AN ADAPTIVE LINEAR ESTIMATOR BASED APPROACH TO BI-DIRECTIONAL MOTION COMPENSATED PREDICTION

Bohan Li[*], *Jingning Han*[†], *Kenneth Rose*[*]

Signal Compression Lab, UCSB[*] and Google LLC[†]

ABSTRACT

Bi-directional motion compensated prediction is widely utilized in video coding. Conventionally, the encoder searches for two motion vectors pointing to reference frames in both directions, and transmits these motion vectors to the decoder. Recognizing that the two reference frames are already available to the decoder, prior work proposed decoder-side motion estimation to extract motion information or optical flow, at the cost of dramatic increase in decoder complexity. This paper proposes a novel bi-directional motion compensation mode that efficiently utilizes the motion information that is already available to the decoder, without recourse to extensive search. An estimation theory based approach is proposed and utilized to provide a high quality prediction, which adaptively combines contributions from multiple motion-compensated references. Experimental results show that the proposed method, while yielding a greatly reduced decoder side complexity, introduces a significant coding gain for a diverse set of video sequences.

Index Terms— Video coding, motion compensation, hierarchical structure, linear estimation

1. INTRODUCTION

Motion compensated prediction is a critical component of modern video codecs [1, 2]. Conventionally, block-based search is performed in a reference frame to determine a match for the current block, and the corresponding motion vector (MV) is coded and transmitted to the decoder. The introduction of hierarchically structured prediction in video codecs added modes with multiple motion vectors, pointing to different reference frames (possibly in different directions) to be coded and transmitted for the current block.

However, conventional motion compensation scheme largely overlooks the motion information that is already available to the decoder. For example, when referring to two reference blocks from reference frames in both the forward and backward directions, the reference frames that have already been decoded and reconstructed by the decoder, implicitly contain relevant motion information that can and should be exploited for encoding the current frame. Thus, the two transmitted motion vectors may contain significant redundancy.

The above realization led researchers to propose decoder-side motion estimation to extract "free" motion information from the reference frames, in the form of either block-based motion [3] or dense motion fields [4, 5]. In another approach [6], optical flow estimation techniques are utilized to enhance the transmitted motion vectors rather than replace them, and in this way exploit the available motion information.

Yet another approach was proposed in [7], where optical flow estimation techniques are used to estimate the dense motion field between the reference frames, which is not used directly to predict the current block, but to interpolate a co-located reference frame. An extra motion estimation step is then performed by the encoder, and the resulting motion vector is transmitted to the decoder. It eliminates possible offsets caused by the implicit assumption of linear motion trajectory and significantly enhances the performance.

Note that beside the benefits of decoder-side motion estimation, in terms of redundancy removal, there are also disadvantages, namely, the impact of quantization error in the reference frames, and considerable increase in decoder complexity. This paper builds on the realization that the motion information available at the decoder not only lies in the reconstructed reference frames themselves, but also in the previously transmitted explicit motion information obtained from the encoder. Such motion information may be of higher quality since the encoder has direct access to the source, uncorrupted by quantization errors. Moreover, since this information has already been decoded, there is no need for extensive motion estimation at the decoder, with the important benefit of maintaining low decoder complexity.

Therefore, we propose the following method with three main steps. First, we utilize the motion vectors that were already transmitted to the decoder between the reference frames and form a set of motion vector candidates for every pixel location in the current frame. Then, from an estimation theory perspective, we treat their corresponding motion compensated references as observations that are correlated with the current pixel. An optimal linear estimator is adaptively determined using local statistics, and the prediction of the current pixel is calculated accordingly. Finally, as proposed in [7], the predictions form a co-located reference frame and a motion vector

This work was supported by Google LLC.

is transmitted in order to eliminate possible offsets from the assumed linear motion.

Experimental results demonstrate that without recourse to extensive motion search (i.e., at minimal complexity increase), the proposed scheme provides high quality prediction, and yields significant coding gains.

2. THE PROPOSED INTERPOLATED MOTION COMPENSATION SCHEME

2.1. Candidate motion vector generation

The first key step of the proposed scheme is to generate candidate motion vectors. Rather than perform decoder-side motion estimation and incur quantization errors and high complexity, the approach reutilizes the motion vectors previously transmitted to derive the MV candidates.

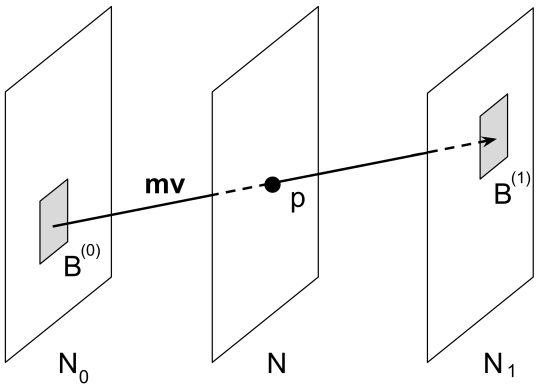

Fig. 1. Illustration of linear motion vector projection. Motion vector $\mathbf{mv}$ between frame N_0 and N_1 intersects frame N at pixel location p.

Consider the set up where the current frame at time N is being processed, and there exist two bi-directional reference frames at time N_0 and N_1. Note that, frame N_0 and N_1 are already decoded and reconstructed, and certain motion information between them is already extracted in the decoding process. For example, as shown in Figure 1, block $B^{(1)}$ in frame N_1 is an inter-predicted block referring to block $B^{(0)}$ in frame N_0, as specified by $\mathbf{mv}$. Note that $\mathbf{mv}$ intersects frame N at the location of pixel p. Assuming that $\mathbf{mv}$ represents the true motion of the block, and that it follows a linear trajectory, then $\mathbf{mv}$ is expected to also represent the motion of the pixels around p in frame N.

Consider next all the available motion vectors that point between frame N_0 and N_1. Let each such motion vector, $\mathbf{mv}_i$, intersect frame N at a respective pixel p_i. For any pixel p in the current frame N, we propose to determine a set of M candidate motion vectors, defined as the M motion vectors whose intersection points p_i are nearest to p.

In order to increase the density of available motion vector in the current frame, we further derive effective motion vectors from available motion vectors that do not directly connect N_0 and N_1. For example, if $\mathbf{mv}_{12}$ points from frame N_1 to N_2 and $\mathbf{mv}_{20}$ points from frame N_2 to N_0, then $\mathbf{mv}_{12} + \mathbf{mv}_{20}$ can also serve as an additional candidate motion vector.

2.2. Linear Estimator Based Interpolation Method

Consider a pixel in the current frame. As discussed in section 2.1, M candidate motion vectors are generated for this pixel, denoted by $\mathbf{mv}_i$ where $i = 1, 2, ..., M$. Let us denote the value of the pixel as y. Given the candidate motion vectors, M references can be generated by linearly combining each reference pair of pixels from the respective frames N_0 and N_1 (denoted as $y_i^{(0)}$ and $y_i^{(1)}$). The combined references are denoted $\mathbf{x} = (x_1, x_2, ..., x_M)^T$. For now let us assume the simple case where $N - N_0 = N_1 - N$ to illustrate the basic idea (then $x_i = 0.5y_i^{(0)} + 0.5y_i^{(1)}$), noting that derivation for a general setup is straightforward.

From an estimation theory perspective, we regard the references x_i as M observations correlated with y. A linear estimator with weight vector $\mathbf{w} = (w_1, w_2, ..., w_M)^T$ is used to generate an estimate of y:

$$\tilde{y} = \mathbf{w}^T \mathbf{x}. \tag{1}$$

The weights $\mathbf{w}$ are determined such that they minimize the mean squared prediction error:

$$\mathbf{w} = argmin_{\mathbf{w}}\{E\{(y - \tilde{y})^2\}\}. \tag{2}$$

A standard result in linear estimation theory, obtained by simple calculus, converts the problem into a set of linear equations:

$$E\{\mathbf{x}\mathbf{x}^T\}\mathbf{w} - E\{\mathbf{x}y\} = 0, \tag{3}$$

where $E\{\mathbf{x}\mathbf{x}^T\}$ is the $M \times M$ correlation matrix of $\mathbf{x}$, and $E\{\mathbf{x}y\}$ is the $M \times 1$ vector containing the cross correlations of x_i and y.

However, how to obtain the proper correlation matrix and the cross correlation vector still remains a challenging problem. As presented in [8] and [9], offline training of the weights can be performed by collecting relevant data from the video codecs to provide estimates of the relevant correlations. In this paper, we propose a different online adaptive method that utilizes the bi-directional prediction scheme to estimate such correlations on-the-fly to better adapt to local statistics.

First, let us consider the actual motion trajectory of the pixel of interest. As shown in Figure 2, the motion trajectory is represented by the dashed line, and the pixel values of the pixels at frame N_0 and N_1 are denoted by $y^{(0)}$ and $y^{(1)}$. Let us further assume an auto-regressive (AR) model along the motion trajectory. Since $N - N_0 = N_1 - N$, the correlations

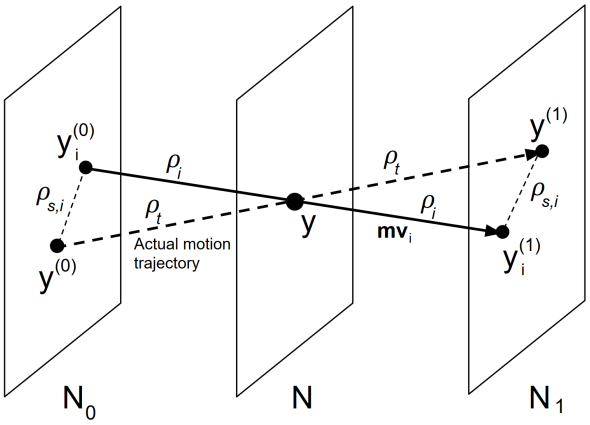

Fig. 2. Illustration of cross correlation calculation. Along the real motion trajectory (the dashed line), ρ_t denotes the temporal correlation coefficient. For each of the MV candidates $\mathbf{mv}_i$, the cross correlation coefficient ρ_i is given by the separable model: $\rho_i = \rho_t \rho_{s,i}$, where $\rho_{s,i}$ is the spatial correlation coefficient.

are symmetric, and we have:

$$\begin{aligned} y &= \rho_t y^{(0)} + n, \\ y^{(1)} &= \rho_t y + n^{(1)}, \end{aligned} \tag{4}$$

where ρ_t is the temporal correlation coefficient. Assuming constant variance σ_y^2, we have: $\rho_t = \frac{E(y^{(0)}y)}{\sigma_y^2} = \frac{E(yy^{(1)})}{\sigma_y^2}$. n and $n^{(1)}$ are white noise random variables that are uncorrelated with y, $y^{(0)}$ and $y^{(1)}$.

Given the AR model of (4), the correlation coefficient between $y^{(0)}$ and $y^{(1)}$, denoted ρ_{12}, is:

$$\rho_{12} = \frac{E(y^{(0)}y^{(1)})}{\sigma_y^2} = \rho_t^2. \tag{5}$$

Now consider a motion vector candidate $\mathbf{mv}_i$, which in general may not be the same as the true motion trajectory as illustrated in figure 2. Here, the separable spatio-tempopral correlation model is further assumed, such that the correlation coefficient, $\rho_i = \frac{E\{y_i^{(0)}y\}}{\sigma_y^2} = \frac{E\{yy_i^{(1)}\}}{\sigma_y^2}$ is given by:

$$\rho_i = \rho_t \rho_{s,i}, \tag{6}$$

where $\rho_{s,i}$ is the spatial correlation coeffcient relevant to the distance between $y^{(0)}$ and $y_i^{(0)}$ (which equals the distance between $y^{(1)}$ and $y_i^{(1)}$ in this setting due to symmetry).

Analogous to (5), the correlation coefficient between $y_i^{(0)}$ and $y_i^{(1)}$, denoted $\rho_{12,i}$ can be related to ρ_i:

$$\rho_{12,i} = \rho_i^2. \tag{7}$$

Now, for a given $\mathbf{mv}_i$, the reference pixel value is $x_i = 0.5y_i^{(0)} + 0.5y_i^{(1)}$. With the same pixel value variance $\sigma_x = \sigma_y = \sigma$, we obtain:

$$E\{x_i y\} = \rho_i \sigma^2 = \sqrt{\rho_{12,i}}\sigma^2. \tag{8}$$

Since the two reference frames are already reconstructed, we propose to estimate $\rho_{12,i}$ on the fly by collecting neighboring data (a 5×5 patch) around pixels $y_i^{(0)}$ and $y_i^{(1)}$. Once $\rho_{12,i}$ is calculated for each $i = 1, 2, ..., M$, the cross correlation vector is calculated according to (8).

Next, the remaining ingredient needed to determine the linear estimator by solving (3), is the correlation matrix, i.e., we need the correlations between candidates $E\{x_{i_1}x_{i_2}\}$ for any i_1 and i_2. From (8), for a given i, we can write x_i as:

$$x_i = \rho_i y + z_i, \tag{9}$$

where z_i is the "innovation" in x_i, that is, what is uncorrelated with y. Therefore,

$$E\{x_{i_1}x_{i_2}\} = \rho_{i_1}\rho_{i_2}\sigma^2 + E\{z_{i_1}z_{i_2}\}, \tag{10}$$

where $E\{z_{i_1}z_{i_2}\}$ is the correlation of the innovations relative to y. We propose to model this correlation with an exponential decay model, i.e., the correlation drops exponentially with the distance from the referred pixels. Note that this distance equals the difference between the two candidate motion vectors, therefore we have:

$$E\{z_{i_1}z_{i_2}\} = exp(-\alpha||\Delta\mathbf{mv}_{i_1,i_2}||)\sigma_{z_1}\sigma_{z_2}, \tag{11}$$

where $\Delta\mathbf{mv}_{i_1,i_2} = \mathbf{mv}_{i_1} - \mathbf{mv}_{i_2}$, and the variance of z_i is given by $\sigma_{z_i}^2 = (1-\rho_i^2)\sigma^2$.

Substituting (11) into (10), we obtain an estimate for the correlation matrix, and (8) provides the cross correlation vector. Therefore the corresponding linear estimator weights can now be obtained by (3) (note that the variance σ^2 cancels out and thus is not needed). The resulting linear estimator is used to generate the interpolated motion compensated prediction for this pixel location.

We re-emphasize that we assumed $N - N_0 = N_1 - N$ in the above derivations for simplicity of presentation. It is straightforward to derive the results for other settings using the same logic, and the details are omitted here for conciseness. Also, the derivation assumed zero-mean random variables, which can be approximated by subtracting a local mean or a constant bias.

2.3. Overall scheme with the co-located reference frame

As discussed in subsection 2.2, for every pixel location in the current frame, an interpolated motion compensated prediction is generated. These per-pixel predictions together form a prediction for the entire frame.

However, note that in subsections 2.1 and 2.2, results are obtained under the assumption of linear trajectory between the two reference frames, that is, the objects are are assumed

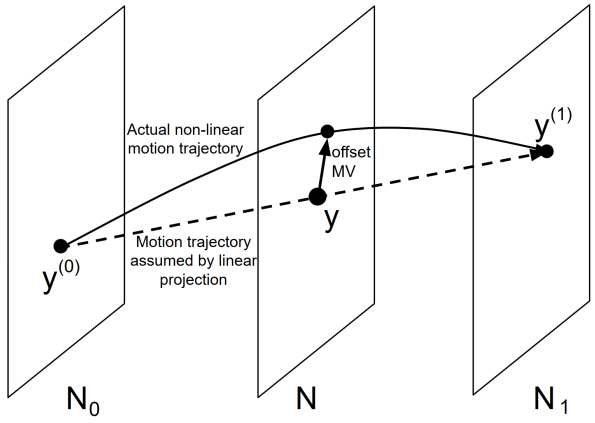

Fig. 3. Illustration of motion offset and the offset motion vector. As the motion trajectory may not be linear, we send an offset MV to the decoder to correct the resulting spatial shift.

to be moving with constant velocity in a constant direction. This in not necessarily true in general, and may result in possible offset or shift between the interpolated prediction and the source, as shown in Figure 3.

To mitigate this problem, we use the concept of co-located reference frame that was proposed in [7]. Instead of using the interpolated prediction directly, a reference frame is first constructed using the interpolated pixels, and then another regular block-based *offset motion vector* is transmitted to correct such possible offsets, as also illustrated in Figure 3.

3. EXPERIMENTAL RESULTS

The proposed method presented in section 2 was implemented in the AV1 video coding software [2]. Two sets of experiments were conducted, where the *baseline* set utilizes the regular compound mode in the AV1 codec, while the *proposed* set adds one extra prediction mode that predicts from the co-located reference frame interpolated by our proposed method. The performance was evaluated over a diverse set of sequences that vary in resolution and motion characteristics, and results are provided for a wide range of bit rates. For each sequence, 100 frames are encoded. Also, in our experiments, we set the parameters $M = 9$ and $\alpha = 0.1$.

The BD-rate reduction [10] of the proposed method compared to the baseline is shown in Table 1. It can be clearly seen that significant improvement (**3.27%** BD-rate reduction on average) is obtained with the proposed method. As an example, the rate-distortion (R-D) curve of bus_cif.yuv (which contains dramatic motions across the sequence) is shown in Figure 4. As can be seen, consistent gain is achieved for a wide range of bit-rate, which further proves the effectiveness of the method.

It should be emphasized that our method not only brings significant bit-rate reduction, but does so at greatly reduced complexity cost to the decoder compared to our prior work in [7], since we do not require decoder side motion search but instead utilize the already available motion vectors.

Table 1. BD-rate reduction using the proposed method.

Sequence	BD-rate change (%)
bus (CIF)	-3.11
mobile (CIF)	-3.64
tempete (CIF)	-0.81
flower (CIF)	-3.54
BlowingBubbles (416x240)	-2.80
BasketballPass (416x240)	-3.97
PartyScene (832x480)	-0.90
FourPeople (1280x720)	-4.97
KristenAndSara (1280x720)	-5.40
Johnny (1280x720)	-3.51
Average	**-3.27**

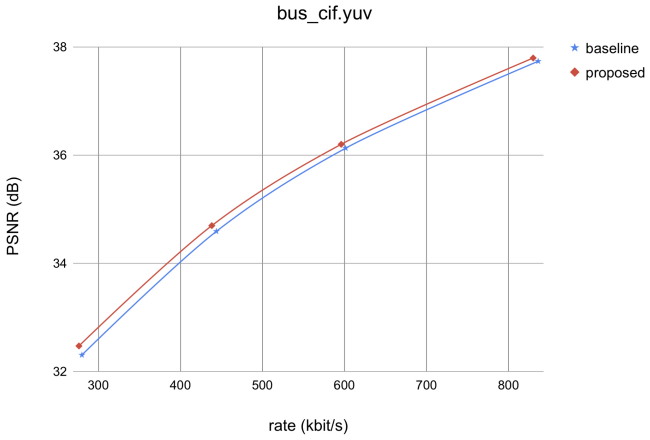

Fig. 4. The R-D curve for bus_cif.yuv.

Moreover, while the effectiveness of the proposed method was evidenced by the above AV1 results, it must be stressed that the design principles are generally applicable to other video codecs that employ motion-compensated prediction with hierarchical structures, such as H.264, HEVC, etc.

4. CONCLUSION

In this paper, a novel estimation-theory-based method is proposed, which effectively utilizes the available motion information at the decoder. The motion compensated prediction is adaptively interpolated by considering various references using an optimal linear estimator determined by local statistics. The proposed method is able to generate high-quality prediction and therefore brings significant coding performance improvement. Moreover, only negligible complexity is needed due to the effective use of existing information.

5. REFERENCES

[1] G. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] Y. Chen et al., "An overview of core coding tools in the av1 video codec," in *Picture Coding Symposium (PCS)*, 2018, pp. 24–27.

[3] S. Klomp, M. Munderloh, Y. Vatis, and J. Ostermann, "Decoder-side block motion estimation for h. 264/mpeg-4 avc based video coding," in *IEEE International Symposium on Circuits and Systems*, 2009, pp. 1641–1644.

[4] Y. Chin and C. Tsai, "Dense true motion field compensation for video coding," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 1958–1961.

[5] S. Klomp, M. Munderloh, and J. Ostermann, "Decoder-side hierarchical motion estimation for dense vector fields," in *Picture Coding Symposium (PCS)*, 2010, pp. 362–365.

[6] A. Alshin and E. Alshina, "Bi-directional optical flow for future video codec," in *2016 Data Compression Conference (DCC)*, March 2016, pp. 83–90.

[7] B. Li, J. Han, and Y. Xu, "Co-located reference frame interpolation using optical flow estimation for video compression," in *Data Compression Conference*, March 2018, pp. 13–22.

[8] W. Lin, T. Nanjundaswamy, and K. Rose, "Adaptive interpolated motion compensated prediction," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 943–947.

[9] W. Lin, T. Nanjundaswamy, and K. Rose, "Adaptive interpolated motion-compensated prediction with variable block partitioning," in *Data Compression Conference*, 2018, pp. 23–31.

[10] G. Bjontegaard, "Calcuation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4*, Apr 2001.