otherwise, \end{cases} \tag{7}$$

where $DistF_i(r)$ is the temporal distance from current frame to the frame related to the motion vector residual r. This function allows to favour the motion information of the most recent frame.

- utilization of the reference block of each residual:

$$C_2 : SW \to [0,1]$$
$$r \mapsto \begin{cases} 1; & r\ related\ to\ spatial\ or\ temporal \\ & neighboring\ blocks \\ \frac{1}{DistB_i(r)}; & otherwise, \end{cases} \tag{8}$$

where $DistB_i(r)$ is the spatial distance between the current block and the block related to the residual r.

- utilization of a spatial dilation:

$$C_3 : SW \to \{0,1\}$$
$$r \mapsto \begin{cases} 1; & r \in V_{r'}, \forall r^{'} \in SW \mid C_0(r') \times \lfloor C_2(r') \rfloor = 1 \\ 0; & otherwise, \end{cases} \tag{9}$$

where $V_{r'}$ corresponds to the 8 neighbors (8-connected) of r', and $\lfloor . \rfloor$ is the integer part operator. This dilation is performed to refine the forecast by modifying the confidence degree of residual positions in the neighborhood of observed residuals. Indeed, the H.264/AVC motion estimation is performed at quarter pel precision. Even in case of a constant motion, it often occurs that the motion vector slightly fluctuates on the ¼ pel scale. Thus, a spatial dilation is performed on the expected residuals which belongs to spatial and temporal neighboring blocks.

2) Forecast Motion Information: We build three forecast motion information lists (*FMI*), which are defined by different combination of these functions. Each resulting confidence degree is applied to the observed residuals defined in III-B. The resulting lists are defined below:

- *FMI-a*: the confidence degree is defined by:

$$C_a(r) = C_0(r) \times \lfloor C_1(r) \rfloor . \tag{10}$$

The *FMI-a* list corresponds to all residuals derived from the previous frame and previously encoded blocks of the current frame. It is based on a *global* motion information.

- *FMI-b*: the confidence degree is defined by:

$$C_b(r) = \begin{cases} C_a(r); & C_a(r) > 0 \\ C_3(r); & otherwise. \end{cases} \tag{11}$$

The *FMI-b* list corresponds to *FMI-a* extended by the addition of residuals obtained by dilation of the 4 residuals of the spatial neighbours blocks and the 9 residuals of the colocated block and its 8 neighbours. These residuals dilations allow to favorize the local information.

- *FMI-c*: the confidence degree is defined by:

$$C_c(r) = \begin{cases} C_a(r) \times \lfloor C_2(r) \rfloor; & C_a(r) \times \lfloor C_2(r) \rfloor > 0 \\ C_3(r); & otherwise. \end{cases} \tag{12}$$

The *FMI-c* list corresponds to the residuals of only the spatial and temporal neighbours blocks and the dilated residuals defined above. It creates a *local* FMI.

D. Step 3 : Mapping of the Residuals

The aim of the third step of the proposed scheme is to define a new distribution of the expected and non-expected residuals. Thus, a mapping table, Map is defined:

$$Map : SW \to SW$$
$$r(r_x, r_y) \mapsto r^m(r_x^m, r_y^m) \tag{13}$$

From the two previous steps we can define two residuals lists, L_0 and L_1, such as $SW = L_0 \cup L_1$ and where:

- L_0 is the list of the expected residuals, initialized as:

$$L_0 = \{r \in SW \mid C(r) > 0\}$$

- L_1 is the list of the non-expected residuals, initialized as:

$$L_1 = \{r \in SW \mid C(r) = 0\}.$$

Another list is defined for the target residuals, $L_m = \{r^m \in SW\}$ at the initialization. For the equation (13), we consequently have $r \in L_0$ or $r \in L_1$ and $r^m \in L_m$.

The mapping table differs for each block and is built in two main steps which upload the three lists during all the process: *step 1* - the expected residuals (L_0) are mapped to available target residuals (L_m) of lower coding cost; *step 2* - the non-expected residuals (L_1) are mapped to remaining target residuals (L_m), potentially of higher coding cost. In this second step, the aim is to minimize the damages induced. For each step, two parameters are involved: the processing order, and the mapping conditions.

For the processing order, the residuals can be ordered by their confidence degree or, as we present in this paper, according to a diamond pattern which assumes that the most probable vectors remain those localy accurately predicted and which are therefore close to the search window center.

Two mapping conditions on the values of the target residuals are defined. For each residual, the question arises to redistribute or not a residual to a more favorable target residual (during step 1), or to the less unfavorable target residual (during step 2). We first introduce the mapping conditions $\triangle_i, i \in \{0, 1\}$, respectively for the lists L_0 and L_1:

$$\triangle_i : \begin{cases} r = (r_x, r_y) \in L_i, \ r^m = (r_x^m, r_y^m) \in L_m \\ r^m = \min_{\forall r^{m'} \in L_m} \left(\left| r_x^{m'} \right| + \left| r_y^{m'} \right| \right) \end{cases} \tag{14}$$

This condition ensures that the distance from the target residual to the search window center is minimal.

The second mapping condition is defined by:

$$\triangle_2 : \begin{cases} r = (r_x, r_y) \in L_0, \ r^m = (r_x^m, r_y^m) \in L_m \\ |r_x| - |r_x^m| \geqslant 0 \ and \ |r_y| - |r_y^m| \geqslant 0 \end{cases} \tag{15}$$

This condition takes care of the entropy coder characteristic by avoiding the mapping of an expected residual on a target residual which belongs to an opposite quadrant of the SW, assuming SW is divided in four quadrants. Indeed, this case would end up with an expensive coding of the residual instead of the expected decrease because of the probability model chosen by the encoder.

Finally, the overall process of the mapping is given below:

- *Initialization*: the lists L_0 and L_1 are ordered following a diamond pattern from the center of the search window to its edges. The mapping table Map is empty.
- *Step 1*: Following the defined diamond pattern order, one processes each residual $r \in L_0$ and maps the target residual $r^m \in L_m$ which verifies the conditions below:

$$Map(r) = r^m \Leftrightarrow \begin{cases} \triangle_0 \\ \triangle_2 \end{cases} \tag{16}$$

After each residual mapping, the two lists are updated:

$$L_0 \longleftarrow L_0 \setminus r \ and \ L_m \longleftarrow L_m \setminus r^m \tag{17}$$

- *Step 2*: In the same way, each residual $r \in L_1$ maps the remaining target residual $r^m \in L_m$ through the condition:

$$Map(r) = r^m \Leftrightarrow \triangle_1 \tag{18}$$

After each residual mapping, the two lists are updated:

$$L_1 \longleftarrow L_1 \setminus r \ and \ L_m \longleftarrow L_m \setminus r^m \tag{19}$$
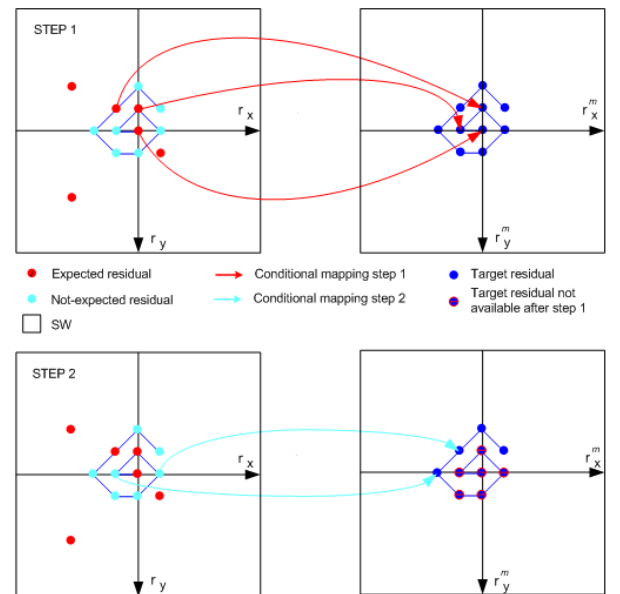

Figure 3.   Example of the two steps of a diamond mapping.

Figure 3 gives a mapping example, the residuals are processed through a diamond pattern centered on the $(0,0)$ position and following the two main steps described above. The figures on the left show the expected residuals of the L_0 list in red and the non-expected residuals of the L_1 list in cyan. The figures on the right show the target residuals of the L_m list ordered through the diamond pattern. The red and blue arrows respectively correspond to the conditional mapping defined by equation (16) and (18).

IV. EXPERIMENTAL RESULTS

A. Experimental Settings

The proposed coding scheme has been implemented and tested in the reference software JM 11.0 KTA 2.4r1. In this section we compare our method to the following configuration of the reference codec: IPP scheme (1 I and 48 P), 1 reference frame, 64×64 search window, EPZS FastFull Search and CABAC entropy coding. The set of sequences tested in this paper corresponds to the JCT-VC test set extended by 4 sequences exhibiting large gains with MV-FMap technique, due to their motion difficult to predict. All these sequences are encoded for the 4 quantization parameters (QP) defined by the JCT-VC Call for Proposals.

Three configurations of MV-FMap have been tested, corresponding to the three different Forecasted Motion Information (*FMI*) lists: *FMI-a*, *FMI-b* and *FMI-c*.

The bitrate savings given in percent are calculated with the Bjontegaard metric [10], recommended by the VCEG, which is an average difference between two RD curves [11].

B. Results

Table I gives the percentage bitrate savings for the three proposed configurations of MV-FMap. For all sequences, the proposed scheme gives a systematic bitrate saving except with *FMI-a* for the *RaceHorses QWVGA* sequence (-0.3%). The bitrate saving reaches 3.5%, 3.0% and 4.9% respectively for *BQTerrace 1080p* with the *FMI*-a and *FMI-b* configurations and finally *Nuts5 WVGA* with the *FMI-c* configuration. This sequence belongs to a class of sequences with a high local motion activity and with a simple background which is adapted for the local approach while the sequence *BQTerrace 1080p* corresponds to a slow camera movement and temporally continuous which is adapted for the global approach. Moreover, the sequence *BQSquare QWVGA*, which gives the lower bitrate savings in average with 0.4%, 0.1% and 0.3% respectively, is a sequence with a simple motion which is already well predicted by the H.264/AVC motion vector predictor. The best average results are obtained for the *FMI-c* configuration which reaches 2.3% bitrate savings in average compared to the *FMI-a* and *FMI-b* configurations which provide respectively 1.1% and 1.4%. This result confirms that the second step of the scheme is a key point of MV-FMap. Indeed, the *FMI-a* and *FMI-b* configurations suffer from a too high number of motion vector residuals gathered in the motion information list with several motion vectors which are not correlated with the current block motion. Especially for the sequences with an high

motion activity like *RaceHorses QWVGA*. This observation is confirmed by the 2.6% gap between the bitrate saving for this sequence for the *FMI-a* and the *FMI-c* configurations.

	FMI-a	FMI-b	FMI-c
BasketBallPass QWVGA	0.2	0.6	1.4
BlowingBubbles QWVGA	0.5	0.6	1.1
BQSquare QWVGA	0.4	*0.1*	*0.3*
RaceHorses QWVGA	*-0.3*	1.1	2.3
BasketBallDrill WVGA	1.5	2.2	3.0
BQMall WGA	0.7	1.2	3.0
PartyScene WVGA	0.7	1.0	2.1
RaceHorses WVGA	0.5	1.5	3.1
Vidyo1 720p	1.6	1.4	2.7
Vidyo3 720p	0.1	0.4	0.9
Vidyo4 720p	1.5	1.9	3.1
BasketBallDrive 1080p	1.3	2.1	2.2
BQTerrace 1080p	**3.5**	**3.0**	1.1
Cactus 1080p	0.4	0.7	2.3
Kimono1 1080p	1.1	1.5	2.0
ParkScene 1080p	0.7	1.3	1.7
Foreman CIF	2.0	1.9	2.7
Ice CIF	0.5	1.6	2.8
Nuts5 WVGA	2.4	2.6	**4.9**
Jets 720p	2.1	2.3	3.7
Average	**1.1**	**1.4**	**2.3**

Table I
PERCENTAGE BITRATE SAVINGS FOR THREE MV-FMAP CONFIGURATIONS.

Finally, the *FMI-c* configuration gives the best results which are explained by the utilization of the motion information which is the most spatially and temporally close to each block to be coded. However, an adaptation to the content could be also profitable for the proposed scheme since it could enable to collect the appropriate number of motion vectors. In the following section, we focus on this best configuration to analyse in depth the scheme efficiency.
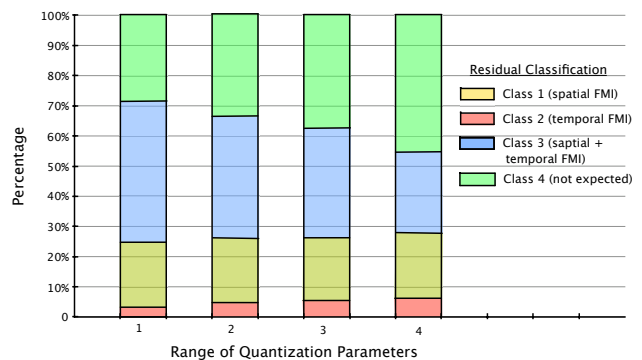
C. Results Analysis

Figure 4. Classification of the motion vector residuals (in percentage) for each range of QP, in average for the test set.

Let us first study the origin of each encoded residual. This *residual origin* is sorted in four classes:

- Class 1: residual only derived from a spatial motion information. The encoded residual belongs to one of the spatial neighboring blocks,

210

- Class 2: residual only derived from a temporal motion information. The encoded residual belongs, in the previous frame, to one of the collocated block and its neighboring,
- Class 3: residual derived from both spatial and temporal motion information,
- Class 4: residual which are non expected. The encoded residual is not in the *FMI*.

Figure 4 gives the origin of encoded motion vector residuals in percentage for four ranges of quantization parameters from high (1) to low bitrate (4). This proportion has been computed in average for all the proposed sequences. In average, the proportion of encoded residuals which are forecasted represents 63% of the cases. This proportion reaches 71% for high bitrate (QP range 1). Among the expected residuals, we observe a constant proportion of encoded residuals which only come from Class1 (22%), the residuals which only come from Class 2 are the lowest in proportion, but this number increases with the QP. Finally, the residuals which belong to the Class 3 are the most selected. These observations confirm the efficiency of the scheme which enables to forecast in most of the cases the encoded residual and thus to reduce its coding cost, especially at low bitrate. This forecast is performed by an efficient combination of spatial and temporal motion information.
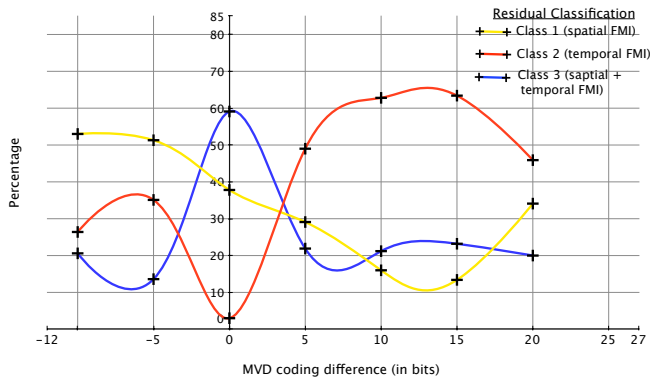

Figure 5. Classification of the motion vector residuals (in percentage) and corresponding coding difference (in bits), in average for the gain test set.

For the cases where the encoded motion vector residuals were expected, Figure 5 gives the origin of these residuals (in percentage) and the difference of corresponding coding cost (in bits). The positive values report a bits saving of the proposed scheme compared to the reference. These results have been computed in average for all the proposed sequences. The three curves respectively correspond to the residuals which belong to the classes 1, 2 and 3. For the case where the residual coding cost is equivalent for the reference scheme and MV-FMap, we observe that the origin of the motion vector residuals is in majority from the Class 3 (60%) and from the Class 1 (38%). Indeed, this corresponds to a case of small motion activity where the motion information is made of similar motion vectors and the H.264/AVC motion vector predictor is efficient. Secondly, for the cases where MV-FMap induces bit saving, the residual origin is mainly

temporal (55% in average). Indeed, MV-FMap takes advantage from the mapping of temporal residuals which compensate some prediction errors of the H.264/AVC median by using the temporal redundancies. Moreover, MV-FMap is highly efficient as soon as the predictor is formed by diverging motion vectors. This is why the spatial motion information provides bit savings, especially high savings (>15 bits). Finally for the losses cases, the residuals mostly belong to the Class 1 (52% in average). This observation highlights the difficult compromise of the number of motion vector residuals gathered in the motion information list (*FMI*) and their origins.

V. CONCLUSION

In this paper, an original approach to improve motion vector coding is presented. The aim of this approach is to exploit the high correlation of the motion vectors between successive frames and blocks to correct the error prediction of the motion vectors before entropy coding, without transmitting any signalization index, by modifying the residuals distribution to privilegiate the forecasted residuals. The proposed scheme is divided in three steps which respectively collect a motion information list, turn this motion information into a confidence degree, and modify the residuals distribution in order to reduce the coding cost of each forecasted residuals.

The reported results compare three configurations which correspond to three different motion information list ranging from global to local only. The local configuration (*FMI-c*) is the most efficient with 2.3% (up to 4.9%) bitrate savings in average on a large set of sequences. A careful study of the results obtained for each configuration tends to show that room remains for further improvement of the scheme.

REFERENCES

[1] "Advanced video coding for generic audiovisual services", version 3. ITU-T Recommendation H.264, ISO/IEC 14496-10 AVC, 2005.
[2] K. Suehring. H.264/AVC and KTA software coordination. Available: http://iphome.hhi.de/suehring/.
[3] G. Laroche, J. Jung, and B. Pesquet-Popescu, "RD Optimized Coding for Motion Vector Predictor Selection", IEEE Trans. on CSVT, vol. 18, pp.1247-1257, Sep, 2008.
[4] K.P. Lim, G. Sullivan, and T. Wiegand, "Text Description of JM Reference Encoding Methods and Decoding Concealment Methods", JVT-N046 contribution, Hong-Kong, Jan. 2005.
[5] J.Dai, O.C. Au, W. Yang, C. Pang, F. Zou, and Y. Liu, "Motion Vector Coding based on Predictor Selection and Boundary-matching estimation", Proc. of MMSP'09, Brazil, 2009.
[6] S.D. Kim, and J.B. Ra, "An Efficient Motion Vector Coding Scheme Based on Minimum Bitrate Prediction", IEEE Trans. on IP 8, 1117-1120, 1999.
[7] K. Won, J. Yang, and B. Jeon, "Motion Vector Coding using Decoder-side Estimation of Motion Vector", IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting, pp.1-4, 2009.
[8] M. Cagnazzo, M.A. Agostini, M. Antonini, G. Laroche and J. Jung, "Motion Vector Quantization for Efficient Low Bit-Rate Video Coding", in Proc. SPIE, the Int. Soc. for Optical Engineering, vol. 7257, Jan. 2009.
[9] M. Takahashi, and M. Yamaguchi, "H.264/AVC Improvement based on Adaptive Motion Vector Coding", Proc. of the 27th conference on Picture Coding Symposium, USA, 2009.
[10] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD-Curves", ITU-T VCEG, Proposal VCEG-M33, Apr. 2001.
[11] J. Jung and S. Pateux, "An Excel Add-in for Computing Bjontegaard Metric and Its Evolution", ITU-T VCEG, Info. VCEG-AE07, Jan. 2007.