REDUCED-RANK CONDENSED FILTER DICTIONARIES FOR INTER-PICTURE PREDICTION

Shunyao Li [*], *Onur G. Guleryuz* [†], *and Sehoon Yea* [†]

[*]U. C. Santa Barbara, Santa Barbara, CA, USA
[†]LG Electronics, Mobile Research Lab, San Jose, CA, USA

ABSTRACT

We consider the motion-compensated temporal prediction loop at the heart of modern video coders. Rather than using motion-compensated reference frame blocks directly as predictors, we incorporate their spatially-filtered versions into the prediction loop. We design adaptive filters that are geared toward successful prediction over sophisticated temporal evolutions involving lighting changes, focus changes, structured noise, and so on. The spatially and temporally varying nature of such video evolutions requires the learning and transmission of many filters, necessitating parameter reduction for compression and related applications. Unlike earlier work that tries to limit parameters by using a small set of general filters, or by restricting to symmetric filters, etc., we propose a novel parametrization of filters in terms of a set of base-filter kernels and modulation weights. Given a filter dictionary of K-tap filters, our work can be seen as providing a reduced-rank, *prediction-optimal* approximation of this dictionary that represents its filters with $K' << K$ parameters.

Index Terms— low rank decompositions, prediction filter, inter-picture prediction, HEVC, VP9

1. INTRODUCTION

Video sequences exhibit many types of temporal evolutions that fall outside of the white noise displaced-frame-difference model. Researchers have thus devised many different spatio-temporal formulations that estimate statistical dependencies and form associated predictors [1, 7, 13, 2, 12]. Since the highly transient nature of video evolutions makes learning of densely parameterized models difficult, established techniques concentrate on simplified transitions and derive practical pel-recursive estimators, Kalman, and Wiener filters [1, 14, 7, 8, 13]. Other pixel-domain formulations focus on interpolation errors, aliasing errors, and specific forms of brightness changes [3, 19, 18, 6, 9].

Motivated by transform domain sparsity, [5] has proposed inter-picture prediction in transform-domain that showcases high performance results over a wide variety of evolutions. With data sparsifying transforms providing the main statistical modeling, simple predictors in transform domain are shown to be adequate for many evolutions. When using localized transforms and translation invariant decompositions the authors can show spatial filtering analogues to their work in terms of filters defined via the transform basis.

The work we present in this paper can be seen as unifying the aforementioned pixel and transform domain motivated work through an adaptive filter parametrization and an associated optimization process. Formulating inter-picture prediction as the spatially adaptive filtering of a reference picture to estimate the picture to be predicted, we propose filters with reduced parametrizations that preserve a high degree of adaptivity. Our work optimally discovers the form of reduced parameterizations that in cases may result in symmetric filters (reduced number of unique tap values), frequency constrained filters (reduced number of unique frequency parameters), etc. In effect, we design the optimal linear reduction of K-tap prediction filters to filters described by K' parameters and determine whether this parameterization should be in pixel-domain (e.g., in terms of tap values) or in transform domain (e.g., in terms of a predefined transform basis), and so on.

The outline of the paper is as follows. Section 2 discusses the basic ideas and defines the condensed prediction filters. A joint optimization framework for the base filter kernels and modulation weights is provided in Section 3. Section 4 discusses simulation results and concludes the paper.

2. BASIC IDEAS

2.1. Inter-Picture Evolution Model

Let $\mathcal{Y}$ denote the current video frame. Consider the motion compensated prediction of $\mathcal{Y}$ using the previously decoded reference frame $\mathcal{X}$ as employed in modern video coders [10, 17, 11]. For convenience of notation concentrate on block y from $\mathcal{Y}$ which is matched via motion estimation to block x from $\mathcal{X}$ [1]. Assume that x and y are lexicographically ordered into $(N \times 1)$ vectors. With some abuse of notation let $f * x$ be the vector that is formed by lexicographically ordering the convolution of the filter f with block x. In this paper we are interested in the temporal evolution model,

$$y = f * x + w, \tag{1}$$

[1]Later sections will incorporate the proposed work within the motion estimation loop. For the time being assume that motion is adequately compensated for and x is of appropriate size to allow correctly filtered values at boundaries. Assume also that any needed de-blocking on the reference frame has already been accomplished.

where f is a $2D$ linear filter and w is white noise.

Observe that while (1) appears to be a "limited linear" model, when one considers the adaptivity provided by f and the block size, it becomes clear that the model is very general and can accomplish most temporal evolutions with ease[2]. Hence, in order to make the problem more concrete, consider the filter dictionary $\mathcal{F} = \{f_1, \ldots, f_M\}$ and the triplet $(y,\ x,\ \mathcal{I})$ so that,

$$\mathcal{I} = m \ \Rightarrow\ y = f_m * x + w. \tag{2}$$

Assume that the filters in $\mathcal{F}$ have K taps. The convolution in (2) can be written as a sequence of scalar products of f_m with overlapping regions in x. Assume each such region is lexicographically ordered into a vector and the resulting vectors are transposed and collected into the rows of the matrix $\boldsymbol{X}$. The model can now be stated as,

$$\mathcal{I} = m \ \Rightarrow\ y = \boldsymbol{X} f_m + w. \tag{3}$$

2.2. Reduced-Rank Parametrization

Let $\boldsymbol{G}$ be the $(K \times K')$ *base filter kernel matrix* with $K' \leq K$. In this paper we approximate the K-tap filter f_m via

$$f_m \ \rightarrow \boldsymbol{G} c_m, \tag{4}$$

where c_m is the $(K' \times 1)$ modulation weight vector. Observe that symmetric filters are obtained if the filters corresponding to columns of $\boldsymbol{G}$ are symmetric, and likewise transform-based filters result when $\boldsymbol{G}$'s columns are defined using transform basis, etc. It is hence clear that a framework that finds the optimal base filter kernels also determines the form of the optimal parameter reduction.

Let $\boldsymbol{F}$ be the $(K \times M)$ matrix whose m^{th} column is f_m. Similarly let $\boldsymbol{C}$ be the $(K' \times M)$ matrix whose m^{th} column is c_m. With the above parametrization it is clear that

$$\boldsymbol{F} \rightarrow \boldsymbol{G}\boldsymbol{C}, \tag{5}$$

i.e., the prediction filter dictionary is approximated using a reduced rank decomposition. We call the resulting dictionary the condensed filter dictionary and will refer to the filters as *condensed prediction filters* (CPF).

2.3. Scaling the Factorization

Since the condensed dictionary is in the form of a factorization, multiplying $\boldsymbol{C}$ by an invertible matrix results in the same CPFs if $\boldsymbol{G}$ is multiplied with the inverse matrix, i.e., $\boldsymbol{G}\boldsymbol{C} = (\boldsymbol{G}\boldsymbol{S}^{-1})(\boldsymbol{S}\boldsymbol{C})$. In a typical application $\boldsymbol{G}$ is fixed or changed infrequently while $\boldsymbol{C}$ provides adaptation. It is hence convenient to scale based on properties desired of $\boldsymbol{C}$.

One such property is that each column of $\boldsymbol{C}$ should be as decorrelated as possible so that the modulation weights can be transmitted with few bits. This can be accomplished by setting $\boldsymbol{S}$ to be the Karhunen-Loeve Transform obtained by the eigen-decomposition of $\boldsymbol{C}\boldsymbol{C}^T$. A better alternative is to set $\boldsymbol{S}$ to be the Sparse Orthonormal Transform [16, 15] in order to ensure each column of $\boldsymbol{C}$ has the fewest number of non-zero values. One can also scale the base filter modulation kernels or the modulation weights to have unit norm.

[2]In particular note that in the trivial but conceptually important limit of a single-pixel block, one can derive an adaptive single-tap filter $f = (y-w)/x$ allowing perfect prediction for $x \neq 0$.

2.4. Spatial Variation

Similar to the block-motion field assumptions prevalent in video coding, in this paper we assume that the filters are defined with the aid of a block-evolution field that assigns a filter parameter to blocks in $\mathcal{Y}$. Hence $\mathcal{I}$ is assumed to be defined over spatial blocks, assigning filters to blocks of varying sizes. Part of the task of the proposed work will be to identify this block decomposition and the filter dictionary.

3. OPTIMIZATION OF CPF

3.1. Optimizing Modulation Weights

Consider the model in (3) with $\mathcal{I} = m$ so that $y = \boldsymbol{X} f_m + w$. Assume zero-mean quantities and let $E[.|m] = E[.|\mathcal{I} = m]$ denote expectation conditioned on $\mathcal{I} = m$. Bold upper-case letters denote matrices. The conditional mean squared error (mse) is given by,

$$E[||y - \boldsymbol{X}\boldsymbol{G}c_m||_2^2|m] = E[y^T y|m] - 2E[y^T \boldsymbol{X}|m]\boldsymbol{G}c_m + c_m^T \boldsymbol{G}^T E[\boldsymbol{X}^T \boldsymbol{X}|m]\boldsymbol{G}c_m. \tag{6}$$

Minimizing (6) in terms of the modulation weights, we obtain

$$\boldsymbol{G}^T E[\boldsymbol{X}^T y|m] = \boldsymbol{G}^T E[\boldsymbol{X}^T \boldsymbol{X}|m]\boldsymbol{G}c_m. \tag{7}$$

Observe that since the noise is white we have

$$E[\boldsymbol{X}^T y|m] = E[\boldsymbol{X}^T \boldsymbol{X}|m] f_m. \tag{8}$$

Assume that $E[\boldsymbol{X}^T \boldsymbol{X}|m]$ is independent of m so that the $x,\ y$ dependency is primarily in terms of f_m, i.e.,

$$E[\boldsymbol{X}^T \boldsymbol{X}|m] \ = \ E[\boldsymbol{X}^T \boldsymbol{X}] = \boldsymbol{R}, \tag{9}$$

$$E[\boldsymbol{X}^T y|m] \ = \ \boldsymbol{R} f_m. \tag{10}$$

This is a reasonable assumption since the temporal change to the next frame, captured by f_m, is typically independent of the reference frame statistics. For example, the same reference block can be imagined to undergo different lighting changes, different focus changes, etc., with the said changes independent of its statistics. Using matrix notation to accommodate $m = 1, \ldots, M$, (7) becomes,

$$\boldsymbol{G}^T \boldsymbol{R}\boldsymbol{F} = \boldsymbol{G}^T \boldsymbol{R}\boldsymbol{G}\boldsymbol{C}. \tag{11}$$

3.2. Optimizing Base Filter Kernels

Plugging (7) into the conditional mse in (6) we obtain,

$$E[||y - \boldsymbol{X}\boldsymbol{G}c_m||_2^2|m] = E[y^T y|m] - f_m^T \boldsymbol{R}\boldsymbol{G}c_m. \tag{12}$$

Assume without loss of generality that different filters are equally likely. The overall mse then becomes

$$\begin{aligned} & E[E[||y - \boldsymbol{X}\boldsymbol{G}c_m||_2^2|m]] \\ & = E[y^T y] - \frac{1}{M}\sum_m f_m^T \boldsymbol{R}\boldsymbol{G}c_m, \\ & = E[y^T y] - \frac{1}{M} Tr[\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{G}\boldsymbol{C}], \end{aligned} \quad (13)$$

where $Tr[.]$ denotes the trace of a matrix. The optimal base filter kernels are thus obtained as the $\boldsymbol{G}$ that maximizes $Tr[\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{G}\boldsymbol{C}]$ subject to (11).

3.3. Joint Optimization

The joint optimization problem can now be stated as

$$\max_{\boldsymbol{G},\boldsymbol{C}} Tr[\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{G}\boldsymbol{C}] \text{ subject to } \boldsymbol{G}^T\boldsymbol{R}\boldsymbol{F} = \boldsymbol{G}^T\boldsymbol{R}\boldsymbol{G}\boldsymbol{C}. \quad (14)$$

Since the CPF is given by the $\boldsymbol{G}\boldsymbol{C}$ product it is clear that one can scale the rows of $\boldsymbol{C}$ and inverse scale the columns of $\boldsymbol{G}$ to arrive at the same filters. Suppose $\boldsymbol{C}$ is scaled so that it has unit energy rows. The below proposition is straightforward.

Proposition 3.1 *Suppose $\boldsymbol{C}$ has unit energy rows. Then, $\max_{\boldsymbol{C}} Tr[\boldsymbol{A}\boldsymbol{C}] = \max_{\boldsymbol{C}} Tr[\boldsymbol{C}\boldsymbol{A}]$ is obtained via*

$$\boldsymbol{C} = \boldsymbol{D}\boldsymbol{A}^T, \quad (15)$$

$$D_{i,j} = \begin{cases} 1/(\sum_k (A_{k,i})^2)^{1/2} & i = j, \\ 0 & i \neq j. \end{cases} \quad (16)$$

Using Proposition 3.1 with (14) we have that the optimal $\boldsymbol{C} = \boldsymbol{D}\boldsymbol{G}^T\boldsymbol{R}\boldsymbol{F}$, with $\boldsymbol{D}$ obtained using the proposition ($\boldsymbol{A} = \boldsymbol{F}^T\boldsymbol{R}\boldsymbol{G}$), provided that the constraint $\boldsymbol{G}^T\boldsymbol{R}\boldsymbol{F} = \boldsymbol{G}^T\boldsymbol{R}\boldsymbol{G}\boldsymbol{C}$ can be satisfied. Let us now see that this is the case.

With the above optimal $\boldsymbol{C}$ the constraint in (14) becomes

$$\begin{aligned} \boldsymbol{G}^T\boldsymbol{R}\boldsymbol{F} &= \boldsymbol{G}^T\boldsymbol{R}\boldsymbol{G}\boldsymbol{D}\boldsymbol{G}^T\boldsymbol{R}\boldsymbol{F}, \\ \tilde{\boldsymbol{G}}^T\boldsymbol{R}\boldsymbol{F} &= \tilde{\boldsymbol{G}}^T\boldsymbol{R}\tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^T\boldsymbol{R}\boldsymbol{F}, \end{aligned} \quad (17)$$

where $\tilde{\boldsymbol{G}} = \boldsymbol{G}\boldsymbol{D}^{1/2}$. Suppose $\tilde{\boldsymbol{G}}^T\boldsymbol{R}\boldsymbol{F}$ is full rank. Then the constraint is satisfied provided that $\tilde{\boldsymbol{G}}^T\boldsymbol{R}\tilde{\boldsymbol{G}} = \boldsymbol{1}$, where $\boldsymbol{1}$ is the $(K' \times K')$ identity. Since $\boldsymbol{R} = E[\boldsymbol{X}^T\boldsymbol{X}]$ is a covariance matrix, let $\boldsymbol{R} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ be its eigen decomposition with orthonormal $\boldsymbol{V}$ of eigenvectors and diagonal $\boldsymbol{\Lambda}$ of eigenvalues. In order to meet the constraint we hence need

$$\boldsymbol{1} = \tilde{\boldsymbol{G}}^T\boldsymbol{R}\tilde{\boldsymbol{G}} = \tilde{\boldsymbol{G}}^T\boldsymbol{V}^T\boldsymbol{\Lambda}\boldsymbol{V}^T\tilde{\boldsymbol{G}},$$

which leads to

$$\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T\tilde{\boldsymbol{G}} = \boldsymbol{H}, \quad (18)$$

where $\boldsymbol{H}$ is a $(K \times K')$ matrix having orthonormal columns. Plugging into the trace expression in (14) results in the trace

$$\begin{aligned} & Tr[\boldsymbol{F}^T\boldsymbol{R}\tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^T\boldsymbol{R}\boldsymbol{F}] \\ & = Tr[\boldsymbol{F}^T\boldsymbol{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T\boldsymbol{F}] \\ & = Tr[\boldsymbol{H}^T\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T\boldsymbol{F}\boldsymbol{F}^T\boldsymbol{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}]. \end{aligned} \quad (19)$$

The following straightforward proposition summarizes the result that we need.

Proposition 3.2 *Suppose $\boldsymbol{Q}$ $(K \times K)$ is a symmetric positive semi-definite matrix. Let $\boldsymbol{Q} = \boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{W}^T$ be its eigen decomposition with the diagonal $\boldsymbol{\Gamma}$ containing the eigenvalues in nonincreasing order, i.e., $\Gamma(1,1) \geq \Gamma(2,2) \geq \ldots \geq \Gamma(K,K)$. Consider $(K \times K')$ orthonormal matrices $\boldsymbol{H}$ with $K' \leq K$. Then*

$$\max_H Tr[\boldsymbol{H}^T\boldsymbol{Q}\boldsymbol{H}] = \sum_{l=1}^{K'} \Gamma(l,l), \quad (20)$$

which can be accomplished by setting $H(i,j) = V(i,j)$, $i = 1,\ldots,K, j = 1,\ldots,K'$.

It is now easy to see that the $\boldsymbol{H}$ that maximizes (19) (up to repeated eigenvalues) corresponds to the K' eigenvectors of,

$$\boldsymbol{Q} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T\boldsymbol{F}\boldsymbol{F}^T\boldsymbol{V}\boldsymbol{\Lambda}^{1/2}, \quad (21)$$

that correspond to the largest K' eigenvalues. We hence have,

Proposition 3.3 *Let $\boldsymbol{R} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ be a $(K \times K)$ covariance matrix and let $\boldsymbol{F}$ be a $(K \times M)$ bank of filters. Set $\boldsymbol{Q} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T\boldsymbol{F}\boldsymbol{F}^T\boldsymbol{V}\boldsymbol{\Lambda}^{1/2}$. Let $\boldsymbol{H}$ $(K \times K')$ be the eigenvectors of $\boldsymbol{Q}$ that are associated with the largest K' eigenvalues. The joint optimization problem in (14) is then solved using*

$$\begin{aligned} \tilde{\boldsymbol{G}} &= \boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{H}, \\ \boldsymbol{G}\boldsymbol{C} &= \tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^T\boldsymbol{R}\boldsymbol{F}. \end{aligned} \quad (22)$$

4. RESULTS

4.1. Implementation Details

We implemented our work within HEVC reference software (HM-14.0) configured for low-delay (IPP...). Each prediction unit (PU) of the reference software was modified to use the proposed work so that reference frame blocks corresponding to PUs (found using PU motion vectors) were filtered using CPFs. As HEVC PUs can be considerably large we derived a CPF quad-tree within each PU using CART-like tree-pruning optimization. Each leaf-node of the CPF quad-tree is assigned a modulation weight, c, and a CPF given by $\boldsymbol{G}c$. $\boldsymbol{G}$ is fixed for the entire sequence. For the below results we constrained $\boldsymbol{G}$ so that $\boldsymbol{G}u = \delta$ where u is the vector of all ones, and delta is the 2-D impulse, i.e., $x * \delta = x$. This in effect allows for the cases that require no filtering to be represented with $c = u$, allowing the CPF to be "ON" for all PUs without requiring the implementation of a "CPF ON/OFF" codec flag.

Given $\boldsymbol{G}$, the optimal CPF quad-tree and the associated modulation weights were determined by minimizing equations of the form,

$$\min_c ||y - \boldsymbol{X}\boldsymbol{G}c||_p + \gamma||c||_1, \quad (23)$$

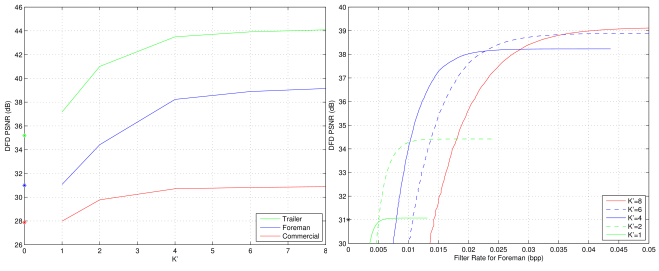
where $p = 1$ for SAE or $p = 2$ for SSE, and $||c||_1$ measures transmission filter rate by assuming a Laplacian model for the quantized modulation weights. The filter rate as well as the rate for specifying the quad-tree were added to the motion estimation optimization loop within HEVC. The decoder applied filtering using the CPF, $\boldsymbol{G}c$, over decoded CPF quad-trees. For the following results we determined a CPF quad-tree using an initial $\boldsymbol{G}_{dct}$ (taken to be transform basis induced filters introduced in [5] corresponding to (5×5) filters, i.e, $K = 25$) and kept this quad-tree fixed. The prediction filter dictionary, $\boldsymbol{F}$, required in our optimizations was obtained through (10).

4.2. Simulations

Figure 1 illustrates the three test sequences we have used (100 frames, CIF). Both the movie trailer and the commercial contain many lighting changes, cross-fades, and changes of focus. Commercial also contains significant noise-like transients (rain effects, etc.) Foreman has strong directional structures which typically lead to directional error patterns.

Fig. 1. Test sequences Trailer, Commercial, and Foreman.

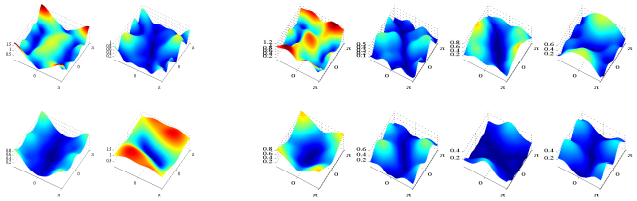
We first study the displaced frame difference (DFD) distortion as a function of K' keeping the CPF quad-trees and QP fixed. Figure 2 (a) shows DFD distortion for the above three sequences. PSNR is calculated for the entire sequence and normalized per-pixel. The points at $K' = 0$ correspond to no filtering applied, whereas $K' = 1$ correspond to the impulse filter multiplied by a scalar, in effect to pixel-based intensity compensation. Observe that while intensity compensation doesn't provide much improvement, as K' is increased, CPFs obtain significant gains. Note also that since the filtering in spatial domain is 25-tap, i.e., $K = 25$, CPFs provide very significant reduction in parameters for given distortion, especially for $K' = 4$.



(a) (b)

Fig. 2. (a) DFD distortion as a function of K'. (b) DFD distortion as a function of filter rate for Foreman.

The parameter reduction $K \rightarrow K'$ is one way CPFs provide compression. Figure 2 (b) illustrates the impact of the derived CPFs on bit-rate for Foreman. The modulation weights were scalar quantized using the codec's transform coefficient quantizer step-size, inverse scaled with the l_2-norm of each filtered reference block, i.e, each modulation weight is thought of as a transform coefficient associated with a basis that is the reference block filtered with the corresponding base filter kernel (more elaborate quantization can also be employed [4]). Entropy based on the aforementioned Laplacian model measures rate. Observe again that CPFs (illustrated in Figure 3) provide significant improvements with marginal increases in rate, especially with $K' = 4$.



(a) $\boldsymbol{G}_F$, $(K' = 4)$ (b) $\boldsymbol{G}_F$, $(K' = 8)$

Fig. 3. Fourier trf. magnitude of the kernels for Foreman.

	$(K' = 4)$			$(K' = 8)$		
	$\boldsymbol{G}_T$	$\boldsymbol{G}_C$	$\boldsymbol{G}_F$	$\boldsymbol{G}_T$	$\boldsymbol{G}_C$	$\boldsymbol{G}_F$
Trailer	43.5	43.3	43.3	44.1	43.9	43.9
Commercial	30.5	30.7	30.4	30.8	30.9	30.7
Foreman	38.0	37.6	38.2	38.8	38.4	39.1

Table 1. Generalizability of the base filter kernels. The DFD PSNR (dB) as a function of the utilized base filter kernels.

Let us now look at Table 1 which illustrates how the base filter kernels specific to each sequence perform when used in establishing the filtering for the others. This experiment scrutinizes the feasibility of finding a universal $\boldsymbol{G}$ to be used over a variety of temporal evolutions. For this purpose we solve for a sequence specific $\boldsymbol{C}$ using the given $\boldsymbol{G}$ via (11). It is clear that the base filter kernels readily translate from one sequence to the next and most of the adaptivity is captured through the modulation weights.

4.3. Conclusion

We proposed prediction-optimal, reduced-rank parametrizations of filter dictionaries targeting inter-picture prediction. Our work allows a high degree of filter adaptivity using few parameters and obtains significant increases in DFD PSNR at a given bit-rate. Experiments show that the derived optimal base filter kernels provide a subspace which is generalizable across video sequences and most of the filter adaptivity is accomplished by the modulation weights within this subspace. Future work will concentrate on improved ways of encoding the modulation weights and their across-PU prediction.

5. REFERENCES

[1] J. Biemond, L. Looijenga, D. Boekee, and R. Plompen. A pel-recursive wiener-based motion estimation algorithm. *Signal Processing*, 13:399–412, December 1987.

[2] S. Efstratiadis and A. Katsaggelos. A model-based pel-recursive motion estimation algorithm. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 1973–1976 vol.4, Apr 1990.

[3] B. Girod. Efficiency analysis of multihypothesis motion-compensated prediction for video coding. *Image Processing, IEEE Transactions on*, 9(2):173–183, Feb 2000.

[4] O. G. Guleryuz, A. Said, and S. Yea. Non-causal encoding of predictively coded samples. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 4812–4816, Oct 2014.

[5] G. Hua and O. Guleryuz. Spatial sparsity-induced prediction (sip) for images and video: A simple way to reject structured interference. *Image Processing, IEEE Transactions on*, 20(4):889–909, April 2011.

[6] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose. Global brightness-variation compensation for video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(8):988–1000, Dec 1998.

[7] J. Kim and J. Woods. Spatio-temporal adaptive 3-d kalman filter for video. *Image Processing, IEEE Transactions on*, 6(3):414–424, Mar 1997.

[8] J. Kim and J. Woods. 3-d kalman filter for image motion estimation. *Image Processing, IEEE Transactions on*, 7(1):42–52, Jan 1998.

[9] H. Lee, S.-C. Lim, H. Choi, S. Jeong, J. Kim, and J. S. Choi. Enhanced block-based adaptive loop filter with multiple symmetric structures for video coding. *ETRI Journal*, 32(4):626–629, Aug 2010.

[10] D. Marpe, T. Wiegand, and G. Sullivan. The h.264/mpeg4 advanced video coding standard and its applications. *Communications Magazine, IEEE*, 44(8):134–143, Aug 2006.

[11] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje. The latest open-source video codec vp9 - an overview and preliminary results. In *Picture Coding Symposium (PCS), 2013*, pages 390–393, Dec 2013.

[12] A. Nosratinia and M. Orchard. New relationships in operator-based backward motion compensation. In *Data Compression Conference, 1995. DCC '95. Proceedings*, pages 391–400, Mar 1995.

[13] M. Ozkan, A. Erdem, M. Sezan, and A. Tekalp. Efficient multiframe wiener restoration of blurred and noisy image sequences. *Image Processing, IEEE Transactions on*, 1(4):453–476, Oct 1992.

[14] R. Rajagopalan, M. Orchard, and R. Brandt. Motion field modeling for video sequences. *Image Processing, IEEE Transactions on*, 6(11):1503–1516, Nov 1997.

[15] O. Sezer and O. Guleryuz. Approximation and compression with sparse orthonormal transforms. *Image Processing, IEEE Transactions on (to appear)*.

[16] O. Sezer, O. Harmanci, and O. Guleryuz. Sparse orthonormal transforms for image compression. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 149–152, Oct 2008.

[17] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1649–1668, Dec 2012.

[18] Y. Vatis, B. Edler, I. Wassermann, D. T. Nguyen, and J. Ostermann. Coding of coefficients of two-dimensional non-separable adaptive wiener interpolation filter, 2005.

[19] T. Wedi. Adaptive interpolation filter for motion and aliasing compensated prediction, 2002.