SPARSE LEAST-SQUARES PREDICTION FOR INTRA IMAGE CODING

Luís F. R. Lucas[1,4], *Nuno M. M. Rodrigues*[1,2], *Carla L. Pagliari*[3], *Eduardo A. B. da Silva*[4], *Sérgio M. M. de Faria*[1,2]

[1]Instituto de Telecomunicações; [2]ESTG, Instituto Politécnico de Leiria, Portugal; [3]DEE, Instituto Militar de Engenharia; [4]PEE/COPPE/DEL/Poli, Univ. Federal do Rio de Janeiro, Brazil; *e-mails: luis.lucas,eduardo@smt.ufrj.br, nuno.rodrigues,sergio.faria@co.it.pt, carla@ime.eb.br*

ABSTRACT

This paper presents a new intra prediction method for efficient image coding, based on linear prediction and sparse representation concepts, denominated sparse least-squares prediction (SLSP). The proposed method uses a low order linear approximation model which may be built inside a predefined large causal region. The high flexibility of the SLSP filter context allows the inclusion of more significant image features into the model for better prediction results.

Experiments using an implementation of the proposed method in the state-of-the-art H.265/HEVC algorithm have shown that SLSP is able to improve the coding performance, specially in the presence of complex textures, achieving higher coding gains than other existing intra linear prediction methods.

Index Terms— Intra Prediction, Least-Squares Minimization, Sparse Coding, Image Coding

1. INTRODUCTION

Intra and inter prediction constitute a fundamental part of modern transform-based image and video coding standards, like H.264/AVC [1] and H.265/HEVC [2]. Intra prediction is used to eliminate the correlation between the target block and its neighboring samples, and it is often derived by extrapolating the reconstructed pixels surrounding the target block. In addition to DC and plane modes, H.264/AVC standard may use up to 8 directional prediction modes, while H.265/HEVC considers 33 possible directions.

Directional prediction provides a reasonable approximation for the edges and contours aligned with directional modes. However, it tends to fail in the presence of arbitrarily-oriented edges and more complex textured regions. Least-Squares Prediction (LSP), proposed in [3], is an alternative approach for intra prediction, particularly proposed for the representation of arbitrarily-oriented edges. LSP has been successfully applied in some state-of-the-art encoders, like the transform-based H.264/AVC standard [4] and the pattern-matching-based MMP algorithm [5], improving their coding performance. Inter prediction approaches based on LSP have been also presented for stereo image coding using MMP algorithm [6]. For more complex textured areas, alternative intra prediction methods have been investigated, extrapolating to the block being encoded similarities in the causal reconstructed parts of the image. Methods based on Template-Matching (TM) and Block-Matching (BM) were used for this purpose [7]. Spatial image prediction has been also carried out through the use of sparse approximation algorithms, based on iterative greedy algorithms such as matching pursuit (MP) [8]. In the context of sparse approximation, neighbor-embedding (NE) methods have recently been proposed for image prediction in H.264/AVC encoder, specifically using the local linear embedding (LLE) and the nonnegative matrix factorization (NMF) frameworks [9, 10].

In this paper, an improved linear prediction algorithm, denominated sparse least squares prediction (SLSP) method, is proposed to complement existing directional intra prediction modes, namely in the presence of images with complex textured areas and high frequencies. SLSP uses an adaptive linear model context that may use more relevant information than traditional LSP algorithms by exploiting a larger causal region. A sparsity restriction on model context forces a low model order in order to avoid overfitting issues and better estimate underlying relationships of training area data. Inspired on NE-methods [9], the *k*-nearest neighbors (*k*-NN) method is used to ensure the low model order. SLSP method has been evaluated within the most recent state-of-the-art H.265/HEVC standard.

The paper is organized as follows. Sections 2 and 3 briefly review the least-squares and the sparse approximation approaches for image prediction, respectively. Section 4 describes the proposed SLSP mode. Experimental results are presented and discussed in Section 5, while Section 6 concludes the paper.

2. LEAST-SQUARES PREDICTION

Least-squares prediction has been successfully applied for image compression [3, 4, 5]. The main idea of LSP is to estimate a set of prediction coefficients from the causal reconstructed data, which are then used in the linear prediction model. By using a local training window, LSP implicitly embeds the local texture characteristics into the prediction coefficients. This is the basis of the edge directed prediction method presented for lossless image coding in [3], which is described as follows.

2.1. LSP algorithm for image compression

Let $X(\mathbf{n})$ denote the image pixel to be linearly predicted, where $\mathbf{n}$ is a two-dimensional vector with the spatial coordinates in an image. By using the N nearest spatial causal neighbors, according to a Nth order Markovian model, the predicted pixel is computed as:

$$\hat{X}(\mathbf{n}) = \sum_{i=1}^{N} a_i X(\mathbf{n} - \mathbf{g}(i)), \tag{1}$$

This work was funded by FCT - "Fundação para a Ciência e Tecnologia", Portugal, under the grant SFRH/BD/79553/2011 and project UID/EEA/50008/2013. The third and fourth authors acknowledge the financial support of CNPq and CAPES/Pro-Defesa, Brazil, under research grant 23038.009094/2013-83.

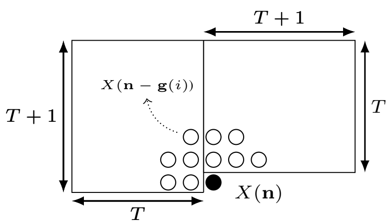

Fig. 1. LSP filter support (pixels represented by white circles) and associated training window.

where $\mathbf{g}(i)$ gives the relative position of each pixel in the filter context (support), and a_i are the filter coefficients. An example of the filter support, as proposed in [5], is illustrated in Figure 1 for the case of order $N = 10$.

In order to avoid the transmission of side information, LSP locally optimizes the prediction coefficients within a causal training window, which is available in both the encoder and the decoder. In [3], a rectangular window that contains $M = 2T(T+1)$ elements is proposed, as show in Figure 1. Consider the training window samples arranged in a column vector $\vec{y} = [X(\mathbf{n} - \mathbf{h}[1]) \ldots X(\mathbf{n} - \mathbf{h}(M))]^T$, with $\mathbf{h}(j)$ representing the relative position of each pixel in the training window, and the matrix $\mathbf{C}$ whose element (j, i) is $X(\mathbf{n} - \mathbf{h}(j) - \mathbf{g}(i))$, representing the ith filter support neighbor associated to the training window pixel $X(\mathbf{n} - \mathbf{h}(j))$. The filter coefficients, $\vec{a} = [a_1 \ldots a_N]^T$, can be determined by least-squares optimization, finding the solution for $\min(\|\vec{y} - \mathbf{C}\vec{a}\|_2^2)$. A well-known closed-form solution for LS problem is given by:

$$\vec{a} = (\mathbf{C^T C})^{-1}(\mathbf{C^T}\vec{y}). \tag{2}$$

2.2. Block prediction using LSP

Despite the first LSP proposals being intended for lossless image coding, efficient block-based implementations for lossy image coding have been also proposed. In [5], an LSP mode based on the previously described algorithm was presented for block-based intra prediction, in MMP algorithm. It proposes to recompute the linear coefficients for each pixel to be predicted, using previously predicted pixels during training procedure when reconstructed ones are unavailable.

An alternative LSP method for block-based intra prediction in the H.264/AVC standard was proposed in [4]. It uses an adaptive training window and filter support, depending on the number of available neighbor blocks. This approach differs from [5] by the fact that LSP training is not performed in a pixel-by-pixel basis. By using a fixed training window for all the block pixels, the same set of common linear coefficients can be used for the whole block pixels, resulting in high computational complexity savings. LSP prediction has been also used for inter prediction between stereo pair views in [6].

3. SPARSE REPRESENTATION METHODS

Sparse representation methods have been increasingly considered in the literature for signal and image processing, including image coding applications [8]. The underlying assumption of these methods is that natural images are composed by few structural primitives or representative features. Sparse prediction tries to approximate the input signal using a linear combination of a small number of these primitives, selected from a large and redundant basis or dictionary.

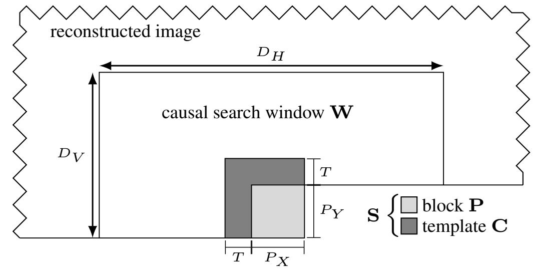

Fig. 2. Approximation support (or template) and search window used in sparse coding and neighbor-embedded methods.

3.1. Sparse prediction algorithm

Consider the N-pixel region $\mathbf{S}$, the union of the N_p-pixel block $\mathbf{P}$ to be predicted, and the N_c-pixel approximation support (or template) $\mathbf{C}$, as illustrated in Figure 2. Note that the pixels from the block $\mathbf{P}$ to be predicted are unknown, and the ones of the template $\mathbf{C}$ are the previously reconstructed pixels. By using an appropriate dictionary, the sparse prediction method estimates the best linear approximation for the known template $\mathbf{C}$ and uses the same model to approximate the corresponding unknown block $\mathbf{P}$.

Let vector $\vec{b}$ be composed by the N pixel values of region $\mathbf{S}$, stacked in a column (assuming the zero value for unknown values of block $\mathbf{P}$). Also, let an $N \times M$ matrix $\mathbf{A}$, represent the basis dictionary. This basis is built by stacking all the texture patches with the shape of region $\mathbf{S}$, which exist in the causal search window $\mathbf{W}$, as shown in Figure 2. An overcomplete dictionary is used, that is, the number of elements M is greater than the size N of each patch.

The dictionary matrix $\mathbf{A}$ (and vector $\vec{b}$), can be separated into two vertically concatenated sub-matrices $\mathbf{A_c}$ and $\mathbf{A_P}$ (and two vectors $\vec{b}_c$ and $\vec{b}_p$), corresponding to the pixels in the spatial location of template $\mathbf{C}$ and predicting block $\mathbf{P}$, respectively. Sparse representation algorithms aim at approximating the template $\mathbf{C}$ (vector $\vec{b}_c$), by solving the following optimization problem:

$$\min_{\vec{x}} \|\vec{b}_c - \mathbf{A}_c\vec{x}\|_2^2 \quad \text{subject to} \quad \|\vec{x}\|_0 \leq \rho \tag{3}$$

where $\|\vec{x}\|_0$ denotes the L_0 norm of $\vec{x}$, *i.e.* the number of non-zero components in $\vec{x}$ and ρ is a parameter that controls the sparsity level.

Since searching for the sparsest solution for this problem is NP-hard, matching pursuit (MP) and orthogonal matching pursuit (OMP) algorithms have been used as heuristic methods to find approximate solutions with tractable computational complexity [11]. When optimal coefficients are found, sparse prediction generates the predicted signal through $\vec{b}_p = \mathbf{A}_p\vec{x}_{opt}$, where $\vec{x}_{opt}$ is the sparsest found solution of (3).

3.2. Neighbor-embedding methods

In the context of sparse prediction, data dimensionality reduction methods have been recently investigated for image prediction. In [9, 10], LLE and NMF-based methods were proposed and evaluated in H.264/AVC intra prediction framework. The principle of these methods is similar to the one previously described for sparse prediction methods. The linear representation for the template $\mathbf{C}$ (refer to Figure 2) is estimated based on the k-NN template patches defined in a matrix $\mathbf{A}_c$, built from the causal search window $\mathbf{W}$. Then, the coefficients are used to predict the whole block $\mathbf{P}$, by linearly combining the co-located pixels in the k-NN block patches. In NE-methods, k-NN algorithm is used to impose the sparsity constraint

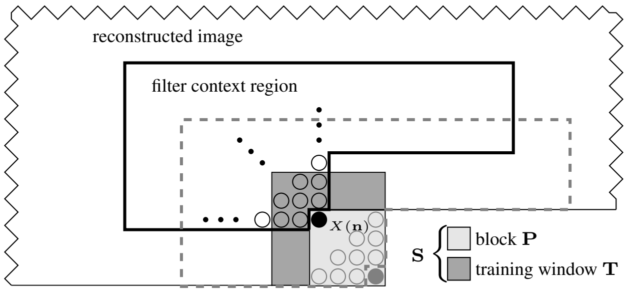

Fig. 3. Proposed training window and available region for filter support of SLSP method.

by choosing the k closest template patches (columns of $\mathbf{A}_c$), to the template $\mathbf{C}$ (vector $\vec{b}_c$), in terms of Euclidean distance. In order to compute the linear weights, LLE and NMF methods solve different constrained least-squares problems. In the case of LLE, the prediction problem is given by

$$\min_{\vec{x}_k} \|\vec{b}_c - \mathbf{A}_c^k \vec{x}_k\|_2^2 \quad \text{subject to} \quad \sum_m \vec{x}_{k_m} = 1 \tag{4}$$

where $\mathbf{A}_c^k$ denotes the submatrix of $\mathbf{A}_c$ containing the selected k-NN patches. In the case of NMF method, a constraint which imposes nonnegative weights is used.

By adjusting the value of k, different sparsity constraints can be tested. The value of k which produces the smallest prediction error should be explicitly signaled to the decoder. Note that template-matching (TM) algorithm can be viewed as a particular case of LLE method, in which the sparsity constraint is $k = 1$. Experimental results presented in [9] using H.264/AVC demonstrate that the described neighbor embedding methods are consistently superior to sparse prediction and TM methods.

4. PROPOSED SPARSE-LSP MODE

The proposed SLSP method combines the concept of sparse prediction with the LSP technique. In practice, SLSP is a redesigned LSP method which may exploit more meaningful data in a larger causal area than traditional LSP algorithms. Only a few positions of this causal area are selected through the k-NN technique to take part of the actual filter support.

In order to estimate the set of coefficients used to predict the block $\mathbf{P}$, SLSP uses a training procedure based on previously described LSP algorithm, in Section 2. The main differences are related to the shape of the training window and filter support. Since the linear coefficients are derived once per block and used to predict all the pixels of the block $\mathbf{P}$, SLSP uses a fixed training window, $\mathbf{T}$, whose shape is illustrated in Figure 3. In regard to the sparse filter support of SLSP, it can be defined in a large causal region which is illustrated in Figure 3, namely when positioned for the first pixel of the block (represented by thick black line) and for the last pixel of the block (represented by dashed gray line). Note that, the asymmetric shape of the filter context area is due to the fact that pixels in the right side of the block $\mathbf{P}$ are unavailable for most cases, *e.g.* for the last pixel of block, as illustrated by gray samples of Figure 3.

As discussed, not all samples of the referred context region are used in proposed linear prediction model, due to the sparse restriction. The use of a low order model not only reduces the algorithm's complexity, but also avoids the overfitting problem in which the model memorizes training data instead of learning its underlying relationships. The actual filter context positions used for linear filtering (*i.e.* positions that may use non null coefficients) are selected using the k-NN algorithm. SLSP uses the training window $\mathbf{T}$ for k-NN search procedure, similar to template $\mathbf{C}$ in NE-methods. The idea is to find the pixels in causal context area that exhibit the highest correlation with the pixels of training window $\mathbf{T}$. To better explain this procedure it is important to understand the connection between the LSP and NE methods, described in the following.

Let $\vec{t} = [X(\mathbf{n}_0 - \mathbf{h}(1)) \dots X(\mathbf{n}_0 - \mathbf{h}(M))]^T$ be the column vector containing the M pixels belonging to training window $\mathbf{T}$, where $\mathbf{n}_0$ denotes the first pixel of the predicting block and $\mathbf{h}(j)$ represents the relative position of the pixels in the training window. Consider the following matrix whose rows correspond to the N-pixel causal context region (before being sparsified) associated with each training window pixel, similar to matrix $\mathbf{C}$ defined in Section 2:

$$\mathbf{V} = \begin{bmatrix} X(\mathbf{n}_0 - \mathbf{h}(1) - \mathbf{g}(1)) & \dots & X(\mathbf{n}_0 - \mathbf{h}(1) - \mathbf{g}(N)) \\ \vdots & & \vdots \\ X(\mathbf{n}_0 - \mathbf{h}(M) - \mathbf{g}(1)) & \dots & X(\mathbf{n}_0 - \mathbf{h}(M) - \mathbf{g}(N)) \end{bmatrix},$$

where $\mathbf{g}(i)$ is the relative position of each pixel in the context region.

One may observe that each column i of matrix $\mathbf{V}$ is equivalent to a displaced version of the training window $\mathbf{T}$, where $\mathbf{g}(i)$, for $i = 1, \dots, N$, corresponds to the displacement vector. Considering this observation, vector $\vec{t}$ and matrix $\mathbf{V}$ can be interpreted as the template $\vec{b}_c$ and dictionary $\mathbf{A}_c$ of NE-methods (see Section 3.2). Actually, matrix $\mathbf{V}$ may be constructed similarly to dictionary $\mathbf{A}_c$, based on the patches existing in a predefined causal search window, as illustrated in Figure 2. The main advantage of SLSP is that generated dictionary is more complete. While NE-methods only include patches whose associated block exists in causal search area, the matrix $\mathbf{V}$ formulated for SLSP method contains all templates (with shape of training window) existing in a equivalent search window, plus those templates whose associated block overlaps the unknown block to predict. In practice, SLSP may be viewed as an extension to NE-methods, able to use an improved context which does not depend only on previous reconstructed pixels, but also on previous predicted pixels in the unknown block. Furthermore, SLSP does not use the linear filter restrictions of NE-methods (see equation (4)).

After estimation of the k matrix columns more correlated with the training window, $\vec{t}$, using k-NN method, the SLSP proceeds by solving the problem $\min(\|\vec{t} - \mathbf{V}_k \vec{a}\|_2^2)$, where $\mathbf{V}_k$ is a sub-matrix of $\mathbf{V}$ which only contains the k chosen columns, and $\vec{a}$ is the vector of the k coefficients. The estimated coefficients are used to predict each pixel of the block by linearly combining the k chosen pixels in the corresponding positions of the filter support:

$$\hat{X}(\mathbf{n}) = \sum_{i=1}^{k} a_i X(\mathbf{n} - \mathbf{g}(i)) \tag{5}$$

where $\mathbf{g}(i)$ gives the relative Euclidean coordinates of the chosen k context pixels, which vary for each encoded block.

5. EXPERIMENTAL RESULTS

The proposed prediction mode was implemented in the state-of-the-art HEVC standard (software HM-13.0). To accommodate the proposed prediction mode in HEVC framework, the directional mode 3 has been replaced by SLSP. Similarly, we implemented some prediction methods proposed in literature for comparison purposes. Three different versions of HEVC, using the LSP mode based on [5], the

Image	LSP		TM		LLE		SLSP	
	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
Barbara	0,23	-3,37	0,12	-1,86	0,27	-3,99	**0,43**	**-6,27**
Roof	0,07	-0,92	0,12	-1,61	0,33	-4,34	**0,44**	**-5,80**
Houses	0,03	-0,41	0,24	-2,98	0,34	-4,21	**0,39**	**-4,72**
Snook	0,01	-0,10	0,28	-3,83	1,00	-13,64	**1,05**	**-14,31**
Wool	0,13	-2,25	0,33	-5,70	0,56	-9,49	**0,62**	**-10,38**
Spincalendar	0,02	-0,37	0,41	-7,45	0,56	-9,90	**0,58**	**-10,18**
Building	0,02	-0,23	0,53	-7,57	**1,09**	**-15,39**	1,09	-15,01
Pan0_qcif	0,00	0,06	0,36	-5,25	0,92	-13,27	**1,12**	**-15,73**
Average	0,06	-0,95	0,30	-4,53	0,63	-9,28	**0,72**	**-10,30**

Table 1. BDPSNR (PSNR) and BDRATE (%) results of HEVC using LSP, TM, LLE and SLSP methods relative to the reference HEVC standard for the selected set of test images.

LLE prediction based on [9] and the template-matching (TM) algorithm, have been developed.

For LLE and TM methods, the template thickness (see Figure 2) was set to $T = 4$. The search window dimensions for these methods was defined to (see Figure 2): $D_V = P_Y + T + 64$ and $D_H = 128 + T + P_X$. In the case of SLSP method, the size of the training window has been chosen equal to the template of TM and LLE methods. Furthermore, the available context region, where matrix $\mathbf{V}$ is defined, was set to match the dimensions of the search window used in TM and LLE methods. This ensures that the three prediction methods can use the same causal information, differing only by the procedure that is used to generate the prediction block. The sparsity parameter k used in SLSP and LLE methods was set to $k = 10$, which corresponds to the model order used by LSP in[5].

Experiments were performed for two sets of sequences. The first one, represented in Figure 4, includes a selected set of images with high frequency features and complex textures, which are not well predicted by traditional directional modes. The second set includes the first frame of HEVC test sequences from 3 classes (B, C and D) as proposed in [12]. Only the luminance channel was encoded and evaluated in these experiments.

Tables 1 and 2 present Bjontegaard Delta PSNR (BDPSNR) and Bjontegaard Delta Rate (BDRATE) [13] results for the proposed SLPS method in HEVC standard, and the three previously mentioned HEVC versions, comparing the performances of each method relatively to the original HEVC standard, for the two proposed sets of test images, respectively.

In Table 1, one may observe that all methods improve HEVC performance, achieving coding gains superior to 1 dB for some images. These methods tend to perform very well in the presence of high frequency areas with complex textures, due to limitations of directional prediction with these features. SLSP presents the highest coding gains among all methods achieving average gains above 0.6 dB (9% of bitrate) over traditional LSP method, 0.4 dB (5% of bitrate) over TM algorithm and almost 0.1 dB (1% of bitrate) over the LLE method [9]. Experimental results using HEVC test sequences in Table 2 present a smaller rate-distortion performance gain, up to 0.2 dB (almost 1% of bitrate). As most of these sequences are smoother than the first set, containing less challenging features, original HEVC framework is able to exploit existing redundancy efficiently. Nevertheless, these results clearly demonstrate that the use of SLSP in HEVC framework produces equal or superior rate-distortion performance than the original standard.

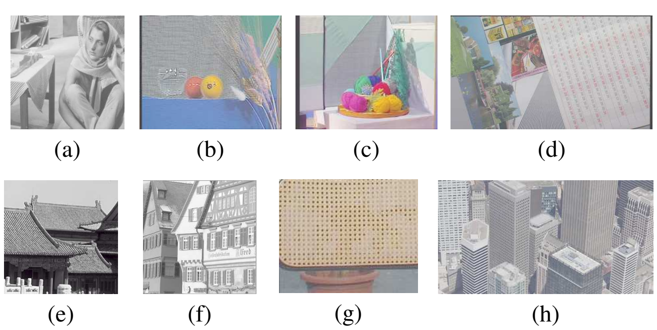

Fig. 4. Selected set of test images: (a) Barbara (512 × 512), (b) Snook (720 × 576), (c) Wool (720 × 576), (d) Spincalendar (1280 × 720), (e) Roof (512 × 512), (f) Houses (512 × 512), (g) Pan0_qcif (176 × 144) and (h) Building (1792 × 944).

Image	LSP		TM		LLE		SLSP	
	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
(D) BasketballPass	0,03	-0,41	0,08	-1,17	0,03	-0,52	**0,08**	**-1,20**
(D) BlowingBubbles	0,01	-0,05	0,01	-0,05	0,00	-0,03	**0,01**	**-0,17**
(D) BQSquare	0,00	0,05	0,03	-0,35	0,01	-0,09	**0,04**	**-0,40**
(D) RaceHorses	-0,02	0,15	-0,02	0,21	-0,03	0,38	**0,00**	**0,00**
(C) BQMall	0,00	-0,05	0,04	-0,68	0,04	-0,72	**0,05**	**-0,78**
(C) BasketballDrill	0,02	-0,50	0,13	-2,61	0,12	-2,51	**0,14**	**-2,96**
(C) PartyScene	0,01	-0,10	0,04	-0,49	0,05	-0,57	**0,05**	**-0,62**
(C) RaceHorses	0,01	-0,15	0,01	-0,24	0,00	-0,05	**0,01**	**-0,24**
(B) BasketballDrive	0,01	-0,26	0,06	-2,39	**0,07**	**-2,99**	0,06	-2,16
(B) BQTerrace	0,01	-0,17	0,11	-1,73	**0,17**	**-2,69**	0,10	-1,55
(B) Cactus	0,00	-0,03	0,05	-1,43	**0,06**	**-1,55**	0,05	-1,35
(B) Kimono1	0,00	-0,01	0,00	-0,06	**0,01**	**-0,18**	0,01	-0,14
(B) ParkScene	0,00	-0,03	0,01	-0,18	0,01	-0,20	**0,01**	**-0,27**
Average	0,01	-0,12	0,04	-0,86	0,04	-0,90	**0,05**	**-0,91**

Table 2. BDPSNR (PSNR) and BDRATE (%) results of HEVC using LSP, TM, LLE and SLSP methods relative to the reference HEVC standard for the first frame of HEVC test sequences.

In regard to computational complexity, coding times may increase one order of magnitude, due to search procedures and LS-optimization problems. All methods have comparable computational complexity, except TM which is less complex. In this work, we did not address computational complexity problem, however we believe that it could be highly improved by using more efficient implementations, CPU multi-core or GPU processing.

Despite the presented method uses a fixed model order and causal context region, similarly to the other evaluated methods, SLSP presents additional flexibility which gives some ability to operate similar to LSP, LLE or TM methods. This can be done through the use of simple constrains on the shape of the filter context region and filter order. This is an interesting property of the proposed SLSP method, which motivates the research on a generalized prediction approach able to exploit different image features using various linear models, defined by explicit constraints on the filter context.

6. CONCLUSIONS

A new LS-based prediction mode with sparse constraints on model context has been presented for intra image coding. SLSP predicts unknown pixels by linearly combining previous reconstructed and predicted pixels. Experimental results showed that when combined with HEVC, SLSP is able to improve rate-distortion results, mainly for contents presenting repeated and complex textures. Furthermore, as the SLSP filter context can be explicitly constrained to operate similarly to other linear prediction methods, it constitutes a worthy solution for a future generalized prediction method with improved adaptation to the image features.

7. REFERENCES

[1] ITU-T and ISO/IEC JTC1, *Advanced video coding for generic audiovisual services*, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2010.

[2] ITU-T and ISO/IEC JTC 1/SC 29 (MPEG), *High efficiency video coding*, Recommendation ITU-T H.265 and ISO/IEC 23008-2, 2013.

[3] Xin Li and M.T. Orchard, "Edge-directed prediction for lossless compression of natural images," *Image Processing, IEEE Transactions on*, vol. 10, no. 6, pp. 813–817, June 2001.

[4] D.C. Garcia and R.L. De Queiroz, "Least-squares directional intra prediction in H.264/AVC," *Signal Processing Letters, IEEE*, vol. 17, no. 10, pp. 831–834, 2010.

[5] D.B. Graziosi, N.M.M. Rodrigues, E.A.B. da Silva, S.M.M. de Faria, and M. de Carvalho, "Improving multiscale recurrent pattern image coding with least-squares prediction," *Image Processing, 16th IEEE International Conference on*, November 2009.

[6] L.F.R. Lucas, N.M.M. Rodrigues, E.A.B. da Silva, and S.M.M. de Faria, "Adaptive least squares prediction for stereo image coding," *Image Processing, 18th IEEE International Conference on*, pp. 2013–2016, September 2011.

[7] S. Cherigui, C. Guillemot, D. Thoreau, P. Guillotel, and P. Perez, "Hybrid template and block matching algorithm for image intra prediction," *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, pp. 781–784, 2012.

[8] M. Turkan and C. Guillemot, "Image prediction: Template matching vs. sparse approximation," *Image Processing (ICIP), 17th IEEE International Conference on*, pp. 789–792, 2010.

[9] M. Turkan and C. Guillemot, "Image prediction based on neighbor-embedding methods," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1885–1898, 2012.

[10] S. Cherigui, C. Guillemot, D. Thoreau, P. Guillotel, and P. Perez, "Correspondence map-aided neighbor embedding for image intra prediction," *Image Processing, IEEE Transactions on*, vol. 22, no. 3, pp. 1161–1174, 2013.

[11] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[12] Frank Bossen, "Common HM test conditions and software reference configurations," *Document JCTVC-L1100*, 2013.

[13] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T SG 16 Q.6 VCEG, Doc. VCEG-M33*, 2001.