Entropy-Constrained Linear Vector Prediction for Motion-Compensated Video Coding

Thomas Wiegand,[†] Markus Flierl, and Bernd Girod[‡]
Telecommunications Institute I
University of Erlangen-Nuremberg
Cauerstr. 7, D-91058 Erlangen, Germany
{wiegand|flierl|girod}@nt.e-technik.uni-erlangen.de

Abstract

We extend the rate distortion theory of motion-compensated prediction to linear predictive models. The power spectrum of the motion-compensated prediction error is related to the displacement error pdfs of an arbitrary number of linear predictor input signals in a closed form expression. The influence of the residual noise level and the gains achievable are investigated. We then extend the scalar approach to motion-compensated vector prediction. The vector predictor coefficients are fixed, but we conduct a search to find the optimum input vectors. We control the rate of the motion compensation data which have to be transmitted as side information to the decoder by minimizing a Lagrangian cost function where the regularization term is given by the entropy associated with the motion compensation data. An adaptive algorithm for optimally selecting the size of the linear vector predictor is given. The designed motion-compensated vector predictors show PSNR gains up to 4.4 dB at the cost of increased bit-rate of 16 kbit/s when comparing them to conventional motion-compensated prediction.

1 Introduction

Block-based motion-compensating video coding schemes achieve data compression by exploiting dependencies between successive frames in the video signal. In order to predict a block in the current frame, a set of past, decoded blocks is searched for the best mapping. It has been observed that linear combinations of past, decoded blocks can reduce the prediction error significantly. The term multi-hypothesis motion-compensated prediction has been coined for this approach in video coding. Examples are B-frames [1] or overlapped block motion-compensation [2]. Multi-hypothesis motion-compensated prediction can be viewed as a special case of linear vector prediction. Thus, we will refer to the linear predictor input signals as hypotheses as well. Linear vector prediction was introduced in [3] and [4] in the context of predictive vector quantization (PVQ) [5]. PVQ is the extension of predictive scalar quantization or differential pulse code modulation (DPCM) to vector valued signals. PVQ has mainly been exploited in speech coding [6].

In this paper, we present both theoretical and experimental results for multi-hypotheses motion-compensated prediction of video signals. Following the theoretical framework for the rate distortion analysis of motion-compensated prediction in [7] and [8], we will derive a closed form expression, where the power spectrum of the prediction error is related to the displacement pdfs of an arbitrary number of scalar input signals to the linear predictor. While in [7] and [8] mainly the effect of prediction accuracy is discussed, we focus in this paper on the effect of number of input signals vs. residual noise level of motion-compensated linear prediction. Previous work of Girod on the subject can be found in [9].

In this work as well as in [9], the bit-rate to transmit the motion-related information is initially neglected. Since motion compensation has to be performed simultaneously at encoder and decoder, the required information

[†]Thomas Wiegand is currently Visiting Researcher at the Department of Electrical Engineering, Stanford University, USA.
[‡]Bernd Girod is currently Visiting Professor at the Department of Electrical Engineering, Stanford University, USA.

needs to be transmitted to the decoder. Hence, motion compensation is performed using blocks instead of scalars in order to lower the bit-rate of the motion information, extending our approach to motion-compensated linear vector prediction. Even then, the motion vector bit-rate may be too high when we consider for example $N = 4$ input vectors and a very low bit-rate application is targeted. Hence, the selection of the motion-compensated linear vector predictors requires a rate distortion framework. We interpret motion compensation as minimum distortion vector quantization of a current image block given its (quantized) past as the code book, subject to a rate constraint. Hence, we can make use of the insights learnt from entropy-constrained vector quantization (ECVQ) [10]. The vectors in our ECVQ interpretation of block-based motion-compensated prediction are image blocks in the video frame that is to be transmitted. The image blocks are vector quantized using individual code books that consist of image blocks of the same size in the previously decoded frames. A code book entry is addressed by translational motion parameters pointing into the past frames which are entropy-coded.

This paper is organized as follows. In section 2, we derive a theoretical model for motion-compensated linear prediction. We evaluate the obtained results numerically and obtain insights into the prediction gains of our approach. The framework for our extension of motion-compensated linear prediction to vector valued signals is introduced in section 3. An algorithm for motion-compensated linear vector predictor is given in section 4. Finally, we incorporate an entropy constraint into the algorithm for motion-compensated linear vector prediction in section 5. We derive an algorithm for adaptively selecting the optimum number of hypotheses per block and demonstrate its performance.

2 A Model for Motion-Compensated Linear Prediction

Let $s[x, y]$ be a scalar two-dimensional signal sampled on an orthogonal grid with horizontal spacing X and vertical spacing Y. We model motion-compensated prediction as the prediction of the signal s by modified versions of itself $\hat{s}_i$. The $\hat{s}_i$ are modified from s in that they are shifted by arbitrary displacements and corrupted by additive noise z_i. The "noise" signal z_i is drawn from a stationary, white random process and is assumed to be uncorrelated to s. It comprises all signal components that cannot be described by a translatory displacement model including camera noise, quantization noise, illumination changes, resolution changes, and so on. Motion-compensation is assumed to be the alignment of the $\hat{s}_i$ to s up to a certain accuracy producing the motion-compensated signals c_i. The alignment data, i.e., the displacement vector field, can never be completely accurate since it has to be transmitted as side information. More precisely,

$$c_i[x, y] = s[x - \Delta_{xi}, y - \Delta_{yi}] + z_i[x, y]. \tag{1}$$

with Δ_{xi} and Δ_{yi} being the horizontal and vertical displacement error. Let the c_i be collected in $\mathbf{c} = (c_1, \ldots, c_N)$ and the z_i in $\mathbf{z} = (z_1, \ldots, z_N)$. Assume that s and with that $\mathbf{c}$ are generated by a jointly wide-sense stationary random process with the real valued power spectral density $\Phi_{ss}(\omega_x, \omega_y)$, the $N \times N$ power spectral density matrices $\Phi_{\mathbf{cc}}(\omega_x, \omega_y)$ and $\Phi_{\mathbf{zz}}(\omega_x, \omega_y)$, as well as the $N \times 1$ cross spectral density vector $\Phi_{\mathbf{c}s}(\omega_x, \omega_y)$. Power spectra and cross spectra are defined according to

$$\Phi_{\mathbf{ab}}(\omega_x, \omega_y) = \mathcal{F}_*\{E\{\mathbf{a}[x + x', y + y']\mathbf{b}^H[x, y]\}\}, \tag{2}$$

where the superscript $\mathbf{b}^H$ denotes the transposed complex conjugate of $\mathbf{b}$, $[x, y] \in \Pi$ are the sampling locations. $E\{\mathbf{a}[x + x', y + y']\mathbf{b}^H[x, y]\}$ is the matrix of space-discrete cross correlation functions between $\mathbf{a}$ and $\mathbf{b}$ which for wide-sense stationary random processes does not depend on x and y but only on the relative shifts x' and y'. Finally, $\mathcal{F}_*\{\cdot\}$ is the 2-D band-limited discrete-space Fourier transform

$$\mathcal{F}_*\{\cdot\} = \sum_{[x,y]\in\Pi} (\cdot) e^{-j\omega_x \frac{x}{X} - j\omega_x \frac{y}{Y}} \quad \forall \quad |\omega_x| < \pi, |\omega_y| < \pi. \tag{3}$$

With the definition

$$\mathbf{D} = \begin{pmatrix} e^{-j\omega_x \Delta_{x1}/X - j\omega_y \Delta_{y1}/Y} \\ \vdots \\ e^{-j\omega_x \Delta_{xi}/X - j\omega_y \Delta_{yi}/Y} \\ \vdots \\ e^{-j\omega_x \Delta_{xN}/X - j\omega_y \Delta_{yN}/Y} \end{pmatrix} \tag{4}$$

we can write down the cross spectral density vector

$$\Phi_{\mathbf{cs}} = \mathcal{F}_*\{E\{\mathbf{c}[x+x', y+y']s^H[x,y]\}\} = E\{\mathbf{D}\}\Phi_{ss}, \tag{5}$$

and the power spectral density matrix

$$\Phi_{\mathbf{cc}} = \mathcal{F}_*\{E\{\mathbf{c}[x+x', y+y']\mathbf{c}^H[x,y]\}\} = \Phi_{ss}E\{\mathbf{D}\mathbf{D}^H\} + \Phi_{\mathbf{zz}}, \tag{6}$$

recalling that $\mathbf{s}$ and $\mathbf{z}$ are assumed to be uncorrelated. We will omit the independent variables (ω_x, ω_y), when there is no danger of confusion. We observe that

$$E\{D\} = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p(\Delta_{x1}, \Delta_{y1}, \ldots, \Delta_{xN}, \Delta_{yN}) \begin{pmatrix} e^{-j\omega_x\Delta_{x1}/X - j\omega_y\Delta_{y1}/Y} \\ \vdots \\ e^{-j\omega_x\Delta_{xi}/X - j\omega_y\Delta_{yi}/Y} \\ \vdots \\ e^{-j\omega_x\Delta_{xN}/X - j\omega_y\Delta_{yN}/Y} \end{pmatrix} d\Delta_{x1}d\Delta_{y1}\cdots d\Delta_{xN}d\Delta_{yN}$$

$$= \begin{pmatrix} \mathcal{F}\{p(\Delta_{x1}, \Delta_{y1})\} \\ \vdots \\ \mathcal{F}\{p(\Delta_{xi}, \Delta_{yi})\} \\ \vdots \\ \mathcal{F}\{p(\Delta_{xN}, \Delta_{yN})\} \end{pmatrix} = \begin{pmatrix} P_1(\omega_x, \omega_y) \\ \vdots \\ P_i(\omega_x, \omega_y) \\ \vdots \\ P_N(\omega_x, \omega_y) \end{pmatrix} := \mathbf{P} \tag{7}$$

Thus, the i'th component $P_i(\omega_x, \omega_y)$ of $E\{\mathbf{D}\}$ is the 2-D Fourier transform $\mathcal{F}\{p(\Delta_{xi}, \Delta_{yi})\}$ of the continuous 2-D pdf of the displacement error Δ_{xi}, Δ_{yi}. Using (7), we can write

$$E\{DD^H\} = \begin{pmatrix} 1 & P_1P_2^* & \cdots & P_1P_N^* \\ P_2P_1^* & 1 & \cdots & P_2P_N^* \\ \vdots & \vdots & \ddots & \vdots \\ P_NP_1^* & P_NP_2^* & \cdots & 1 \end{pmatrix} := \mathbf{Q} \tag{8}$$

which holds under the assumption that the displacement error vectors $(\Delta_{xi}, \Delta_{yi})$ and $(\Delta_{xk}, \Delta_{yk})$ are mutually statistically independent for $i \neq k$.

It is well understood how to predict a scalar s from a vector valued signal $\mathbf{c}$, such that the mean square of the prediction error

$$e[x,y] = s[x,y] - \mathbf{h}[x,y] * \mathbf{c}[x,y] \tag{9}$$

is minimized. In (9), the asterisk $*$ denotes 2-D convolution, i.e.,

$$\mathbf{h}[x,y] * \mathbf{c}[x,y] = \sum_{u=-\infty}^{\infty}\sum_{v=-\infty}^{\infty} \mathbf{h}[u,v]\cdot\mathbf{c}[x-u, y-v] \tag{10}$$

Note $\mathbf{h}[x,y]$ is a row vector of impulse responses. The power density of the prediction error is

$$\Phi_{ee} = \Phi_{ss} - \Phi_{\mathbf{sc}}\mathbf{H}^H - \mathbf{H}\Phi_{\mathbf{cs}} + \mathbf{H}\Phi_{\mathbf{cc}}\mathbf{H}^H = \Phi_{ss} - 2\Re\{\mathbf{H}\Phi_{\mathbf{cs}}\} + \mathbf{H}\Phi_{\mathbf{cc}}\mathbf{H}^H, \tag{11}$$

where $\Re\{\cdot\}$ denotes taking the real part and $\mathbf{H} = \mathbf{H}(\omega_x, \omega_y) = \mathcal{F}_*\{\mathbf{h}[x,y]\}$ is a row vector of N complex transfer functions. Using Eqs. (5) and (6) and combining them with (7) and (8) , we can recast (11) as

$$\Phi_{ee} = \Phi_{ss} - 2\Re\{\mathbf{H}\mathbf{P}\Phi_{ss}\} + \mathbf{H}(\Phi_{ss}\mathbf{Q} + \Phi_{\mathbf{zz}})\mathbf{H}^H, \tag{12}$$

Recalling that Φ_{ss} is real-valued, we can rewrite Eq. (12) as

$$\Phi_{ee} = \Phi_{ss}\left(1 - 2\Re\{\mathbf{H}\mathbf{P}\} + \mathbf{H}\left(\mathbf{Q} + \frac{\Phi_{\mathbf{zz}}}{\Phi_{ss}}\right)\mathbf{H}^H\right), \tag{13}$$

The optimum Wiener filter $\mathbf{H}^*$ that minimizes (12) is given by

$$\mathbf{H}^* = \Phi_{\mathbf{cs}}^H \Phi_{\mathbf{cc}}^{-1} = \mathbf{P}^H \left(\mathbf{Q} + \frac{\Phi_{\mathbf{zz}}}{\Phi_{ss}} \right)^{-1} \tag{14}$$

which yields when incorporated into (13)

$$\Phi_{ee} = \Phi_{ss} \left(1 - \mathbf{P}^H \left(\mathbf{Q} + \frac{\Phi_{\mathbf{zz}}}{\Phi_{ss}} \right)^{-1} \mathbf{P} \right) \tag{15}$$

the best prediction performance achievable.

We can use Eqs. (13) and (15) to evaluate some interesting cases numerically. Like in [7, 8, 9], we assume that the video input signal s has an isotropic power spectrum

$$\Phi_{ss}(\omega_x, \omega_y) = \frac{2\pi}{\omega_0^2} \left(1 + \frac{\omega_x + \omega_y}{\omega_0^2} \right)^{-\frac{3}{2}}. \tag{16}$$

The noise power spectrum is assumed to be flat

$$\Phi_{\mathbf{zz}}(\omega_x, \omega_y) = \frac{\sigma_{\mathbf{z}}^2}{4\pi} \mathbf{E}_{(N \times N)}, \tag{17}$$

with $\mathbf{E}_{(N \times N)}$ being the identity matrix of dimension $(N \times N)$. The isotropic pdfs of the displacement errors is assumed to be equal for all c_i, and is given by

$$p(\Delta_x, \Delta_y) = \frac{1}{2\pi\sigma_\Delta^2} \exp \left\{ -\frac{\Delta_x^2 + \Delta_y^2}{2\sigma_\Delta^2} \right\}, \tag{18}$$

with the corresponding power spectral density

$$P(\omega_x, \omega_y) = \exp\{-2\sigma_\Delta^2(\omega_x^2 + \omega_y^2)\}. \tag{19}$$

In contrast to [9] where variations of the motion compensation accuracy, i.e., various values of σ_Δ are analyzed, we restrict the motion compensation accuracy such that spatial displacements are only multiple of the sampling grid X and Y. Assuming equal vertical and horizontal spacing ($X = Y$), the minimum displacement error variance is given as $\sigma_\Delta^2 = X^2/12$.

We define

$$G = 10 \log_{10} \left\{ \frac{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \Phi_{ss}(\omega_x, \omega_y) d\omega_x d\omega_y}{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \Phi_{ee}(\omega_x, \omega_y) d\omega_x d\omega_y} \right\} \tag{20}$$

in order to measure prediction gain. Fig. 1 shows the prediction gain G as function of number of hypotheses N when using the Wiener filter $\mathbf{H}^*$ according to (14). The curves are computed evaluating (15) for various values of $\sigma_{\mathbf{z}}^2 = 0.1, 0.02, 0.01, 0.001, 0.0001$, numerically. The relative gains when increasing the number of hypotheses depend on the noise level $\sigma_{\mathbf{z}}^2$. At the extreme values for $\sigma_{\mathbf{z}}^2$ in our plot, the gain between the cases number of hypotheses $N = 1$ and $N = 2$ is 0.9 dB for $\sigma_{\mathbf{z}}^2 = 0.1$ and 2.1 dB for $\sigma_{\mathbf{z}}^2 = 0.0001$ while the gains between number of hypotheses $N = 1$ and $N = 8$ are 3.4 and 7.2 dB for these noise levels respectively.

The case when simply averaging the N hypotheses, i.e., $\mathcal{F}_*^{-1}\{\mathbf{H}\} = \mathbf{h} = (\frac{1}{N}, \ldots, \frac{1}{N})$ is depicted in Fig. 2. The dashed lines correspond to averaging the hypotheses while the solid lines correspond to optimum Wiener filtering according to (14). Here, completely different tendencies can be observed. First of all, the prediction gains for $N = 1$ hypotheses are much smaller in case of optimal filtering especially for high noise levels $\sigma_{\mathbf{z}}^2$. But, the relative gains when increasing the number of hypotheses from $N = 1$ to $N = 2$ for the averaging case are increasing for increasing noise levels, i.e., we gain 3 dB when operating at noise level $\sigma_{\mathbf{z}}^2 = 0.1$ but only 2.6 dB at noise level $\sigma_{\mathbf{z}}^2 = 0.0001$. The curves for the averaging and optimal filtering cases merge, when the respective prediction error power spectra are flat.

Relating our results to rate distortion theory, we can predict the bit-rate savings ΔR achievable for memoryless encoding of the prediction error using (20) or calculate the rate distortion bound by

$$\Delta R = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log_2 \left(\frac{\Phi_{ss}(\omega_x, \omega_y)}{\Phi_{ee}(\omega_x, \omega_y)} \right) d\omega_x d\omega_y. \tag{21}$$

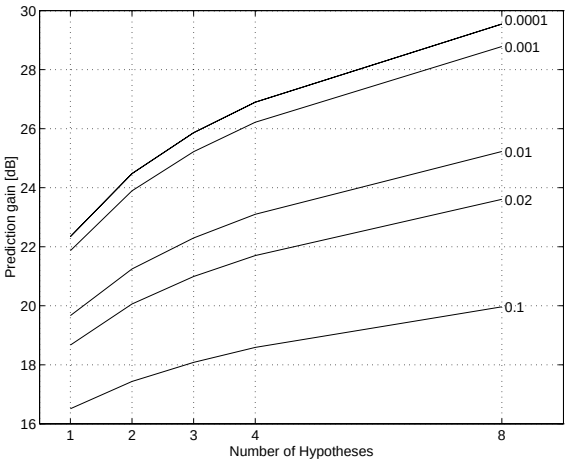

Figure 1: Prediction gain in [dB] versus number of hypotheses, i.e., input vectors N, when using the Wiener filter $\mathbf{H}^*$. The numbers 0.1, 0.02, 0.01, 0.001, 0.0001 indicate various values of $\sigma^2_{\mathbf{z}}$.

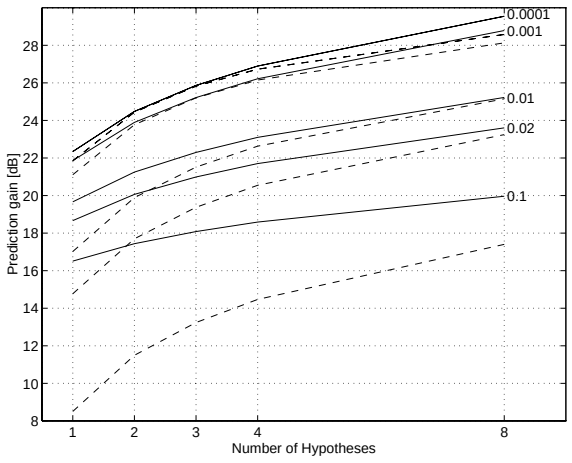

Figure 2: Comparison of prediction gain in [dB] of the Wiener filter $\mathbf{H}^*$ (solid) against the case of averaging the hypotheses $\mathcal{F}_*^{-1}\{\mathbf{H}\} = \mathbf{h} = (\frac{1}{N}, \ldots, \frac{1}{N})$(dashed).

3 Motion-Compensated Linear Vector Prediction

The gains predicted in the previous chapter motivate the design of motion-compensated linear predictors. However, since the information required for motion compensation has to be transmitted as side information to the decoder, we cannot operate on scalars. Hence, we propose to use motion-compensated linear vector predictors instead.

Consider an image of A_h lines and A_w pixels per line in a video sequence at time instant k given by $\mathbf{I}_k$. Let us denote a partitioning of $\mathbf{I}_k$ into blocks of height a_h and width a_w with spacing $a_h \times a_w$ as follows

$$\mathcal{S}_k(a_h \times a_w) = \{\mathbf{S}_{1,1,k}, \ldots, \mathbf{S}_{j,i,k}, \ldots, \mathbf{S}_{b_h,b_w,k}\} \tag{22}$$

with $b_h = A_h/a_h$ and $b_w = A_w/a_w$. Hence, the block $\mathbf{S}_{j,i,k}$ is extracted at vertical position $j \cdot a_h$ and horizontal position $i \cdot a_w$. Note that the image partitioning $\mathcal{S}_k$ is disjunct. Assume a video source code applied to an image $\mathbf{I}_k$ producing $\hat{\mathbf{I}}_k$ as the mapping of the partition $\mathcal{S}_k(a_h \times a_w)$ into its reconstruction $\mathcal{R}_k(a_h \times a_w)$. More precisely, the source code investigated in this work operates independently on each block $\mathbf{S}_{j,i,k}$ and maps it into $\mathbf{R}_{j,i,k}$.

Our block source code employs a linear vector predictor that has motion-compensated support from reconstructed images $\hat{\mathbf{I}}_l$. Let us write the partitioning of image $\hat{\mathbf{I}}_l$ into overlapping blocks with spacing 1×1 as

$$\mathcal{R}_l(1 \times 1) = \{\mathbf{R}_{1,1,l}, \ldots, \mathbf{R}_{j,i,l}, \ldots, \mathbf{R}_{A_h-a_h,A_w-a_w,l}\}. \tag{23}$$

The motion estimation is denoted by selecting a block out of the set of partitioned frames $\{\mathcal{R}_{l1}, \ldots, \mathcal{R}_{li}, \ldots, \mathcal{R}_{lM}\}$ that are available to both encoder and decoder using a 3-D index $\mathbf{v}_n = (x_n, y_n, m_n)$, $1 \leq n \leq N$. The result of this selection process is collected in the set

$$\mathcal{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_n, \ldots, \mathbf{C}_N\}. \tag{24}$$

With these definitions, we define our linear vector predictor as

$$\tilde{\mathbf{S}}_{j,i} = \sum_{n=1}^{N} h_n \cdot \mathbf{C}_n. \tag{25}$$

The $\tilde{\mathbf{S}}_{j,i}$ and $\mathbf{C}_n$ are matrices of dimension $a_h \times a_w$ while h_n is a scalar weighting value.

Note that the original definition of the linear vector predictor permits much more degrees of freedom in that the scalar weights h_n in our definition correspond to matrices in the general vector predictor definition, e.g., see [5]. We can also construct this predictor by writing $\tilde{\mathbf{S}}_{j,i}$ and $\mathbf{C}_n$ as column vectors of size $a_h a_w \times 1$. Then each scalar h_n would be a matrix $\mathbf{H}_n$ of size $a_h a_w \times a_h a_w$. Since typical numbers for a_h and a_w are 16, the matrix $\mathbf{H}_n$ would be of dimension $256 \times 256 = 65536$. Estimation of these high dimensional predictor matrices $\mathbf{H}_n$, for example by using the Wiener equations, would require extremely high complexity. Hence, we restrict the vector predictor to the definition given by (25). But, we will return to this issue in the next section.

4 Optimal Hypothesis Selection Algorithm

Before we move on let us get some insights about the properties of our linear vector prediction model. In order to achieve optimal prediction performance, the filter $\mathbf{h}$ should be a Wiener filter. But then, its coefficients have to be updated simultaneously at encoder and decoder or, if that is not possible, transmitted as side information occupying bit-rate. In addition to that, one might consider to use the large vector prediction model with the coefficient matrix $\mathbf{H}_n$ of size $a_h a_w \times a_h a_w$ for input blocks $\mathbf{C}_i$ of size $a_h \times a_w$ in order to achieve decorrelation between the input vectors. In this case, transmission of the updated predictor coefficients is in most cases not useful because of the high bit-rate associated with that. Hence, a rule for simultaneous up-date of the vector predictor coefficients is needed, like 3-D Kalman filtering [11]. In contrast to that, given fixed prediction vectors $\mathbf{h}$, our approach to the problem is to find a set $\mathcal{C}$ of optimum hypotheses in order to obtain the prediction gain.

For that, each input vector or hypotheses $\mathbf{C}_\nu$ is selected using a 3-D index $\mathbf{v}_\nu = (x_\nu, y_\nu, m_\nu)$, which has to be transmitted as side information to the decoder. This address is relative to the position of the predicted block $\tilde{\mathbf{S}}_{i,j,k}$. We define the search space for the selection of the blocks $\mathbf{C}_\nu$ to be the set of all allowed $\mathbf{v}_\nu$. Thus, the cost for transmitting the 3-D indexes is incurred instead of the bit-rate required to update the prediction filter. In fact, it will turn out later in this section that a compromise between these two approaches is optimum in the rate distortion sense, which our results will indicate.

Allowing a search space of size $[-a, a] \times [-a, a]$ within each previous frame and m previous frames to be used, a full search algorithm to find the optimum set $\mathcal{C}$ of n input vectors implies a complexity of

$$C_f = \left[m(2a+1)^2\right]^n \tag{26}$$

block comparisons. For practical parameters ($a = 15$, $m = 10$, $n = 4$), the complexity of $C_f = 8.5 \cdot 10^{15}$ block comparisons is computationally too demanding.

An iterative algorithm, which is patterned after the Iterated Conditional Modes (ICM) of Besag [12], avoids searching the complete space by successively improving n optimal conditional solutions. Convergence to a local optimum is guaranteed, because the algorithm prohibits an increase of the error measure. A relative decrease of the rate-distortion measure of less than 0.5% indicates practical convergence. Our iterative version of ICM is called Optimal Hypothesis Selection Algorithm (OHSA) and is given in Fig. 3.

The algorithm finds a locally optimal set of n input vectors or hypotheses by minimizing the cost function we defined in step 0. In step 0, we also specified the conditional search space to refine input vector $\mathbf{C}_\mu^{(i)}$, to be the cube around its position in the 3-D collection of the partitioned frames $\{\mathcal{R}_{l1}, \ldots, \mathcal{R}_{li}, \ldots, \mathcal{R}_{lM}\}$. We initialize the OHSA with n hypotheses by applying the rule of *Splitting One Hypothesis*. The computational demand of finding a single hypothesis in the set of $[-a, a] \times [-a, a] \times [1, m]$ is rather moderate. Therefore, we split this optimal 1-hypothesis into n n-hypotheses with identical positions in search space and associate the corresponding predictor coefficients h_ν.

For each n-hypothesis in each iteration, OHSA performs a full search within a conditional search space in which an optimal conditional n-hypothesis has to be found. The size of the conditional search space $[-b, b] \times [-b, b] \times [-b, b]$ affects the quality of the local optimum and the complexity of the algorithm, which is

$$C_i = m(2a+1)^2 + In(2b+1)^3 \tag{28}$$

search positions for I iterations. For practical parameters, $b = 4$, and $I = 3$ iterations, the complexity is reduced by factor $4.6 \cdot 10^{11}$ to $C_i = 1.8 \cdot 10^4$ search positions.

Step 0: Given a motion-compensated vector predictor with n input vectors, we define the cost function

$$D(\mathbf{C}_1, \ldots, \mathbf{C}_\mu, \ldots, \mathbf{C}_n) = \left\| \mathbf{S} - \sum_{\substack{\nu=1 \\ \nu \neq \mu}}^{n} \mathbf{C}_\nu h_\nu - \mathbf{C}_\mu h_\mu \right\|_2^2 \tag{27}$$

subject to minimization for each original block $\mathbf{S}$. Define the size of the conditional search space as $[-b, b] \times [-b, b] \times [-b, b]$. Initialize the algorithm with n hypotheses $(\mathbf{C}_1^{(0)}, \ldots, \mathbf{C}_n^{(0)})$ and set $i := 0$.

Step 1: Set $i := i + 1$ and $\mu = 0$.

Step 2: Set $\mu := \mu + 1$

Step 3: Find $\mathbf{C}_\mu^{(i+1)}$ by minimizing the cost function (27) in the conditional search space

$$\min_{\mathbf{C}_\mu^{(i+1)}} D(\mathbf{C}_1^{(i+1)}, \ldots, \mathbf{C}_{\mu-1}^{(i+1)}, \mathbf{C}_\mu^{(i+1)}, \mathbf{C}_{\mu+1}^{(i)}, \ldots, \mathbf{C}_n^{(i)})$$

Step 4: If $\mu \leq n$, go to step 2, else continue.

Step 5: As long as the target function decreases, go to step 1.

Figure 3: Optimal Hypothesis Selection Algorithm

Figures 4 and 5 reflect the influence of the conditional search space size b and demonstrate the performance of the algorithm. In each input vector refinement step in OHSA, the vectors inside the cube $[-b, b] \times [-b, b] \times [-b, b]$ around the position of the old input vector in the 3-D collection of the partitioned frames is searched. Figures 4 and 5 show PSNR vs. number of hypotheses for the sequences *Foreman* and *Mother-Daughter* when performing motion compensation experiments with QCIF frames, i.e., $A_w = 176$ and $A_h = 144$. The results are averaged over 10s of video when searching blocks of horizontal size $a_w = 16$ and vertical size $a_h = 16$ dependent on conditional search space size b. b is varied over 2, 4, and 8. The search range consists of original blocks, thus the experiment here is to predict an original block by blocks in 10 past frames that are sampled at 7.5 frames/s.

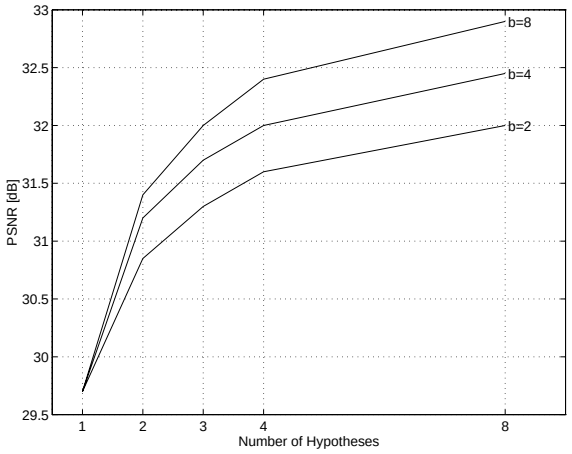

Figure 4: PSNR vs. number of hypotheses for the sequence *Foreman* (QCIF, 7.5 fps, 10s), when searching 16×16 blocks dependent on conditional search space size b. b is varied over 2, 4, and 8.

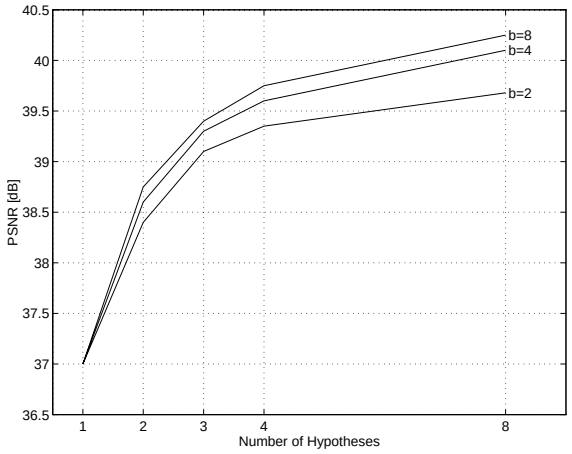

Figure 5: Prediction error and the number of hypotheses for the sequence *Mother-Daughter* (QCIF, 7.5 fps, 10s). Simulation conditions as for the sequence *Foreman* in Fig. 4.

Comparing the results in Figs. 4 and 5 which are measured with real signals with the performance figures

predicted by our our model calculations in section 2, we observe the following. The gains predicted by our theoretical model for 2, 3 and to a certain extend also for 4 hypothesis are encouragingly close to the measured values. For the realistic residual noise level $\sigma_{\mathbf{z}}^2 = 0.01$, our theory predicts 1.6, 2.6 and 3.4 dB gain when increasing the number of hypothesis from 1 to 2, 3 and 4, respectively. For the corresponding cases, we measured for the sequence *Forman* gains of 1.7, 2.3, and 2.7000 dB while the experiments with *Mother-Daughter* showed gains of 1.75, 2.4 and 2.75 dB. Note that the prediction by our theoretical model becomes inaccurate already for 4 hypotheses while the gap between calculated and measured gains for 8 hypotheses is very high. We believe that the OHSA has problems to find that many uncorrelated hypotheses in the restricted search space in our experiment.

5 Entropy-Constrained Motion-Compensated Linear Vector Prediction

So far, we were only concerned with prediction gains and neglected the bit-rate associated with transmitting the information to perform motion compensation in our linear vector predictor. Also, we expect the number of hypotheses effective in the rate distortion sense to vary from block to block. The OHSA neither determines the optimal number of hypotheses in a multi-hypothesis, nor does it take the bit-rate associated with the hypothesis selection into account. Hence, we incorporate an entropy constraint into (27)

$$J(\mathbf{C}_1, \ldots, \mathbf{C}_\mu, \ldots, \mathbf{C}_n) = \left\| \mathbf{S} - \sum_{\substack{\nu=1 \\ \nu \neq \mu}}^{n} \mathbf{C}_\nu h_\nu - \mathbf{C}_\mu h_\mu \right\|_2^2 + \lambda \left(\sum_{\substack{\nu=1 \\ \nu \neq \mu}}^{n} |\gamma(\mathbf{C}_\nu)| + |\gamma(\mathbf{C}_\mu)| \right), \tag{29}$$

where $|\gamma(\mathbf{C}_\nu)|$ denotes the bit-rate associated with the variable length encoding $\gamma(\mathbf{C}_\nu$. We can run the OHSA with (29) instead of (27) and obtain N hypotheses which are locally optimal in the rate distortion sense. The result will, of course, depend on λ, and thus on bit-rate available for signaling the hypotheses selected.

There still remains the question regarding the optimum value of n for a particular block. To solve this problem, we again apply rate-constrained techniques. For a given maximal number N, we determine the optimal number of hypotheses for each original block by running the OHSA for all numbers n from 1 to N and picking the one that minimizes the rate-distortion measure

$$\min_{n:1 \leq n \leq N} \left\{ \left\| \mathbf{s} - \mathbf{c}^{(n)} h^{(n)} \right\|_2^2 + \lambda |\gamma(\mathbf{c}^{(n)})| \right\} \tag{30}$$

The entropy code to transmit the motion information in the variable hypotheses case is patterned similar to universal codes, where the entries in the first code book indicates how many hypotheses are transmitted and with that a pointer into the second code book is given. The second code book contains the code words actually associated to the motion information.

Figures 6 and 7 compares five different motion-compensated vector predictors in terms of average PSNR vs. average bit-rate to transmit the information to perform motion compensation at the decoder. We evaluate the rate-distortion performance for the designed predictors ($\lambda = 100$) by predicting the test sequences *Foreman* and *Mother-Daughter* for various values of the Lagrange multiplier values $(25, 50, 100, \ldots, 1600)$. The optimal number of hypotheses in the rate-distortion sense depends significantly on the rate constraint. We compare the adaptive method which determines the optimum number of hypotheses according to (30) with four curves generated by methods with a constant number of hypotheses. The four curves for the method with constant numbers of hypothesis are generated using the OHSA with (29) incorporated instead of (27) and setting $n = 1, 2, 3$ and 4. We observe that the adaptive method outperforms the methods with constant number of hypotheses.

Focusing on the sequence *Foreman*, the overall PSNR gain when comparing the adaptive motion-compensated linear vector predictor to conventional motion compensation, i.e., the case of $n = 1$ constant input vectors, is about 2.3 dB at the cost of 13 kbit/s increased side information. Note that the reference in this case was motion compensation using the last 10 frames. Note that in existing video coding algorithms, motion compensation is performed by using the last decoded frame only. Though not given in the figures, the PSNR for the corresponding

motion compensation experiment using the last frame is 27.7 dB at 7 kbit/s. When comparing that to the results produced by the adaptive motion-compensated vector predictor, we obtain PSNR gains of 4.4 dB at the cost of 16 kbit/s that we have to transmit as side information.

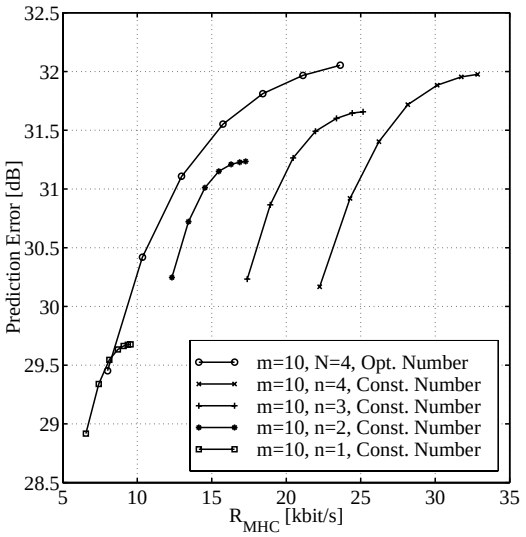

Figure 6: PSNR vs. bit-rate of the for the sequence *Foreman* (QCIF, 7.5 fps, 10s), and 16×16 blocks.

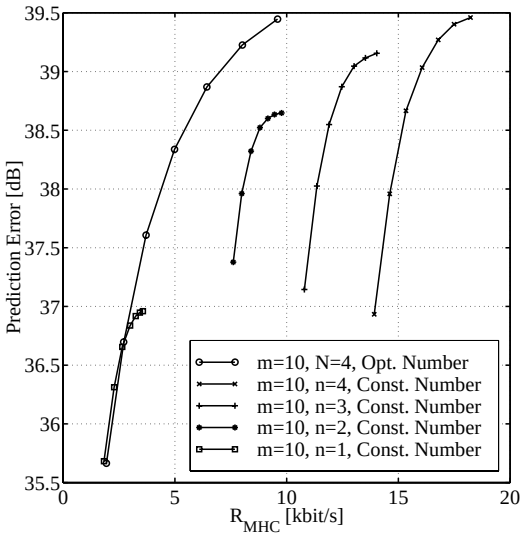

Figure 7: PSNR vs. bit-rate of the for the sequence *Mother-Daughter* (QCIF, 7.5 fps, 10s), and 16×16 blocks.

6 Conclusions

In this paper, we investigate the efficiency of motion-compensated multi-hypothesis prediction for video coding. The theoretical analysis based on second order statistics yields insights to what extend the increase of the number of input vectors can reduce energy of the prediction error.

The extension of the scalar approach to motion-compensated vector prediction demonstrated how closely our model calculations are related to practical experiments. For a realistic residual noise level the theory predicts 1.6, 2.6 and 3.4 dB gain when increasing the number of hypothesis from 1 to 2, 3 and 4, respectively. For the corresponding cases, we measured, for example, for the sequence *Forman* gains of 1.7, 2.3, and 2.7 dB which were achieved by motion-compensated linear vector prediction with 16×16 blocks.

Due to the time and space variant statistics in real video signals, the vector predictor coefficients are not updated, i.e., we keep the coefficient sets fixed and conduct a conditional search to find the optimum input vectors. This search is further accelerated by using equal, scalar weights. Thus, instead of decorrelating the input signals by adapting the prediction filter, we search the optimum set of input vectors for a given prediction filter.

We incorporate an entropy constraint into the search algorithm for the vector predictor input signals. An adaptive algorithm for optimally selecting the size of the linear vector predictor is given. When comparing the case of several input vectors to just one input vector both having prediction support of 10 frames, a PSNR gain of 2.3 dB at the cost of 13 kbit/s increased side information is obtained. Note that in existing video coding algorithms, motion compensation is performed by using only the last decoded frame. For this case, the designed motion-compensated vector predictors show PSNR gains up to 4.4 dB at the cost of increased bit-rate of 16 kbit/s. These results are very encouraging regarding the gains achievable when incorporating entropy-constrained motion-compensated linear vector prediction into a full video algorithm including prediction error coding, which is subject to future work.

References

[1] S.-W. Wu and A. Gersho, "Joint Estimation of Forward and Backward Motion Vectors for Interpolative Prediction of Video", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 684–687, Sept. 1994.

[2] M. T. Orchard and G. J. Sullivan, "Overlapped Block Motion Compensation: An Estimation-Theoretic Approach", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 693–699, Sept. 1994.

[3] V. Cuperman and A. Gersho, "Adaptive Differential Vector Coding of Speech", in *GLOBECOM'82*, Dec. 1982, pp. 1092–1096.

[4] T. R. Fischer and D. J. Tinnen, "Quantized Control with Differential Pulse Code Modulation", in *21st Conference on Decision and Control*, Dec. 1982, pp. 1222–1227.

[5] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston,Dordrecht,London, 1992.

[6] V. Cuperman and A. Gersho, "Vector Predictive Coding of Speech at 16 kbits/s", *IEEE Transactions on Communications*, vol. 33, no. 7, pp. 685–696, July 1995.

[7] B. Girod, "The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video Sequences", *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, Aug. 1987.

[8] B. Girod, "Motion-Compensating Prediction with Fractional-Pel Accuracy", *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[9] B. Girod, "Efficiency Analysis of Multi-Hypothesis Motion-Compensated Prediction for Video Coding", *IEEE Transactions on Image Processing*, 1997, Submitted for publication.

[10] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-Constrained Vector Quantization", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 1, pp. 31–42, Jan. 1989.

[11] J. Kim and J. W. Woods, "Spatio-Temporal Adaptive 3-D Kalman Filter for Video", *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 414–424, Mar. 1997.

[12] J. Besag, "On the Statistical Analysis of Dirty Pictures", *J. Roy. Statist. Soc. B*, vol. 48, no. 3, pp. 259–302, 1986.