

Motion Vector Coding based on Predictor Selection and Boundary-matching Estimation

Jingjing Dai ^{#1}, Oscar C. Au ^{#1}, Wen Yang ^{#1}, Chao Pang ^{#1}, Feng Zou ^{#1}, and Yu Liu ^{*2}

[#] *Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong*
¹ {jjdai, eeau, eeyangw, pcece, fengzou}@ust.hk

^{*} *MTI/ECE, Applied Science and Technology Research Institute
HK Science Park, Shatin, NT, Hong Kong*
² liuyu@astri.org

Abstract—In most of recent video coding standards based on block-based hybrid coding scheme, especially in the state-of-the-art H.264/AVC standard, motion vector information occupies a considerable portion of the whole compressed bitstream. Therefore, the efficient coding of motion vectors has become an essential objective to further reduce the bitrate. In this paper, we propose a novel motion vector coding method based on predictor selection and estimation. First, the optimal motion vector predictor (MVP) is chosen from a predefined predictor candidate set consisting of both spatial predictors and temporal predictors to minimize the number of bits used for encoding motion vector difference (MVD). Then, in order to avoid sending extra bits for informing the decoder which candidate is selected by the encoder, boundary-matching (BM) estimation is applied at the decoder side to find out the optimal predictor. The basic principle of BM estimation is to preserve the spatial continuity of boundaries between currently reconstructed block and its neighbors. Simulation results show that compared to the original H.264/AVC codec, the proposed scheme improves coding efficiency for various video sequences.

I. INTRODUCTION

In the hybrid block-based video coding scheme, motion estimation and compensation are used to exploit the high temporal redundancy between successive frames; the motion-compensated block from already decoded frames is used as prediction for the currently coded region. In order to obtain the motion-compensated block at the decoder side, motion vectors need to be transmitted by the encoder. The H.264/AVC standard [1] is the most state-of-the-art coding scheme today, and its basic coding structure follows the traditional block-based hybrid coding. Compared to its predecessors, H.264 has achieved a significant bitrate reduction by employing a good many advanced coding tools such as variable block-size inter prediction, 1/4-pel motion estimation, and efficient intra prediction. However, accompanying the overall reduction of bitrate, the proportion of bits occupied by motion vectors in the H.264 bitstream has been increased, which can reach up to 50% at high QP values for some sequences. Therefore, the

efficient coding of motion vectors is of great essence to further improve the compression efficiency of H.264.

Plenty of efforts have been made on motion vector coding, and many different approaches have been presented in the literature. According to [2], these coding methods mainly fall into two categories. The first one is lossy coding, such as [3]. The second one, which is more widespread and will be investigated in this paper, is lossless coding. In [4], Bongsoo Jung et al. propose a new macroblock coding mode of “pooled zero vector coding” as an efficient representation when all 4×4 blocks have zero MVD. In [5], Sung Deuk et al. choose the MVP from three neighbouring motion vectors based on minimum bitrate prediction, and an extra identifier is encoded to indicate the predictor choice. In [2], Guillaume Laroche et al. propose a RD optimized spatio-temporal coding scheme for the prediction of inter motion vectors and skip mode, in which the motion vector selection is based on a modified RD cost and the choice of predictors should be encoded to inform the decoder. In [7], Jungyoun Yang et al. propose a motion vector coding scheme that allows usage of an optimal predictor without consuming additional bits to signal the predictor choice to the decoder. In [6], Juan Liu et al. employ median motion vector and co-located motion vector as two candidates for MVP, and the choice between these two is based on local statistical characteristics. Both of the method in [6] and [7] use more than one predictor candidates, and avoid sending additional information to inform the decoder which candidate is selected at the encoder side. However, the selected predictor is not always optimal in the sense of minimizing the bitrate. In addition, the template matching technique used in [7] is computing-intensive. In this paper, a novel motion vector coding scheme based on predictor selection and BM estimation is proposed. First, to fully exploit both the spatial and the temporal correlation in the motion vector fields, a predictor candidate set composed of several spatial and temporal predictors is defined. After that, the optimal predictor is selected to minimize the bitrate. Then, BM estimation is applied at the decoder side to find out this optimal predictor without additional information for indicating which predictor

MMSp'09, October 5-7, 2009, Rio de Janeiro, Brazil.
978-1-4244-4464-9/09/\$25.00 ©2009 IEEE.

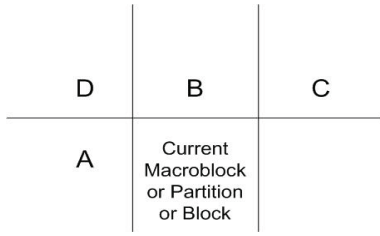


Fig. 1. Neighboring blocks of current block

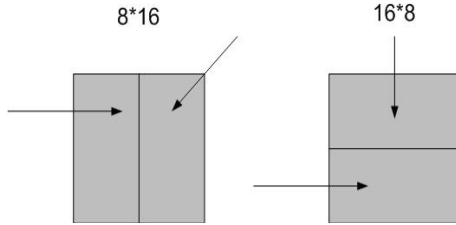


Fig. 2. MVP for P8×16 macroblock and P16×8 macroblock

is chosen by the encoder. The basic idea of the proposed BM estimation comes from the motion vector recovery technique in error concealment aiming at minimizing the total variation between the internal boundary and the external boundary of the currently reconstructed block [9, 10].

The rest of this paper is organized as follows: in Section II, we present the motion vector coding scheme in H.264/AVC; the detailed description of the proposed motion vector coding algorithm based on predictor selection and BM estimation is given in Section III, followed by the experimental results in Section IV; then, Section V analyzes the proposed scheme and discusses the direction for future work; finally, Section VI concludes this work.

II. MOTION VECTOR CODING IN H.264/AVC

In H.264/AVC, the motion vector of current block is coded differentially with respect to a spatial prediction $mv_{H.264}$, which is generated by the motion vectors of spatially adjacent blocks as shown in Figure 1. If the blocks A, B, C or D are coded in intra mode (mode without motion vector), the corresponding motion vector is considered to be zero. Figure 2 illustrates the MVP for macroblocks of type P16×8 and P8×16. For P8×16 macroblock, $mv_{H.264}$ for the left partition is mv_A , and that for the right partition is mv_C ; for P16×8 macroblock, $mv_{H.264}$ for the up partition is mv_B , and that for the bottom partition is mv_A . For macroblocks of other types, if only one of the blocks A, B and C has the same reference frame as current block, the motion vector of this block is used as $mv_{H.264}$. Otherwise, $mv_{H.264}$ is defined as the median of $\{mv_A, mv_B, mv_C\}$, horizontal and vertical components separately. If block C is not available, for example, block C is out of slice boundary, the motion vector of block D is used instead. After $mv_{H.264}$ is obtained, the MVD is calculated as the difference between the motion vector of current block and this predictor, and then written into the bitstream through entropy coding.

$mv_{H.264}$ is very simple and can work well for sequences with a relatively uniform motion. Unfortunately, for sequences with fast or complex motions, $mv_{H.264}$ is not always the promising one for minimizing the number of coding bits of MVD.

III. PROPOSED ALGORITHM

In this section, we present the proposed motion vector coding algorithm in details. For lossless coding of motion vectors, MVP has a significant effect on the coding efficiency. If there are more choices for MVP, it is very possible that we can find a predictor which has a closer correlation with current motion vector compared to $mv_{H.264}$, and thus fewer bits are needed to represent the MVD. Based on this argument, our proposed scheme defines a predictor set which is composed of multiple predictor candidates. Then, the best predictor is chosen out of this set to minimize the number of bits dedicated to MVD. To further reduce the bitrate, we apply BM estimation to find out the chosen predictor at the decoder side so that the extra bits needed to transmit the choice of predictor candidate can be saved.

A. Predictor candidate set and selection scheme

In video sequences, the motion of an object has high correlation in both the spatial domain and the temporal domain [8]. Thus, in order to fully exploit the correlation in motion vector fields so as to improve the accuracy of MVP, both temporal predictor and spatial predictor must be included in the predictor candidate set. In the proposed scheme, we choose the motion vector at the same position in the previous frame mv_{col} as the temporal predictor. If the current block and its co-located block belong to the same object, which is very frequent for most video sequences, there is a high possibility that they undergo a similar motion. The spatial predictors consist of $mv_{H.264}$, and mv_A . If any one of the neighboring block A, B and C does not belong to the same object with the current block, its motion vector may distract the H.264 median predictor away from the real motion vector of the current block, which leads $mv_{H.264}$ to be a false prediction. In order to overcome this defect, mv_A , mv_B and mv_C should be separately used as predictor candidates. However, in order to reduce the extra information bits as well as guarantee the efficiency of BM estimation (this will be discussed in Section V), we only use mv_A in the proposed scheme. Also, the zero motion vector mv_{zero} is included for better prediction of static background region in video. So the predictor candidate set is $\{mv_{H.264}, mv_A, mv_{col}, mv_{zero}\}$, and the four candidates are indicated by pmv_i , $i = 0, 1, 2, 3$.

The target of predictor selection is to choose the predictor that minimizes the number of bits consumed by MVD. Therefore, the best predictor pmv_{best} is defined as follows:

$$pmv_{best} = \arg \min_{pmv_i} r(mv - pmv_i), i \in \{0, 1, 2, 3\}, \quad (1)$$

where $r(\cdot)$ is the function used to measure the number of bits for encoding MVD, and mv represents the motion vector of current block.

B. Predictor estimation based on boundary-matching

This section gives a detailed description of the proposed BM estimation, with which we can find out pmv_{best} at the decoder side to avoid sending extra bits for specifying the candidate index of pmv_{best} (idx_{best}). The basic idea of BM estimation originates from the technique of recovering motion vector in error concealment. In decoding process, the knowledge of idx_{best} is required in order to reconstruct the motion vector. Thus, additional bits are needed to transmit this information, which attenuates the benefit of using a more accurate predictor. BM estimation is designed to deal with how to detect pmv_{best} automatically so that those additional bits can be saved.

The proposed BM estimation seeks pmv_{best} based on the spatial continuity between current block and its neighbors. In the process of decoding, if a predictor other than pmv_{best} is used to reconstruct the motion vector, there is a high chance that the corresponding reconstructed block is not consistent with its neighbors. In other words, if the correct predictor is used, the reconstructed block can agree with the neighbors. So the target of BM estimation is to find out the predictor that makes the reconstructed block match its neighbors best. The extent of such matching is measured by the boundary matching error (BME), which represents the match distortion between the internal and external boundaries of the currently reconstructed block. As shown in Figure 3, internal boundaries stand for the boundary pixels of current block while external boundaries stand for the surrounding pixels in the spatially neighboring blocks; the letters N, W, E, S are short for North, West, East and South. BME is defined as the sum of absolute differences (SAD) between the internal boundary and the external boundary:

$$\begin{aligned}
 BME &= \omega_N \sum_{k=0}^{N_w-1} |p_{rec}(x+k, y) - p_{rec}(x+k, y-1)| \\
 &+ \omega_W \sum_{k=0}^{N_h-1} |p_{rec}(x, y+k) - p_{rec}(x-1, y+k)| \\
 &+ \omega_S \sum_{k=0}^{N_w-1} |p_{rec}(x+k, y+N_h-1) \\
 &- p_{rec}(x+k, y+N_h)| \\
 &+ \omega_E \sum_{k=0}^{N_h-1} |p_{rec}(x+N_w-1, y+k) \\
 &- p_{rec}(x+N_w, y+k)|, \quad (2)
 \end{aligned}$$

where N_w , N_h are the width and height of current block, respectively; (x, y) indicates the position of the most top-left pixel and p_{rec} corresponds to the reconstructed value. ω_N , ω_W , ω_S , ω_E denote the weighting coefficients for North, West, South and East boundary. $\omega_N=1$ if the north neighboring block is available, otherwise $\omega_N=0$. So are the definitions of ω_W , ω_S , and ω_E .

At the decoder side, for every predictor candidate pmv_i , the corresponding motion vector mv_i can be obtained by

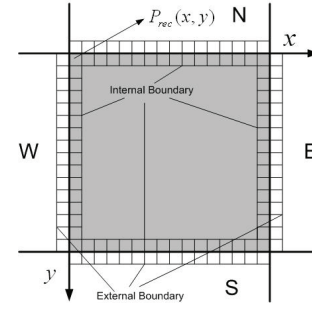


Fig. 3. Illustration of internal boundaries and external boundaries

adding pmv_i to the decoded MVD. Then, the pixel values of internal boundaries can be reconstructed. To differentiate the generated reconstruction of internal boundaries relevant to different predictor candidates, the reconstructed value that corresponds to pmv_i is denoted by p_{rec_i} . When encoding and decoding current block, only the pixels in the left and up region have been reconstructed, thus, only the information of west and north external boundaries can be made use of in our scheme. Therefore, the BME related to pmv_i , represented by BME_i , is calculated by

$$\begin{aligned}
 BME_i &= \sum_{k=0}^{N_w-1} |p_{rec_i}(x+k, y) - p_{rec}(x+k, y-1)| \\
 &+ \sum_{k=0}^{N_h-1} |p_{rec_i}(x, y+k) - p_{rec}(x-1, y+k)|. \quad (3)
 \end{aligned}$$

According to the target of BM estimation mentioned above, the predictor that has the minimum BME is assumed to be the estimated optimal predictor pmv_{est} , that is,

$$pmv_{est} = \arg \min_{pmv_i} BME_i, i \in \{0, 1, 2, 3\}. \quad (4)$$

If pmv_{est} satisfies $pmv_{est} = pmv_{best}$, it means that the decoder can detect the predictor selected by the encoder automatically. In this case, the encoder does not need to consume bits to transmit idx_{best} . Since such estimation procedure can also be performed at the encoder side, the encoder can recognize whether the decoder can find pmv_{best} or not.

C. Syntax for extra signaling information

This section is to introduce the syntax for extra signaling information, which includes idx_{best} and a 1-bit flag $flag_{est}$. $flag_{est}$ is to signal the decoder whether to apply BM estimation to seek pmv_{best} or not. At the encoder side, after the encoder selects pmv_{best} according to equation (1) and calculates the MVD, it works as follows:

- Step 1: Check if all the predictor candidates are identical. If so, go to Step 4; otherwise, go to Step 2.
- Step 2: Check if the decoder can estimate pmv_{best} by BM estimation. If can, go to Step 3(a); otherwise, go to Step 3(b).

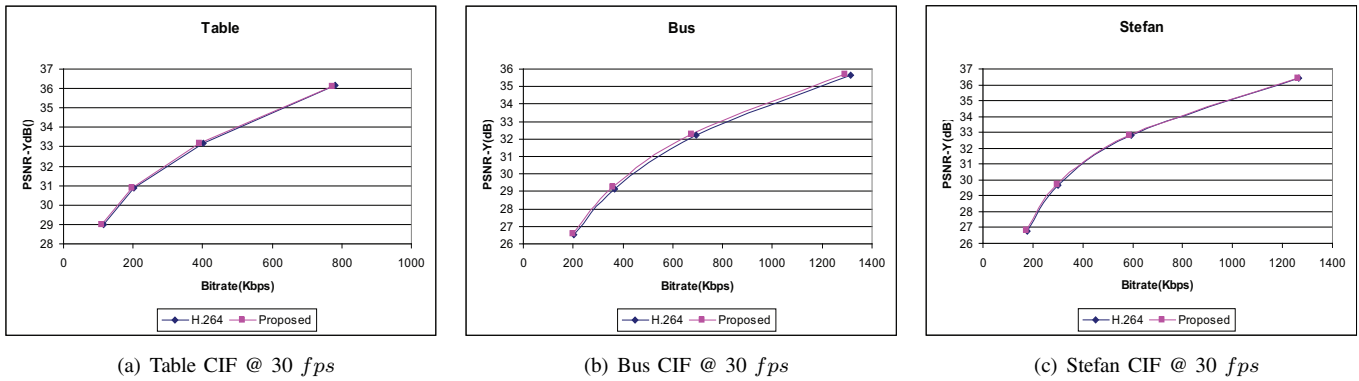


Fig. 4. Rate Distortion Performance Comparison

- Step 3(a): Set $flag_{est}$ to 1, and encode $flag_{est}$. Go to Step 4.
- Step 3(b): Set $flag_{est}$ to 0, and encode $flag_{est}$ and idx_{best} . Go to Step 4.
- Step 4: Encode MVD.

Accordingly, the decoding process is described below:

- Step 1: Check if all the predictor candidates are identical. If so, use the first predictor candidate as MVP, decode MVD and go to Step 4; otherwise, go to Step 2.
- Step 2: Decode $flag_{est}$. If $flag_{est}$ equals to 1, go to Step 3(a); otherwise, go to Step 3(b).
- Step 3(a): Decode MVD, and perform BM estimation to get the MVP. Go to Step 4.
- Step 3(b): Decode idx_{best} and MVD, and use $pmv_{idx_{best}}$ as the MVP. Go to Step 4.
- Step 4: Reconstruct the motion vector by adding MVP to MVD.

When all the predictors in the candidate set are the same, there is only one choice for MVP, so the encoder does not need to send any information about the predictor choice, and the decoder just chooses the first predictor candidate as MVP.

IV. EXPERIMENTAL RESULTS

For the evaluation of the performance, the proposed method is implemented on H.264/AVC KTA (Key Technical Area) reference software KTA 2.2 [11] (KTA software is developed for next generation standard based on the framework of jm11.0). The coding efficiency of the proposed method is compared with that of the H.264/AVC standard @ Baseline profile. As recommended in [12], we select the prediction structure IPPP, a search range of 32×32 , 4 reference frames, and RD-optimization On. The test sequences are Foreman, Bus, Stefan, Silent, and Table (CIF sequences, @ 30 fps). Four quantization parameters (QPs) of 28, 32, 36, and 40 are used. For Baseline profile, CAVLC entropy coding is used, thus the extra signalling information (idx_{best} and $flag_{est}$) introduced in our scheme is encoded simply by binary coding. Since four predictor candidates are defined, 2 bits are needed to encode idx_{best} .

For a numerical comparison, the performance of the proposed method is evaluated in terms of the Bjontegaard Delta

bit rate (BDBR) and the Bjontegaard Delta PSNR (BDPSNR) [13]. Those quantities give the average bit rate and PSNR differences over several QP values of two RD curves. Table I depicts the BDBR and BDPSNR of the proposed method compared to the H.264/AVC standard, from which we can find that the proposed method offers a compression gain for all the test sequences by an average bitrate reduction of 2.1% with an equivalent or even better PSNR quality. The proposed method reduces the bit rate especially for those video sequences with fast or complex motion fields, such as Sequence Bus and Table, because of more precise predictor. Also, the motion information in these sequences consumes a relatively large proportion of the total bits, which makes the improvement of motion vector coding more distinctive. The bitrate saving for Silent sequence is relatively small due to the fact that this sequence is mainly composed of a large object with small motions, in which case coding bits for motion vectors occupy a small portion of the total bits and at the same time, the spatial median predictor can work well. In fact, the percentage of the selection of $mv_{H.264}$ is much greater than that of other candidates in Silent sequence. For example, when QP equals to 28, $mv_{H.264}$ is selected about 60% of the time. Thus, predictor choices other than $mv_{H.264}$ can only give little improvement.

TABLE I
CODING PERFORMANCE COMPARISON

Sequence	BDPSNR (dB)	BDBR (%)
Foreman	0.05	-1.2
Bus	0.19	-3.8
Stefan	0.08	-1.7
Silent	0.04	-1.0
Table	0.1	-2.6
Average	0.09	-2.1

Figure 4 exhibits the rate-distortion performance for sequence Table, Bus and Stefan. In this figure, the efficiency of the proposed method is illustrated for every rate point. In addition, the proposed method has better performance especially when bit rate is low, which may be explained by the fact that motion vector information will take a significant part of the whole bitstream at low bit rates.

V. DISCUSSION

For lossless coding of motion vectors, the efficiency is closely related to the choice of MVP. The H.264 standard uses only one spatial predictor, which is not always optimal in minimizing the number of bits for MVD. If more predictors are available, it is likely that a more accurate prediction can be obtained. Hence, multiple predictor candidates are defined in our algorithm. The number of candidates can affect the coding efficiency to a great extent. By using more predictors, there is a higher likelihood that it requires fewer bits to encode MVD. However, when there are more choices, it becomes harder to discover pmv_{best} at the decoder side. In addition, when BM estimation fails, more bits are required to transmit idx_{best} . For compromise, we choose four predictors, and simulation results shown above demonstrate that this configuration can give good results.

The four predictors used in our scheme are $mv_{H.264}$, mv_a , mv_{col} and mv_{zero} , which are chosen to exploit both the spatial correlation and temporal correlation of motion vector fields. In fact, an adaptive set of predictors according to local statistical characteristics is expected to increase the gain. For example, in our method, mv_a is included as a candidate to compensate for the case that any one of block A, B and C does not belong to the same object with current block. Improvement is possible if mv_a , mv_b and mv_c can all be chosen as the predictor candidate and to choose which one is adaptively based on their similarities with $mv_{H.264}$.

When more than one predictor candidates exist, the encoder needs to inform the decoder which candidate is chosen. However, the extra bits needed will consume a considerable number of bits. If the decoder can estimate the choice of predictor candidates automatically, such bits can be saved. To estimate the predictor at the decoder side, many characteristics can be made use of, such as the smoothness of motion vector fields, the spatial continuity of current block's internal boundaries and external boundaries, the temporal continuity of current block's external boundaries and the reference block's external boundaries, etc. In our scheme, the spatial continuity is considered. To further increase the detection rate, more factors such as those mentioned above can be taken into account.

Last, let us give a simple illustration of the effect of decoder side predictor estimation on the bitrate reduction. Assume the total number of blocks is N_{total} , and the number of blocks whose pmv_{best} can be detected by the decoder using BM estimation is N_{est} . The detection rate DR is defined as N_{est}/N_{total} . If there is no decoder side predictor estimation, 2 bits are needed for each idx_{best} , thus the total number needed is $2N_{total}$. If the decoder side predictor estimation is enabled, the total number becomes $N_{est} + 3(N_{total} - N_{est})$, which is reduced by $(DR - 0.5)$. Therefore, the introduction of decoder side estimation scheme can lead to a bitrate reduction when DR is larger than 50%, which is the general case that the average value of DR in our simulation is about 70%. Since idx_{best} consumes a considerable proportion (on average 7%

to 8%) of the whole bitstream, decoder side estimation can contribute a bitrate reduction of about 1.5%.

VI. CONCLUSION

In this paper, a novel motion vector coding method based on predictor selection and boundary-matching estimation is proposed. First, the best predictor is selected from a predefined candidate set to minimize the bitrate. Then, boundary-matching estimation is applied for the decoder to find out the selected predictor. Based on whether the decoder can get the correct predictor or not, the encoder chooses to send different information.

Experimental results show that the proposed method can reduce the bitrate compared to H.264/AVC. In particular, the proposed method is effective for sequences with fast or complex motions.

ACKNOWLEDGMENT

This work has been supported by the Hong Kong Applied Science and Technology Research Institute (ASTRI) in the Future Multimedia Standards Project (ART/037).

REFERENCES

- [1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," Doc. JVT-G050r1, Mar. 2003.
- [2] G. Laroche, J. Jung, and B. Pesquent-Popescu, "RD optimized coding for motion vector predictor selection," IEEE Trans. Circuits Syst. Video Technol., vol. 18, pp. 1247-1257, Sep. 2008.
- [3] Y. Y. Lee, and J. W. Woods, "Motion vector quantization for video coding," IEEE Trans. Image Process., vol. 4, pp. 378-382, Mar. 1995.
- [4] B. Jung, and B. Jeon, "Pooled zero vector coding for enhanced compression of motion vectors," in Proc. IEEE APCCAS, 2008, pp. 1743-1746.
- [5] S. D. Kim, and J. B. Ra, "An efficient motion vector coding scheme based on minimum bitrate prediction," IEEE Trans. Image Process., vol. 8, pp. 1117-1120, Aug. 1999.
- [6] J. Liu, B. Feng, Z. Ma and W. Liu, "Adaptive motion vector prediction based on spatiotemporal correlation," in Proc. IEEE WiCom, 2006, pp. 1-4.
- [7] J. Yang, K. Won, and B. Jeon, "Motion vector coding with selection of an optimal predictive motion vector," SPIE Opt. Eng. Lett., vol. 48, Jan. 2009.
- [8] Iain E. G. Richardson. H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedias, UK: John Wiley, 2003
- [9] W. M. Lam, A. R. Reibman and B. Liu, "Recovery of lost or erroneously received motion vectors," in Proc. IEEE ICASS, 1993, vol. 3, pp. 417-420.
- [10] Y. K. Wang, W. M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabbouj, "The error concealment feature in the H.26L test model," in Proc. IEEE ICIP, 2002, pp. 729-732.
- [11] K. Suehring, "H.264/AVC Software Coordination" [Online]. Available: <http://iphome.hhi.de/suehring/tml/>
- [12] T. K. Tan, G. J. Sullivan, and T. Wedi, "Recommended simulation common conditions for coding efficiency experiments" ITU-T VCEG, Berlin, Germany, 2008, Input/Discussion VCEG-A110.
- [13] Calculation of Average PSNR Differences between RD-Curves, ITU-T VCEG, 2001, Proposal VCEG-M33.