Multihypothesis Prediction using Decoder Side Motion Vector Derivation in Inter Frame Video Coding

Steffen Kamp, Johannes Ballé, and Mathias Wien

Institut für Nachrichtentechnik, RWTH Aachen University, Germany
{kamp,balle,wien}@ient.rwth-aachen.de

ABSTRACT

In this paper, a multihypothesis prediction scheme for inter frame video coding is proposed. Using a template matching algorithm, motion vectors are derived at the decoder side instead of explicitly coding the motion vectors into the bitstream. Therefore, higher numbers of hypotheses can be used in the averaging process at no additional coding cost. The proposed scheme has been implemented into the H.264/AVC reference software. Simulation results show bitrate reductions compared to H.264/AVC of 7.7% on average for the tested video sequences. It is shown that part of the performance gain is due to rounding effects in H.264/AVC sub-pixel interpolation which can be exploited in the averaging calculation of the proposed multihypothesis prediction. Experiments with an improved interpolation filter for both reference scheme and the proposed scheme still yield bitrate reductions of 4.7% on average.

Keywords: Multihypothesis prediction, Video Coding, H.264/AVC, Template matching, Sub-pixel interpolation

1. INTRODUCTION

Video coding exploits spatial and temporal correlation found in natural image sequences for bitrate reduction. Inter frame prediction plays an especially important role by using motion compensated regions from previously coded pictures for signal prediction. Assuming that the content between temporally adjacent frames remains similar for many typical video sequences, temporal prediction is the main contributor to the coding efficiency in many video coding standards such as MPEG-4[1] and H.264/AVC.[2] Typically, the encoder performs a motion estimation (ME) to determine translational displacements of regions and encodes the displacements as motion vectors into the bitstream (forward motion coding). Additional gains can be achieved by performing multihypothesis prediction,[3] using a linear combination of multiple temporal predictions to obtain the final prediction signal. H.264/AVC uses multihypothesis prediction in B slices, where up to two temporal predictions are linearly combined. However, the bitrate required for coding the additional motion vectors limits the number of hypotheses that can be used for efficient video coding.[4]

Recently, template matching (TM) based schemes have been successfully applied to inter prediction, using pixel-by-pixel prediction,[5] 8×8 block based prediction,[6] multihypothesis prediction[7] and multiple reference pictures.[8] These schemes allow for temporal prediction without the need to transmit actual motion vector data. Instead, motion vectors are derived using a template matching algorithm at the decoder side. While this increases the decoder complexity, significant bitrate reductions have been observed.

In this paper, we extend our previous work on decoder side motion vector derivation[8] (DMVD) in H.264/AVC to multihypothesis prediction.[3,4] Compared to the scheme by Suzuki et al.,[7] the individual prediction signals are taken from all reference pictures available to the encoder instead of only the first reference picture. In order to limit the decoder complexity, the decoder side motion estimation uses a restricted search range around the H.264/AVC motion vector predictor. We further derive the expectation of the rounding error in sub-pixel interpolation as used in H.264/AVC inter prediction. By adjusting the rounding offset in the linear combination calculation of the presented multihypothesis scheme, the rounding error inherent to H.264/AVC sub-pixel interpolation can be partially equalised, resulting in additional bitrate reduction.

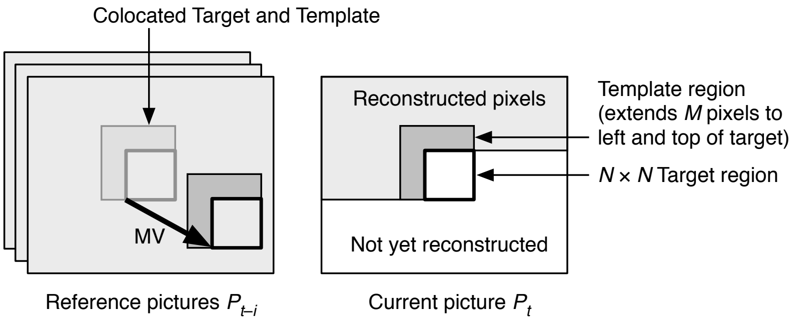
Colocated Target and Template
MV
Reference pictures P_{t-i}
Reconstructed pixels
Not yet reconstructed
Current picture P_t
Template region (extends M pixels to left and top of target)
$N \times N$ Target region

Figure 1. Template Matching.

2. BACKGROUND

2.1 Inter Prediction in H.264/AVC

In H.264/AVC,[2] the video signal is decomposed into a regular grid of macroblocks (MB) consisting of 16×16 luma samples and their associated chroma samples. For the case of P slices, each MB is coded using either intra or inter prediction. The available inter prediction modes allow for the subpartitioning of a macroblock where a single motion vector is coded for each (sub-)partition. Additionally, a reference picture list index is transmitted for each partition, specifying which reference picture to use for motion compensation. Sub-partitions inside a 8×8 partition share the same reference picture index.

The coding of motion vectors is performed predictively, where a motion vector predictor (MVP) is derived from motion vectors of (sub-)partitions adjacent to the current (sub-)partition. Then, only the difference between the MVP and the actual motion vector needs to be coded (motion vector difference, MVD). This allows a very efficient coding in areas with regular or identical motion.

For most prediction modes, the residual signal is obtained as the difference between the motion compensated prediction signal and the original signal. The residual is then quantised and coded into the bitstream. A special prediction mode is the Skip mode, where neither MVD nor residual signal are coded. Instead, the prediction is obtained solely by motion compensation using the MVP.

2.2 Template Matching

The proposed decoder side motion vector derivation scheme (DMVD) uses template matching (TM) at the encoder and decoder side in order to derive motion information. TM exploits correlation between the pixels from blocks adjacent to the prediction target block and those in already reconstructed reference pictures. The basic principle is shown in Figure 1: In order to derive motion information for a $N \times N$ target region in the current picture P_t, an inverse-L shaped template region is defined extending M pixels from the top and left of the target region. The template region only covers already reconstructed pixels of P_t. Then, the best displaced template region in the reference pictures P_{t-i} is determined by minimising the sum of absolute differences (SAD) between the samples of the current and reference template regions. The spatio-temporal displacement of the best-matching template is used as motion vector (MV) and reference picture index i for motion compensated prediction of the target.

3. DECODER SIDE MOTION VECTOR DERIVATION

3.1 Basic Scheme

The proposed scheme is based on our previous work on decoder side motion vector derivation (DMVD)[8] and will be briefly summarised here.

H.264/AVC employs motion compensated prediction for the 16×16, 16×8, 8×16 and 8×8 macroblock types. Typically, the encoder determines the motion vectors using a block matching algorithm and codes the displacement vector into the bitstream. A residual signal is then obtained by performing motion compensated

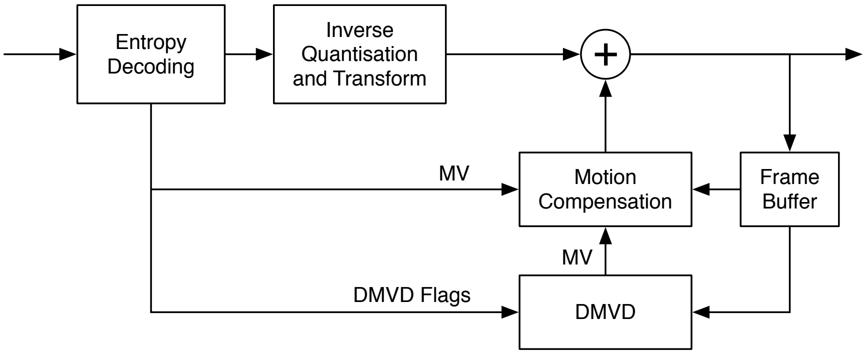

Figure 2. Simplified block diagram of decoder with DMVD.

prediction using the coded motion vector and calculating the difference to the original signal. The residual signal is then quantised and coded into the bitstream. We have extended the available P macroblock types to optionally use TM to derive motion information at the decoder side instead of explicitly coding the motion vectors in the bitstream. A block diagram of the modified decoder is depicted in Figure 2. As the MV derivation can be performed identically at the encoder and decoder side, the encoder only needs to signal that DMVD should be used to derive motion vectors and reference indices instead of explicitly coding the information into the bitstream.

For the 16×16, 16×8 and 8×16 types one additional flag per partition is present in the bitstream, signalling whether forward motion coding or DMVD is to be used for the macroblock. In the 8×8 type, DMVD is only allowed for sub-blocks coded in the 8×8 sub-type. A single flag per 8×8 sub-type is coded for DMVD. It is important to note that while DMVD allows the decoder to derive the motion vector autonomously, in some cases DMVD may not find a suitable vector. Therefore, the encoder performs a rate-distortion optimised decision between enabling DMVD for a specific (sub-)partition and explicitly signalling the motion vectors.

For each type the DMVD search is performed in all reference pictures available for the prediction of the current macroblock. Thus, the reference picture index can be derived in addition to the motion vectors, providing additional bitrate savings. Due to larger temporal distances, motion vector norms tend to increase when using higher reference picture indices. However, by centring the TM search window on the motion vector predictor (MVP) for the respective reference picture, the search range can be limited. Our experiments show that a search range of ±2 pixels horizontally and vertically is sufficient for providing most of the gains of DMVD.

In the described DMVD scheme, the decoding process for DMVD blocks is dependent on previously decoded samples. This includes samples in the current picture which may be part of the search template, as well as samples in decoded reference pictures for the DMVD search. In our previously published scheme,[8] the availability of syntax elements in the bitstream has been dependent on DMVD results for P-Skip macroblocks, therefore leading to a bitstream parsing problem in the presence of errors in decoded samples (e. g. lost slices). However, we have found that the coding efficiency contribution of DMVD in P-Skip MBs tends to be negligible for higher qualities and picture sizes. Therefore, we have disabled DMVD in P-Skip macroblocks for this paper, yielding fully parsable bitstreams even if slices are lost on the channel. In this case, DMVD will derive incorrect MVs at the decoder. While the effects of transmission errors are subject to further research, we assume decoding artefacts to be similar in nature to H.264/AVC without DMVD.

3.2 Multihypothesis Prediction

During the template matching stage the SAD for the template is calculated as cost for each tested motion vector. Instead of just identifying the motion vector with minimum cost, a set of L vectors with lowest cost is determined for each target among all available reference pictures.

The computational complexity of the matching itself is virtually identical to the single hypothesis case as all possible vectors have to be tested in any case. The hypotheses are not further jointly optimised, so only a single TM pass needs to be performed. However, some additional complexity and storage is required to maintain the set of motion vectors as ordered list sorted by matching cost during the DMVD process. Our proposed scheme therefore allows for a very efficient signalling of multihypothesis prediction, using only a single flag per (sub-)partition and requiring no additional bitrate for coding the individual motion vectors. This facilitates the use of more than two hypotheses (as in conventional B slices). The averaging is performed using

$$S(x,y) = \left\lfloor \frac{1}{L} \left(\left(\sum_{l=0}^{L-1} R(x+x_l, y+y_l, t_l) \right) + U \right) \right\rfloor , \tag{1}$$

where $S(\cdot)$ is the final prediction value, x_l and y_l are the vector components and t_l is the reference frame index belonging to the l-th derived motion vector. $R(\cdot)$ denotes the reference picture signals and U is a rounding offset. We have examined two rounding strategies: $U = \frac{L}{2}$ rounds half-way cases away from zero, $U = 0$ rounds all fractional values towards zero. The motivation for the selection of the rounding offset U is discussed in the following section.

3.2.1 Rounding Effects in Sub-pixel Interpolation

In H.264/AVC, motion compensation is performed with quarter-pixel accuracy for the luma component. Half-pixel positions are obtained by applying a 6-tap FIR filter with coefficients $H := [1, -5, 20, 20, -5, 1]$ in either horizontal or vertical direction. Afterwards, the samples are normalised through division by $\sum_i H_i = 32$. Diagonal half-pixel samples are obtained by successive horizontal and vertical filtering using H, with subsequent division by $\sum_i H_i \cdot \sum_i H_i = 1024$. Quarter-pixel samples are bilinearly interpolated from two neighbouring half-pixel positions using the 2-tap filter $Q = [1, 1]$ and division by 2. All division operations are performed in integer precision as $\hat{x} = \lfloor \frac{x+2^{n-1}}{2^n} \rfloor$ with $n = 5$ for the 6-tap filter applied once (horizontal or vertical half-pixel positions), $n = 10$ for the 6-tap filter applied twice (diagonal half-pixel positions) and $n = 2$ for the bilinear case (quarter-pixel positions). Therefore, half-way cases are always rounded to the next greater integer value.

The residual value of the division can be expressed as $r(x,n) = \hat{x} - \frac{x}{2^n}$. Disregarding clipping effects, the expectation of the rounding error then becomes $E\left(r(x,n)\right) = \sum_{r(x,n)} r(x,n) \cdot p\left(r(x,n)\right)$. If we further assume that all residual values of the division are equally distributed with $p\left(r(x,n)\right) = \frac{1}{2^n}$, we obtain $E\left(r(x,n)\right) = \frac{1}{2} p\left(r(x,n) = \frac{1}{2}\right) = \frac{1}{2^{n+1}}$.

Then, using Eq. (1) and the dependencies between sub-pixel values we can derive the expectation of the rounding error for individual sub-pixel positions $e\left(s_x, s_y\right)$ where s_x and s_y denote the horizontal and vertical sub-pixel positions. We obtain:

$$e\left(0,0\right) = 0 \tag{2}$$

$$e\left(0, \frac{1}{2}\right) = e\left(\frac{1}{2}, 0\right) = \frac{1}{64} \tag{3}$$

$$e\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2048} \tag{4}$$

$$e\left(0, \frac{1}{4}\right) = e\left(0, \frac{3}{4}\right) = e\left(\frac{1}{4}, 0\right) = e\left(\frac{3}{4}, 0\right) = \frac{33}{128} \tag{5}$$

$$e\left(\frac{1}{2}, \frac{1}{4}\right) = e\left(\frac{1}{2}, \frac{3}{4}\right) = e\left(\frac{1}{4}, \frac{1}{2}\right) = e\left(\frac{3}{4}, \frac{1}{2}\right) = \frac{1057}{4096} \tag{6}$$

$$e\left(\frac{1}{4}, \frac{1}{4}\right) = e\left(\frac{1}{4}, \frac{3}{4}\right) = e\left(\frac{3}{4}, \frac{1}{4}\right) = e\left(\frac{3}{4}, \frac{3}{4}\right) = \frac{17}{64} \tag{7}$$

I. e. the expectation of the rounding error can be estimated to be $\frac{1}{64}$ for horizontal and vertical half-pixel positions, $\frac{1}{2048}$ for diagonal half-pixel positions, and approximately $\frac{1}{4}$ for quarter-pixel positions. If we assume that all possible sub-pixel positions are chosen with equal probability for motion compensated prediction, the average expected rounding error is $\sum_{s_x,s_y} e\left(s_x, s_y\right)/16 = \frac{6467}{32768} \approx 0.197$.

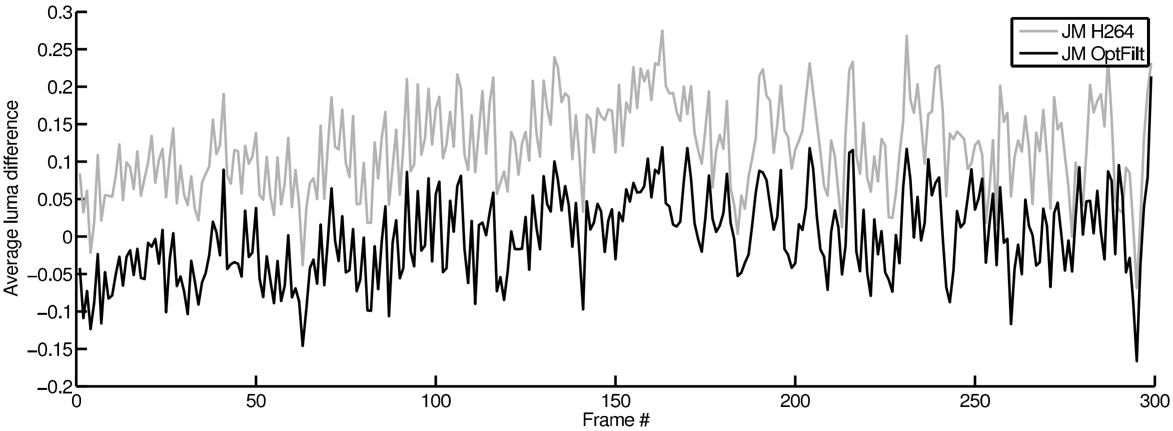

Figure 3. Average luma difference between prediction signal and decoded signal for unmodified JM (average: 0.12) and JM with optimised interpolation filter (average: −0.001).

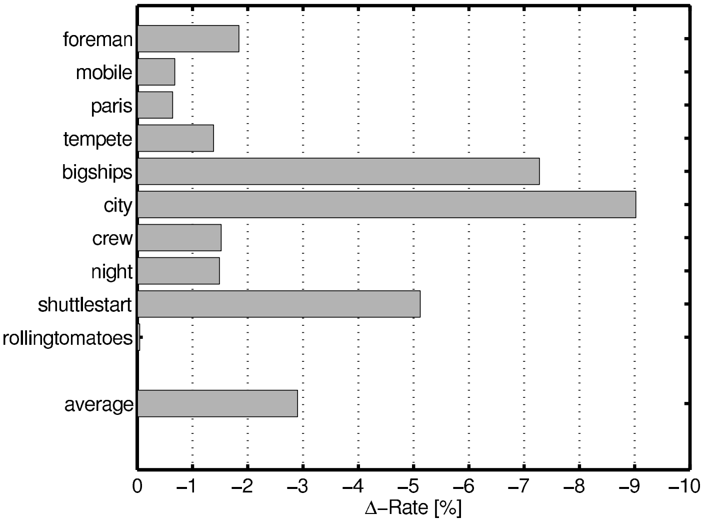

Figure 4. Average bitrate savings when using an optimised interpolation filter compared to the unmodified JM.

The expected rounding error can also be interpreted as a brightness increase of the prediction signal over the original image. Figure 3 shows the average luma difference between the prediction signal (without residual) and the actual decoded signal for individual frames of sequence *Mobile* at QP $= 0$. The experimentally found average luma difference 0.12 of the unmodified JM slightly deviates from the theoretically derived value of 0.197 but confirms our general assumption. This brightness increase needs to be compensated by corresponding coding of residual data or techniques such as weighted prediction, increasing the total required bitrate.

By using $U = 0$ in Eq. (1), the expectation of the rounding error due to multihypothesis averaging in DMVD is $\frac{1-L}{2L}$ when assuming statistically independent sub-pixel positions of DMVD MVs contributing to a single multihypothesis prediction. As this expected averaging error is less than zero for all $L > 1$, an appropriate selection of U can partially alleviate the rounding error introduced by sub-pixel interpolation. We have confirmed this assertion in our results in Section 4.

Table 1. Simulation settings according to VCEG common conditions for coding efficiency experiments.[10]

Sequences	Foreman, Mobile, Tempete (CIF, 30 Hz), Paris (CIF, 15 Hz) BigShips, ShuttleStart, City, Crew, Night (720p, 60 Hz, 150 frames) RollingTomatoes (1080p, 60 Hz, 60 frames)
Prediction structure	IPPP. . . , High Profile
Entropy coding	CABAC
Quantisation parameter	QPI: 22, 27, 32, 37; QPP=QPI+1
Reference pictures	4
Template size M	4
Target size N	16 (16×16 type) 8 (16×8 and 8×16 types) 4 (8×8 type)
Search range	±2 pixels

3.2.2 H.264/AVC with Improved Sub-pixel Interpolation

In order to assess the effects of rounding due to sub-pixel interpolation, we have implemented a modified interpolation filter into the JM reference software. While we have retained the half-pixel filter H and the bilinear interpolation for quarter-pixel positions, rounding is performed as the last step in the sub-pixel filtering stage. I. e. the expectation of the rounding error is almost zero for all sub-pixel positions. Therefore, the prediction signal has no increased brightness compared to the decoded signal, as can be seen from Figure 3 where the average luma difference is 0.001 for the improved interpolation filter ("OptFilt"). Figure 4 shows average bitrate savings[9] of JM with optimised filter compared to an unmodified JM. Using the filter with "late" rounding provides average bitrate savings of 2.9%.

4. SIMULATION RESULTS

We have implemented the proposed scheme into the H.264/AVC reference software (JM 13.2). The simulation conditions are according to the VCEG common conditions for coding efficiency[10] and are summarised in Table 1. Coding efficiency comparisons are relative to the unmodified JM software using the Bjøntegaard average delta bitrate (BD-Rate) assessment method.[9]

The results of the proposed multihypothesis prediction scheme using DMVD using different number of hypotheses for averaging and the original H.264/AVC interpolation filter are shown in Figure 5. It can be observed that the prediction quality improves with an increasing number of hypotheses, although on average saturation occurs above 4 hypotheses. At 8 hypotheses, the performance decreases again as the additional hypotheses deteriorate the prediction quality. The upper graph shows results with $U = 0$, where on average 7.7% bitrate is saved when using 4 hypotheses. Peak savings are observed for *City* with 17% bitrate reduction. The effect of the rounding offset U is visible when comparing with $U = \frac{L}{2}$ in the lower graph: Using 4 hypotheses now only reduces the bitrate by 4.6% on average with a peak reduction of 9.9% for *City*.

When using the improved interpolation filter (Figure 6) the impact of the rounding offset U is mostly eliminated, as the brightness of the prediction signal matches the original signal. For both, $U = 0$ and $U = \frac{L}{2}$ bitrate reductions of 4.7% are obtained when using 4 hypotheses. *ShuttleStart* still benefits from $U = 0$ with 6.3% ($U = 0$) compared to 3.4% ($U = \frac{L}{2}$) which can be attributed to the sequence characteristics: The background in *ShuttleStart* slowly gets darker due to the camera adjusting to the increasing brightness in the centre. This fade is better approximated using the downward rounding in multihypothesis prediction for $U = 0$.

5. CONCLUSION

This paper proposed a multihypothesis prediction scheme using decoder side motion vector derivation for inter frame video coding. A specific implementation with optimisations for H.264/AVC has been presented, yielding 7.7% bitrate reductions on average for a broad set of test sequences. It was further shown that part of the

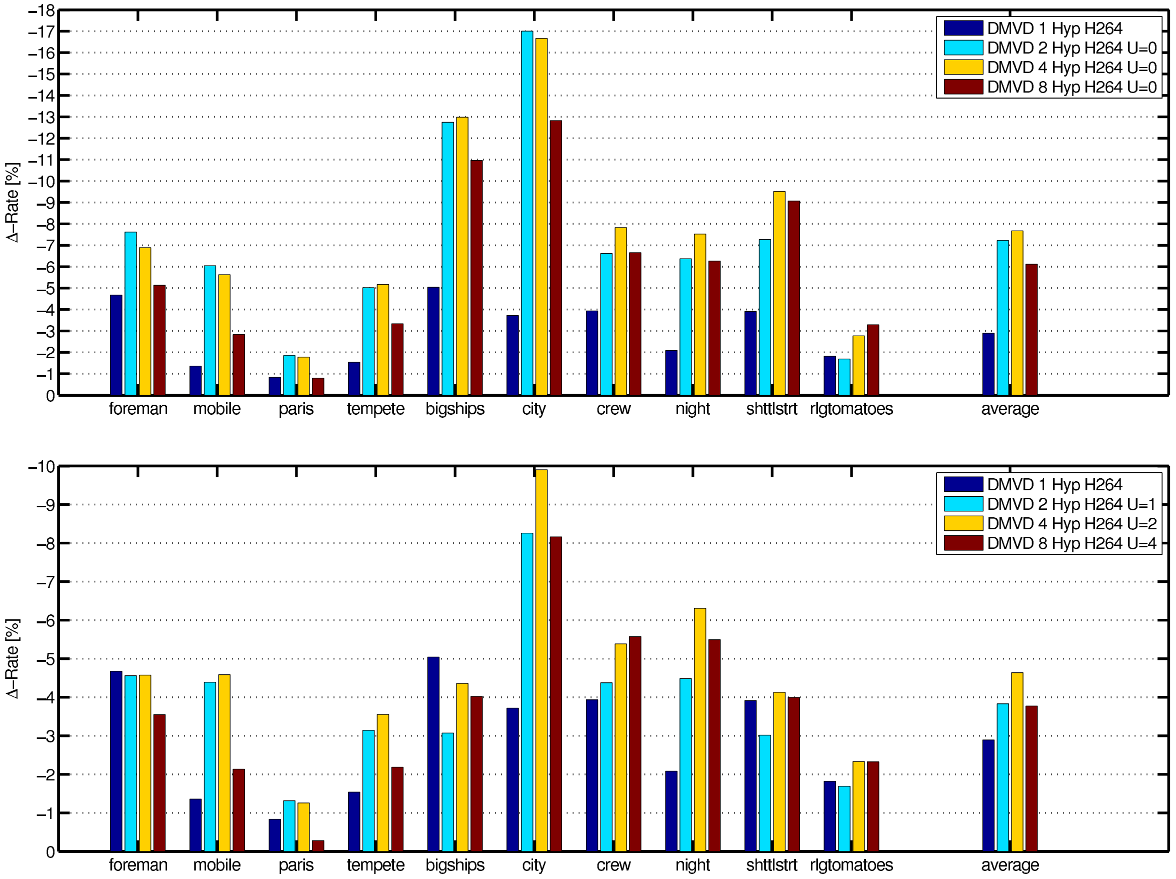

Figure 5. Bitrate savings of DMVD relative to the unmodified JM, both with original H.264/AVC interpolation filter, number of hypotheses L, rounding offset $U = 0$ (top) and $U = \frac{L}{2}$ (bottom).

performance gain over H.264/AVC can be attributed to rounding errors in sub-pixel interpolation which can be alleviated by the multihypothesis averaging calculation. With the JM and our propsed scheme both using an improved sub-pixel interpolation filter, average bitrate reductions of 4.7% are observed for the proposed motion vector derivation scheme.

REFERENCES

[1] ISO/IEC 14496-2, “Information technology – Generic coding of audio-visual objects: Visual,” (1998).

[2] *ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services* (Mar. 2005).

[3] Sullivan, G. J., “Multi-hypothesis motion compensation for low bit-rate video coding,” in [*Proc. IEEE Int. Conference on Acoustic Speech and Signal Processing ICASSP ’93*], 427–440 (Apr. 1993).

[4] Flierl, M., Wiegand, T., and Girod, B., “Rate-constrained multihypothesis prediction for motion-compensated video compression,” *IEEE Transactions on Circuits and Systems for Video Technology* **12**, 957–969 (Nov. 2002).

[5] Sugimoto, K., Kobayashi, M., Suzuki, Y., Kato, S., and Boon, C. S., “Inter frame coding with template matching spatio-temporal prediction,” in [*Proc. IEEE Int. Conference on Image Processing ICIP ’04*], 465–468 (Oct. 2004).

[6] Suzuki, Y., Boon, C. S., and Kato, S., “Block-based reduced resolution inter frame coding with template matching prediction,” in [*Proc. IEEE Int. Conference on Image Processing ICIP ’06*], 1701–1704 (Oct. 2006).

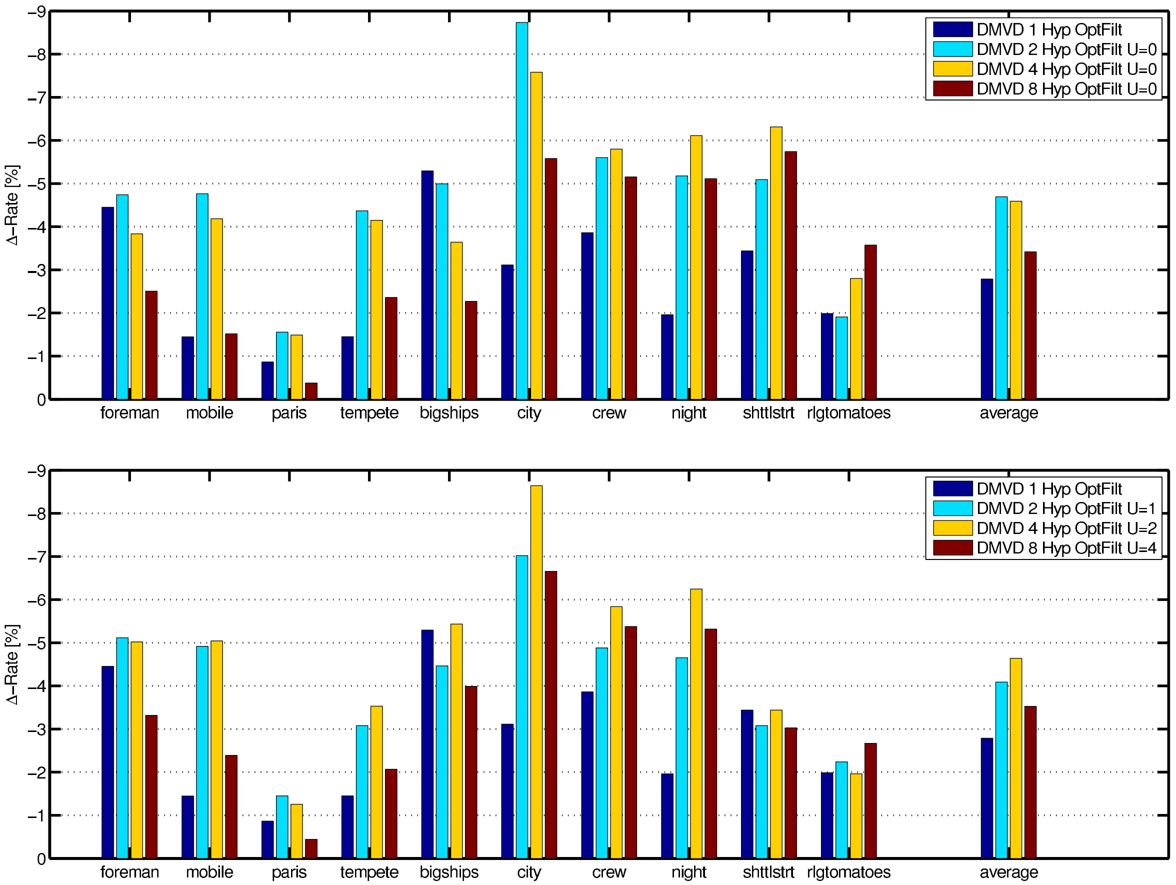

Figure 6. Bitrate savings of DMVD relative to JM without DMVD, both with improved interpolation filter, number of hypotheses L, rounding offset $U = 0$ (top) and $U = \frac{L}{2}$ (bottom).

[7] Suzuki, Y., Boon, C. S., and Tan, T. K., "Inter frame coding with template matching averaging," in [*Proc. IEEE Int. Conference on Image Processing ICIP '07*], III–409–III–412 (Sept. 2007).

[8] Kamp, S., Evertz, M., and Wien, M., "Decoder side motion vector derivation for inter frame video coding," in [*Proc. IEEE Int. Conference on Image Processing ICIP '08*], 1120–1123 (Oct. 2008).

[9] Bjøntegaard, G., "Calculation of average PSNR differences between RD curves," Doc. VCEG-M33, ITU-T SG16/Q6 VCEG, 13th Meeting, Austin, TX, USA (Apr. 2001).

[10] Tan, T. K., Sullivan, G. J., and Wedi, T., "Recommended simulation common conditions for coding efficiency experiments, revision 2," Doc. VCEG-AH010r2, ITU-T SG16/Q6 VCEG, 34th Meeting, Antalya, Turkey (Jan. 2008).