JOINT INTER-INTRA PREDICTION BASED ON MODE-VARIANT AND EDGE-DIRECTED WEIGHTING APPROACHES IN VIDEO CODING

Yue Chen[⋆]*, Debargha Mukherjee*[†]*, Jingning Han*[†]*, Kenneth Rose*[⋆]

[⋆]Department of Electrical and Computer Engineering, University of California Santa Barbara, CA 93106
[†]Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043
E-mail: [⋆]{yuechen,rose}@ece.ucsb.edu, [†]{debargha,jingning}@google.com

ABSTRACT

Most modern video compression codecs, like VP9, HEVC and H.264, encode square or rectangular blocks either by inter prediction or intra prediction. A joint inter-intra predictor that combines motion compensation and intra extrapolation by two novel weighting schemes is proposed to improve compression quality. Prior work on joint prediction employs inter-intra weights that only rely on the pixel locations. As an enhancement, we design a weighting approach by also considering the angle of intra prediction, which is the actual direction that the intra prediction errors evolve. Moreover, our second approach, inspired by prior work on geometric-partition-based motion compensation, breaks the limitation of traditional quad-tree partition by jointly using different predictors that implies soft step weighting functions for new and existing objects co-occurring around irregular motion edges. The proposed joint prediction approaches deliver consistent coding gains, as shown by extensive experiments on the experimental branch of VP9, Google's open source video compression tool.

Index Terms— Joint prediction, geometry partitioning, VP9, video coding

1. INTRODUCTION

Inter prediction, namely, motion compensation, copies the best matching block (or the linear combination thereof) with the lowest prediction error in reconstructed previous frames. On the other hand, intra prediction employs reconstructed neighboring pixels in the current frame to generate a prediction block by extending these pixels in particular patterns. Blocks in P frames have access to both decoded previous frames and decoded boundary pixels. However, when encoding blocks in P frames, current mainstream video codecs[1, 2] switch between motion compensation and intra prediction such that they always neglect one part of the available references. The under-utilization of references renders the predictor sub-optimal due to the fact that the optimal prediction is not always generated from a single source of reference. Especially in block-based prediction, motion compensation picks the best "matching" block in the sense of minimal block-wise prediction error regardless of the fact that pixels close to the boundary can be usually well estimated from the neighboring reconstructions. This motivates our joint inter-intra prediction approach that efficiently combine both predictions depending on the directionality of intra predictions as well as the location of motion edges separating areas within the same block but preferring different types of predictions.

Recent approaches[3, 4] on jointly exploiting inter and intra predictions combine them with a combination scheme defined by:

$$\tilde{x}_{com} = w_{inter}(i,j)\tilde{x}_{inter} + w_{intra}(i,j)\tilde{x}_{intra}, \quad (1)$$

where $w_{inter}(i,j) + w_{intra}(i,j) = 1$. Weighting coefficients $w_*(i,j)$ could either be constant across the block[3] or only depend on pixel locations (i,j) in the block[4] reflecting an assumption that the reliability of intra predictions is only relevant to pixel locations regardless of how the prediction is generated. This assumption is however over-simplified due to the fact that the direction to generate intra predicted pixels also have impact on the reliability distribution. While other more sophisticate adaptive algorithms[5, 6, 7] that exploit neighoring decoded blocks to design the optimal weights dramatically increase the codec complexity. Hence we first propose to apply *mode-variant* and position-variant weighting coefficients. Compared to non-adaptive prior work[3, 4], this approach could improve the prediction without adding complexity or even side information.

Note that all the prior work is implemented in a traditional partitioning scheme that only allows square or rectangular (with w/h ratio = 1:2 or 2:1) prediction units. This scheme has been shown R-D suboptimal for video compression[8]. Indeed, prior work on geometric partition based motion compensation[9, 10] implies that the coding performance can be raised by slicing inter blocks into more shapes, e.g., wedges, for which more accurate motions are applied to better fit contours of moving objects. Alternatively, other than the case that slices have distinct motion vectors, it is very likely to have different types of content on either side of the edge. One part could unveil new content that usually prefer

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

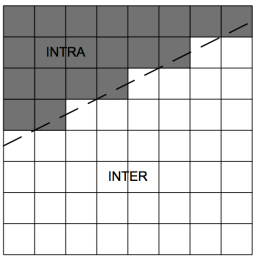

Fig. 1. Joint predictor based on motion-edge-directed weights.

intra prediction while the other part contains old content that can be well motion compensated. Hence to create a richer joint predictor, we also propose to generate combined prediction by applying different predictors on two slices in a block, as illustrated by Fig.1. This predictor, also a special case of (1) with binary weights, will thus allow content around motion edges, to be predicted by the most efficient methods. Moreover, to avoid high computation cost in the decoder due to motion edge estimation, we set up a codebook containing representative geometric partitions approximating real edges.

Both the proposed joint predictors focus on improving block predictions not well handled by separate inter or intra prediction, and are added as new coding options for blocks in P frames. Experiments validate that these methods reduce the bitrate consistently on video test sets with various resolutions.

2. JOINT PREDICTION USING INTRA MODE DIRECTED WEIGHTS

Consider an intra predicted block generated by copying pixels in a certain direction along which predictions can be written as:

$$\tilde{x}_k = \hat{x}_0, \tag{2}$$

where x_0 is the boundary pixel to be copied, and k denotes the distance from target pixel to x_0 (see Fig.2, an example of 117° intra mode). The prediction errors $|x_k - x_0|$ often increase by k due to the decreasing correlation with x_0, R_{x_0,x_k} along the angled line. While the optimal weights to combine motion compensation and one intra prediction are given by

$$\begin{aligned} w_{intra}(k) &= \frac{\sigma_{inter}(k)^2}{\sigma_{intra}(k)^2 + \sigma_{inter}(k)^2}, \\ w_{inter}(k) &= \frac{\sigma_{intra}(k)^2}{\sigma_{intra}(k)^2 + \sigma_{inter}(k)^2}, \end{aligned} \tag{3}$$

where $\sigma_{intra}(k)^2$ is the squared intra prediction error[4]. The inter prediction error $\sigma_{inter}(k)^2$, commonly considered as independent of the position, can be approximated as a constant inside a block given the stationary nature of images. Therefore, optimal w_{intra} is clearly a decreasing function along the prediction angle rather than only relevant to the 2-D location (i, j).

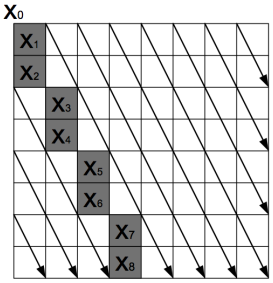

Fig. 2. Intra prediction following one direction.

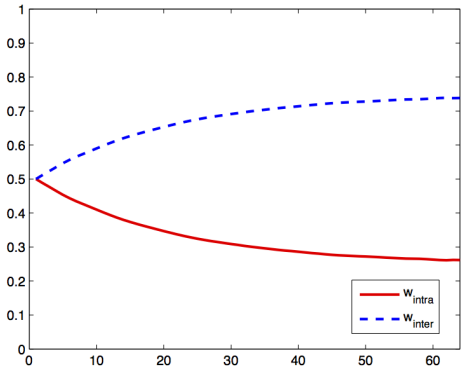

Fig. 3. The weighting function along the extrapolation direction.

First we design a 1-D decreasing weighting function shown in Fig.3

$$w_{intra}(k) = e^{-ak} + b \tag{4}$$

to capture the decaying inter-pixel correlation along the prediction angle. To generate 2-D joint inter-intra prediction, this 1-D weighting is applied along every angled line of predicted pixels. In VP9 we have the flexibility to combine motion compensation with 10 intra predictions that stand for different angles, and hence 10 sets of 2-D weighting coefficients $w(i, j|mode)$, which translate into 10 joint inter-intra prediction modes, are defined for them. Note that except the intra mode index, there is no additional side information to transmit because weights are preset for each mode.

3. JOINT PREDICTION USING MOTION EDGE DIRECTED WEIGHTS

In this section, based on prior discussions in Sec.1, a framework, that segments a traditional prediction block into irregular slices and applies different predictions on either side, is proposed to improve state of the art compression tools by better adapting to motion edges. While either estimating the motion edge exactly or directly representing it as a 2-D array, obviously will increase codec complexity or bit-rates heavily to code the side information. Hence we circumvent these

difficulties by using an edge codebook. Representative candidate segmentations are defined as entries in this codebook for efficient edge searching, whose result is transmitted as the index of the entry in the codebook.

3.1. Defining Geometric Block Partitions

To simplify the presentation and implementation, we restrict the candidate edges to oblique lines. We model these edges by the function of a line

$$f(x,y) = a_1(x - a_2\frac{w}{4}) + a_3(y - a_4\frac{h}{4}), \tag{5}$$

within a $w \times h$ block, where $(a_2\frac{w}{4}, a_4\frac{h}{4})$ denotes the coordinates of a pixel on the edge and the slope is defined by a_1/a_3. To combine intra prediction and motion compensation, the weighting coefficients in this framework are preliminarily given as a binary weighting scheme, namely a *hard mask*

$$w_{inter}(x,y) = \begin{cases} 1, & \text{if } f(x,y) \geq 0 \\ 0, & \text{if } f(x,y) < 0, \end{cases}$$
$$w_{intra}(x,y) = \begin{cases} 0, & \text{if } f(x,y) \geq 0 \\ 1, & \text{if } f(x,y) < 0. \end{cases} \tag{6}$$

The codebook records the integer parameters (a_1, a_2, a_3, a_4) defining each partition. In our experiments, we limited the maximum overhead size to 5 bits for 32 possible partitions, thus the slopes $\frac{a_1}{a_3}$ are chosen from a small set $\{0, \pm 1, \pm 0.5, \pm 2\}$ where the '$\pm$' determines two distinct weightings, that are complementary to each other, under the same partition.

Consider images partitioned by quad-tree method which has been shown R-D suboptimal for compression, larger blocks usually require more precise geometric segmentation to fit object contours while also being able to afford more overheads. Therefore in our codebook, the number of candidate edges varies with block sizes as shown in Table1. For example, the codebooks for 8×8 and 16×16 blocks are illustrated in Fig.4.

Table 1. Number of masks

Block size	Number of masks	Overhead/bits
$4 \times 4 - 8 \times 8$	8	3
$8 \times 16 - 32 \times 32$	16	4
$32 \times 64 - 64 \times 64$	32	5

3.2. Soft Masks

As we observed in our experiments, predicting the slices from different references will often create spurious high frequency components around the segmenting line. Indeed, the introduction of unwanted high frequency components make the transform coefficients less compact thereby limiting the compression efficiency. Note that this is also one reason why transforms larger than prediction units are rarely used in video codecs. Instead of directly applying binary weights in (6), soft masks are employed to avoid such loss in coding efficiency in joint predictions. Similar to the approach in Sec.2, a 1-D soft step weighting function $g(t)$ (see Fig.5) is designed to combine inter and intra predictions smoothly around the partitioning line while still retaining the binary weighting scheme for pixels far away. Specifically, to generate 2-D prediction, the inter and intra predictors for each pixel (x, y) within a block are combined with weights w_{intra} and $w_{inter} = 1 - w_{intra}$, where w_{intra} is obtained as:

$$w_{intra} = g\left(\frac{f(x,y)}{\sqrt{a_1^2 + a_3^2}}\right) \tag{7}$$

with the argument representing the distance to the edge along the direction perpendicular to the partitioning line. This soft weighting enables us to efficiently recoup the benefits of fine geometric partitioning with only a few coarse partitions.

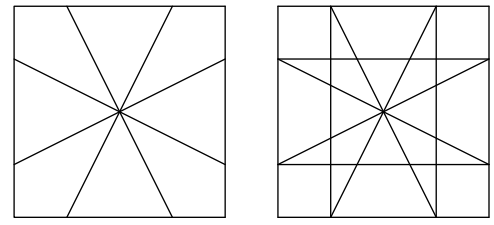

Fig. 4. Candidate edges for 8×8(L) and 16×16(R) blocks.

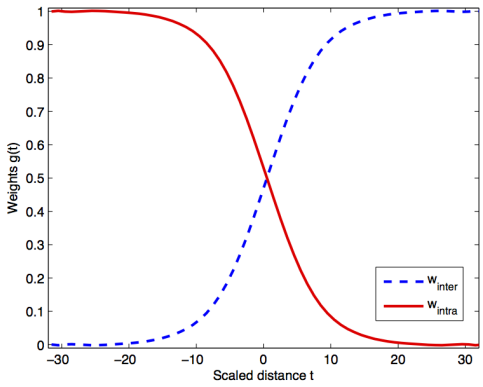

Fig. 5. The weighting function applied to smooth predictions around the edge.

3.3. Algorithm Implementation

Ideally the best joint inter-intra predictor for a given block size is obtained by a 3-way search over the intra mode I_{opt}, the best mask k_{opt} and the best motion vector - reference frame combination $(\overrightarrow{v}, ref)$. Note that the motion estimation needs to be conducted only for the part of the wedge defined by k_{opt} that contributes an inter-predictor, rather than

for the whole block. This is referred to as masked motion search, and it can be implemented either using soft weights or hard weights, with the former yielding somewhat better performance. Likewise the search for the best intra mode needs to only consider the part of the block that contribute to the intra predicted wedge within the block. Unfortunately, this 3-way search if done exhaustively will be too demanding for the encoder. In order to keep the encoding complexity manageable without running the risk of losing too much in quality, we propose a sub-optimal fast search strategy where we use full-block searches combined with masked motion search only once per block as follows:

1. Determine the initial motion vector $\overrightarrow{v}_0$for the whole block and the best intra mode I_0 for intra-only prediction for the whole block.
2. Generate the joint predictors for all candidate masks using the motion vector $\overrightarrow{v}_0$ and the intra mode I_0. Then pick the R-D optimal mask k_{opt} as the one that yields the best R-D cost among these.
3. Search the optimal intra mode I_{opt} for the joint predictor using the mask k_{opt} and the motion vector $\overrightarrow{v}_0$.
4. Run a masked motion search for the preditor using the intra mode I_{opt} and the mask k_{opt}. The motion vector is refined as $\overrightarrow{v}_{opt}$.
5. The final joint prediction is defined by I_{opt}, k_{opt} and $\overrightarrow{v}_{opt}$.

4. EXPERIMENTAL RESULTS

For comparison, the above joint inter-intra prediction scheme was implemented by modifying the VP9 framework, and incorporating them as configurable experiments (*–enable-interintra –enable-masked_interintra*) in the experimental branch of the libvpx repository of the WebM project[11]. Joint predictors with 10 mode-directed weights and edge-directed weights whose codebook sizes vary by block-sizes (see Table 1). In our experiment, high profile of VP9 is used, with quad-tree partitioning (block-sizes ranging from 4×4 to 64×64), compound inter prediction, full motion search for all inter modes except for joint predictors which use fast search, super-pixel precision, and other high-quality features turned on. *30* full $derf$ clips at medium resolutions and *15* full $stdhd$ clips at HD resolutions were coded in IPPP format at target bit-rates ranging from 50kbps to 55000kbps. We evaluated the coding performance by progressively enabling the two proposed techniques. The performance gains, in terms of bit-rate reduction, for some sequences and averaged over the test sets are shown in Table 2 at different qualities. Clearly, both techniques provide consistent gains on top of the reference software. Moreover, the improvements due to the motion-edge-directed weighting, represented by comparing coder B with coder A in Table 2, are obvious at low to medium qualities especially for videos with high motion objects, e.g., $cheer$ and $pedestrian$.

Table 2. Bit-rate reduction due to the proposed joint prediction approaches relative to the VP9 reference software. Coder A: VP9 with only the joint predictor using intra-mode-directed weighting enabled, Coder B: VP9 with both the proposed joint predictors enabled.

Sequence	Coder	PSNR		
		32	36	40
football@CIF	A	0.32	0.78	0.76
	B	1.85	1.93	1.61
foreman@CIF	A	-1.65	0.68	0.97
	B	0.38	0.96	1.35
ice@CIF	A	-0.08	0.13	-0.05
	B	1.99	2.25	0.89
crew@CIF	A	1.42	1.31	2.15
	B	3.23	2.79	2.96
bus@CIF	A	1.16	0.08	0.51
	B	1.49	0.82	0.82
cheer@SIF	A	1.50	1.03	0.67
	B	3.53	1.97	1.19
mobcal@720p	A	0.27	2.17	0.49
	B	0.30	2.13	0.96
oldtown@720p	A	0.28	2.43	0.96
	B	1.64	2.68	3.16
pedestrian@1080p	A	1.76	2.63	2.75
	B	4.51	5.60	4.49
riverbed@1080p	A	4.61	2.36	1.38
	B	4.90	2.51	1.42

Test set	Coder	Average	PSNR		
			32	36	40
derf	A	0.392	0.526	0.270	0.380
	B	0.860	1.070	0.883	0.628
stdhd	A	0.900	0.674	0.619	1.407
	B	1.485	1.532	1.196	1.723

5. CONCLUSION

A novel joint inter-intra prediction scheme, based on two weighting approaches considering the nature of intra prediction or the location of motion edges, is proposed for P frame coding. In the intra-mode-directed approach, exploiting the Markov property of pixels, we design the weights to combine inter and intra predictions as following a 1-D weighting function along the prediction angle. Also we recoup the benefits of geometric partition, while without adding too much complexity, by creating a partition codebook and employing a soft masking technique. We have shown that the use of intra-mode-variant weighting, together with motion-edge-directed weighting supported by the soft masking technique, provides considerable coding gains.

6. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] J. Bankoski, R.S. Bultje, A. Grange, Q. Gu, J. Han, J. Koleszar, D. Mukherjee, P. Wilkins, and Y. Xu, "Towards a next generation open-source video codec," *IS&T/SPIE Electronic Imaging*, 2013.

[3] J. Xin, K. N. Ngan, and G. Zhu, "Combined inter-intra prediction for high definition video coding," *Picture Coding Symposium*, 2007.

[4] R. Cha, O. Au, X. Fan, X. Zhang, and J. Li, "Improved combined inter-intra prediction using spatial-variant weighted coefficient," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[5] J. Seiler and A. Kaup, "Spatio-temporal prediction in video coding by spatially refined motion compensation," *IEEE International Conference in Image Processing (ICIP)*, pp. 2788–2791, 2008.

[6] J. Seiler, H. Lakshman, and A. Kaup, "Spatio-temporal prediction in video coding by best approximation," *Picture Coding Symposium*, pp. 1–4, 2009.

[7] J. Seiler, T. Richter, and A. Kaup, "Spatio-temporal prediction in video coding by non-local means refined motion compensation," *Picture Coding Symposium*, pp. 318–321, 2010.

[8] Minh N Do, Pier Luigi Dragotti, Rahul Shukla, and Martin Vetterli, "On the compression of two-dimensional piecewise smooth functions," in *Image Processing, 2001. Proceedings. 2001 International Conference on*. IEEE, 2001, vol. 1, pp. 14–17.

[9] R. Ferreira, E. Hung, R. de Queiroz, and D. Mukherjee, "Efficiency improvements for a geometric-partition-based video coder," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1009–1012.

[10] O. Divorra Escoda, P. Yin, C. Dai, and X. Li, "Geometry-adaptive block partitioning for video coding," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 1.

[11] "The webm project," http://www.webmproject.org.