SPATIO-TEMPORAL PREDICTION IN VIDEO CODING BY BEST APPROXIMATION

Jürgen Seiler, Haricharan Lakshman, and André Kaup

Chair of Multimedia Communications and Signal Processing,
University of Erlangen-Nuremberg,
Cauerstr. 7, 91058 Erlangen, Germany
{seiler, kaup}@LNT.de, haricharan.lakshman@hhi.fraunhofer.de

ABSTRACT

Within the scope of this contribution we propose a novel efficient spatio-temporal prediction algorithm for video coding. The algorithm operates in two stages. First, motion compensation is performed on the block to be predicted in order to exploit temporal correlations. Afterwards, in order to exploit spatial correlations, this preliminary estimate is spatially refined by forming a joint model of the motion compensated block and spatially adjacent already decoded blocks. Compared to an earlier refinement algorithm, the novel one only needs very few iterations, leading to a speedup of factor 17. The implementation of this new algorithm into the H.264/AVC leads to a maximum reduction in data rate of up to nearly 13% for the considered sequences.

Index Terms— Video coding, Prediction, Extrapolation

1. INTRODUCTION

In the past few years, the amount of video data transmitted over digital channels has steadily increased. For this it is necessary that the video sequences are compressed by an encoder in order to reduce the data rate. Fortunately, video sequences can be strongly compressed. Most modern hybrid video codecs as e. g. the H.264/AVC [1] use two different strategies for compressing the video sequence: prediction of the video signal to be coded and entropy coding of the prediction residual and the side information. Within the scope of this contribution we will focus on the first one. In this step, an estimate of the signal parts to be coded is generated from already transmitted areas, i. e. previous frames or already processed regions from the actual frame. Since only already transmitted areas are used for generating the estimate, the decoder can predict the signal in the same way as the encoder. Thus, instead of transmitting the quantized and entropy coded original video signal, only the quantized and entropy coded prediction error has to be transmitted. Therefore, besides the entropy coding, the abilities of a video codec directly depend on how efficiently the video signal to be coded can be predicted.

In most modern video codecs the prediction of the signal part to be coded is obtained by exploiting either temporal or spatial correlations. Thereby, spatial prediction is obtained by skillfully continuing the signal from already transmitted regions into the region being processed. On the other hand, the temporal prediction is performed by applying motion compensation (MC) on the region being coded, as described in [2]. For this, a region in a previous frame is sought that fits the area to be coded best. The displacement of this region then is transmitted to the decoder as side information and the decoder then can form the prediction signal by taking the corresponding region from the already decoded frames. Although modern encoders can adaptively switch between spatial and temporal prediction in order to form the best predictor, a combined usage of temporal and spatial correlations only is applied rarely. Only few existent prediction algorithms exploit both correlations at the same time. As examples for this group of algorithms the 'Inter Frame Coding with Template Matching Spatio-Temporal Prediction' by [3] or the 'Pixelwise Adaptive Spatio-Temporal Prediction' by [4] can be mentioned.

In [5] we proposed a new spatio-temporal prediction algorithm, the spatial refinement of motion compensation by Frequency Selective Approximation (FSA). This algorithm is able to reduce data rate significantly compared to pure temporal prediction. Unfortunately, this algorithm needs many iterations to form an adequate predictor and thus is computationally very expensive. We now want to propose a new algorithm for spatial refinement, the Relaxed Best Approximation (RBA), which is able to generate the model nearly as effectively as the original algorithm but with only very few iterations needed. The algorithm is based on the 'Best Approximation' proposed by [6].

In the next sections we will outline the idea of spatial refinement with FSA and especially with the new algorithm in detail. We will also prove its abilities to improve the prediction quality in simulations with the H.264/AVC encoder and will show the reduction in computational complexity compared to the algorithm presented in [5].

2. PROBLEM FORMULATION AND BACKGROUND

We consider a block based video coder, operating in line scan order. Let the block actually being processed be denoted by area $\mathcal{B}$. This block is joined by 4 blocks that have already been transmitted and are known to the decoder as well. These

blocks are subsumed in area $\mathcal{R}$. We now regard the so called projection area $\mathcal{P}$, shown in Fig. 1, of 3×3 blocks centered by the block $\mathcal{B}$. Besides $\mathcal{R}$ and $\mathcal{B}$, this square area contains four blocks that have not been coded yet. The novel idea of the spatial refinement proposed in [5] is that first a preliminary temporal extrapolation is formed by motion compensation for the block $\mathcal{B}$. By transmitting the motion vector as side information, the decoder can perform the motion compensation in the same way. In a second step, a model is generated for the union $\mathcal{A} = \mathcal{R} \cup \mathcal{B}$, called approximation area. Finally the samples corresponding to $\mathcal{B}$ are taken from the model and are used as predictor. As the model incorporates information from the temporally extrapolated block $\mathcal{B}$ as well as from the spatially adjacent blocks $\mathcal{R}$ this will form a better predictor than the purely temporal one.

Let the intensities of the samples in area $\mathcal{P}$ be denoted by $f[m,n]$ and the model, representing the refined signal, be denoted by $g[m,n]$. (m,n) represent the spatial coordinates and area $\mathcal{P}$ is of size $M \times N$ samples. The parametric model

$$g[m,n] = \sum_{k \in \mathcal{K}} c_k \varphi_k[m,n] \tag{1}$$

emanates from a weighted superposition of the two-dimensional basis functions $\varphi_k[m,n]$ with appropriate weights c_k. The set $\mathcal{K}$ covers all basis functions used for modeling.

For the purpose of spatial refinement of the motion compensated signal, the samples of $\mathcal{R}$ that are close to $\mathcal{B}$ need to have more influence than the ones far away. This non-uniform influence is incorporated by means of the later used weighting function

$$w[m,n] = \begin{cases} \mu & ,\forall\,(m,n) \in \mathcal{B} \\ \hat{\rho}^{\sqrt{\left(m-\frac{M-1}{2}\right)^2+\left(n-\frac{N-1}{2}\right)^2}} & ,\forall\,(m,n) \in \mathcal{R} \\ 0 & ,\text{ else} \end{cases} \tag{2}$$

Hence the block $\mathcal{B}$ gets the constant weight μ and the samples in $\mathcal{R}$ get an exponentially decreasing weight with an increasing distance, controlled by the decay factor $\hat{\rho}$.

The model $g[m,n]$ now should be generated in such a way as to minimize the weighted approximation error energy

$$E = \sum_{(m,n) \in \mathcal{P}} w[m,n]\,(f[m,n] - g[m,n])^2. \tag{3}$$

According to (3), the model generation for spatio-temporal prediction can be viewed as an error minimization task. It is important to notice that the traditional approach of minimizing the error by taking partial derivatives with respect to the unknown coefficients c_k and equating them to zero leads to an underdetermined system of equations because the number of known samples is less than the total number of points considered. For such problems, as shown in [7], sparsity based solutions are advocated because they are capable of capturing important characteristics of a signal. However, direct solutions using l_0 quasi-norm as a sparsity measure are NP-Hard according to [8].

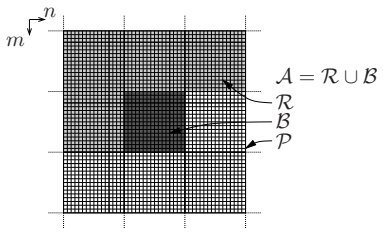

Fig. 1. *Projection area $\mathcal{P}$ containing the approximation area $\mathcal{A}$ consisting of area $\mathcal{R}$ subsuming the reconstructed blocks and the block $\mathcal{B}$ to be predicted*

The Frequency Selective Approximation (FSA) algorithm from [6] employed in [5] is an iterative error minimization procedure which produces a sparse solution. It belongs to the class of Greedy Approximation techniques in which the signal is approximated in terms of one additional basis function per iteration. This involves the selection of a basis function $\varphi_u[m,n]$ and the computation of the optimal expansion coefficient c_u corresponding to the selected basis function. In each iteration, the basis function is chosen in such a way that the reduction of the weighted residual energy is maximized. After generating the parametric model $g[m,n]$, the area of interest is cut out and used as predictor for the block being coded. For a detailed discussion of FSA, please refer to [5, 6].

The block diagram in Fig. 2 shows the position of the spatial refinement step in a generalized hybrid video encoder. The deblocking filter, as e. g. used in the H.264/AVC [1], is an optional feature that can be used in addition to the spatial refinement without interfering with it. As shown in [5], the gain obtainable by deblocking adds to the gain obtainable by spatial refinement. This is since deblocking only improves the reference frames and therewith motion compensation but does not incorporate spatial correlations for prediction.

3. BEST APPROXIMATION

This section introduces the Best Approximation (BA), originally proposed in [6] and discusses its advantages. The basic idea of BA is close to FSA as in every iteration step, one basis function is added to the model generated so far. The basis function selected is, as in FSA, the one that maximizes the approximation error energy decrement. But unlike FSA where the residuum is approximated just by the selected basis function in each iteration step, BA modifies the expansion coefficients of all the already selected basis functions in order to produce the best possible approximation using the selected set. The expansion coefficients for the selected basis functions are calculated by solving a projection problem in least squares sense. According to approximation theory, such an algorithm can be categorized as an Orthogonal Greedy Approximation whose anatomy is similar to Greedy Approximation but has a faster convergence [9].

Let $g^{(\nu)}[m,n]$ represent the parametric model and $\mathcal{K}^{(\nu)}$ the set of selected basis functions in the ν-th iteration step. The

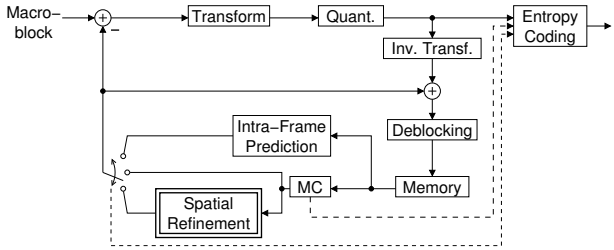

Fig. 2. *Block diagram of a hybrid video encoder with spatial refinement*

new model, in which the coefficients of all the selected basis functions are updated, can be written as

$$g^{(\nu+1)}[m,n] = g^{(\nu)}[m,n] + \sum_{u \in \mathcal{K}^{(\nu+1)}} \Delta c_u \varphi_u[m,n]. \quad (4)$$

The weights Δc_u can be computed by setting the partial derivatives of the weighted error energy with respect to all Δc_u to zero. This yields a system of linear equations of size $\left|\mathcal{K}^{(\nu+1)}\right|$ for the coefficients Δc_u

$$\sum_{(m,n)\in\mathcal{P}} w[m,n]\left(f[m,n] - g^{(\nu)}[m,n]\right)\varphi_k[m,n] =$$
$$\sum_{u\in\mathcal{K}^{(\nu+1)}} \Delta c_u \sum_{(m,n)\in\mathcal{P}} w[m,n]\,\varphi_k[m,n]\,\varphi_u[m,n], \quad \forall k \in \mathcal{K}^{(\nu+1)} \quad (5)$$

Solving this linear system gives all the coefficients Δc_u which are then used to update the parametric model

$$c_u^{(\nu+1)} = c_u^{(\nu)} + \Delta c_u, \ \forall \ u \in \mathcal{K}^{(\nu+1)}. \quad (6)$$

These steps of selecting one basis function and updating the expansion coefficients of all selected basis functions are repeated until a predefined maximum number of iterations is reached. By updating the expansion coefficients of all the selected basis functions in one iteration step, the algorithm requires a smaller number of iterations compared to FSA.

Relaxation Scheme

The spatial refinement of temporally predicted data depends on the ability of the parametric modeling to combine the important characteristics of the regions $\mathcal{R}$ and $\mathcal{B}$. Although BA yields the best possible approximation of union $\mathcal{R}\cup\mathcal{B}$ in each iteration step, it might not result in a better spatial refinement. Additionally, in each iteration of BA, a system of linear equations needs to be solved, which adds computational complexity. In order to tackle these issues, we introduce a relaxation scheme which does not only improve the refinement performance of BA but also provides a reduction in complexity. The new scheme performs a Relaxed Best Approximation (RBA) by selecting all the basis functions that provide at least a specified fraction of the maximum reduction in error energy into the chosen set for a particular iteration. Therefore, the relaxation parameter τ between 0 and 1 is introduced that controls which basis functions to be added to the model in a certain iteration step. Thus, the steps of RBA are defined as follows:

1. Select all basis functions $\varphi_u[m,n]$ that satisfy
$$\Delta E_{\varphi_u}^{(\nu)} \geq \tau \cdot \max_{\varphi_k} \Delta E_{\varphi_k}^{(\nu)} \quad (7)$$
with $\Delta E_{\varphi_u}^{(\nu)}$ being the reduction in residual energy by selecting basis function φ_u
2. Update model $g^{(\nu+1)}[m,n]$ according to (5) and (4)
3. Compute new residual and iterate

So, by using RBA, in every iteration step, several basis functions can be added to the model. This results in a reduction of the number of iterations needed to set up the model for forming the predictor and with that the overall complexity will become a lot smaller than with FSA or BA.

4. SIMULATION SETUP AND RESULTS

In order to evaluate the abilities of the proposed algorithm, we implemented this new prediction mode into the H.264/AVC reference software JM 10.2, Baseline Profile, Level 2.0 . For motion compensation, quarter pixel accuracy is applied at a search range of 16 pixels and one reference frame. For comparing the refined prediction with the original pure motion compensation and spatial refinement by FSA, the rate control is switched off and 10 fixed QPs from 16 to 43 are used.

In order to evaluate the prediction quality, the sequences "Crew", "Foreman" and "Vimto" in CIF are encoded at 30 frames per second with three different settings: pure motion compensation for prediction, spatial refinement by FSA [5] and spatial refinement by the proposed RBA. As the spatial refinement might not increase the prediction quality for every macroblock, the encoder has to compare the refined and the unrefined prediction signal with the original block and has to signal the decoder if the refinement step is applied. To account for this, one bit per macroblock is added to the data for the two refinement algorithms as a worst case assessment for the emerging additional side information.

The weighting function $w[m,n]$ used for spatial refinement is the same for FSA and RBA with the decay factor $\hat{\rho}$ chosen to 0.8 and the weighting of the motion compensated block chosen to $\mu = 0.5$. According to [5], for both the algorithms, the set of basis functions used for model generation are the functions of the two-dimensional Fourier transform, since this set is especially suited for natural images. According to the previous experiments, FSA uses 200 iterations, whereas the proposed RBA uses only 4 iterations to form the model. For RBA the factor τ is set to 0.5 and 20 basis functions are maximally added to the model in one iteration step. Fortunately, the above mentioned parameters all are not very critical and can be varied in a relatively wide range without heavily affecting the prediction performance.

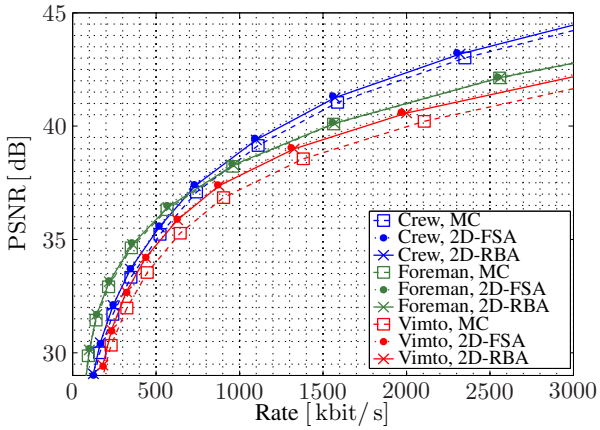

Fig. 3. *RD-curves for the first* 99 *P-frames of the used CIF-sequences at* 30 *frames per second with direct motion compensation (MC) and spatial refinement by FSA [5] and RBA*

Sequence	FSA [5]		RBA	
	Avg. Rate Reduction	Avg. PSNR Gain	Avg. Rate Reduction	Avg. PSNR Gain
"Crew"	7.31%	0.37 dB	6.20%	0.31 dB
"Foreman"	3.20%	0.13 dB	1.42%	0.06 dB
"Vimto"	13.42%	0.66 dB	12.61%	0.62 dB

Table 1. *Achievable average relative rate reduction and average* PSNR *gain according to [10] for spatial refinement by FSA and the proposed RBA*

In Fig. 3 the rate-distortion curves for the first 99 P-frames of the considered sequences are shown. For each sequence the figure contains the curves for the cases that only pure motion compensation is applied for prediction and that spatial refinement is used, either with FSA or the proposed RBA. Obviously, both refinement algorithms lead to a reduction in data rate needed to obtain a certain quality. Unfortunately, for the sequence "Foreman" this gain is small and cannot be seen well in the figure. Hence and to quantify the gain, Tab. 1 lists the average rate reduction and average PSNR gain compared to motion compensation. Both averages are calculated according to [10] and one can see a maximum reduction in data rate of up to 13% and a mean reduction of about 7% for the regarded sequences.

Comparing the improvement introduced by the spatial refinement with FSA and RBA, it becomes apparent, that FSA is slightly better than RBA. But one major drawback of FSA is the large number of iterations needed to generate the model. For this reason Tab. 2 shows the mean calculation time per frame for the spatial refinement for both algorithms. The spatial refinement step was carried out in MATLAB v7.6 on a Intel Core 2 @ 2.4 GHz. The motion compensated block and the spatially adjacent blocks are exported to MATLAB for refinement. Afterwards the refined block is retransfered to JM for the further coding steps. According to Tab. 2, we can see that RBA needs only about $1/17$-th of the processing time of FSA, which could be further improved by solving the system of equations from (5) more efficiently. Considering this large reduction in processing time, the small increment in data rate compared to FSA becomes negligible.

Sequence	Mean time / frame		Time Gain
	FSA [5]	RBA	
"Crew"	217.49 sec	13.03 sec	16.69
"Foreman"	214.93 sec	12.15 sec	17.69
"Vimto"	217.07 sec	12.43 sec	17.46

Table 2. *Mean processing time per frame for spatial refinement*

5. CONCLUSION

Regarding the above presented results, one can easily see that prediction in video coding can significantly gain from using spatial as well as temporal information to form the predictor. Within the scope of this contribution we presented a novel spatial refinement algorithm which produces an average reduction of the data rate of up to 13% at maximum for the regarded sequences. In addition, this new algorithm needs only very few iterations to form the predictor and is about 17 times faster than an earlier proposed algorithm.

Furthermore, our current research is focused on combining the spatial refinement with more sophisticated prediction techniques and on a further complexity reduction.

6. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560–576, July 2003.

[2] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 858–876, June 1995.

[3] K. Sugimoto, M. Kobayashi, Y. Suzuki, S. Kato, and C. S. Boon, "Inter frame coding with template matching spatio-temporal prediction," *Proc. Int. Conf. on Image Processing (ICIP)*, vol. 1, pp. 465–468, Oct. 2004.

[4] M. G. Day and J. A. Robinson, "Residue-free video coding with pixelwise adaptive spatio-temporal prediction," *IET Image Processing*, vol. 2, no. 3, pp. 131–138, 2008.

[5] J. Seiler and A. Kaup, "Spatio-temporal prediction in video coding by spatially refined motion compensation," in *Proc. Int. Conf. on Image Processing (ICIP)*, San Diego, USA, 12.-15. Oct. 2008, pp. 2788–2791.

[6] A. Kaup and T. Aach, "Coding of segmented images using shape-independent basis functions," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 937–947, July 1998.

[7] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[8] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, April 1995.

[9] V. Temlyakov, "Weak greedy algorithms," *Advances in Computational Mathematics*, vol. 12, no. 2-3, pp. 213–227, Feb. 2000.

[10] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," ITU-T VCEG Meeting, Austin, Texas, USA, document VCEG-M33, Tech. Rep., April 2001.