Image Coding using Generalized Predictors based on Sparsity and Geometric Transformations

Luís F. R. Lucas∗§, Nuno M. M. Rodrigues⊥†, Eduardo A. B. da Silva§,
Carla L. Pagliari‡ and Sérgio M. M. de Faria∗⊥

∗Instituto de Telecomunicações, Portugal; †ESTG, Instituto Politécnico de Leiria, Portugal;
‡DEE, Instituto Militar de Engenharia; §PEE/COPPE/DEL/Poli, Universidade Federal do Rio de Janeiro, Brazil;
e-mails: luis.lucas,eduardo@smt.ufrj.br, nuno.rodrigues,sergio.faria@co.it.pt, carla@ime.ubr

Abstract—Directional intra prediction plays an important role in current state-of-the-art video coding standards. In directional prediction, neighbouring samples are projected along a specific direction to predict a block of samples. Ultimately, each prediction mode can be regarded as a set of very simple linear predictors, a different one for each pixel of a block. Therefore, a natural question that arises is whether one could use the theory of linear prediction in order to generate intra prediction modes that provide increased coding efficiency. However, such an interpretation of each directional mode as a set of linear predictors is too poor to provide useful insights for their design.

In this paper we introduce an interpretation of directional prediction as a particular case of linear prediction, that uses first-order linear filters and a set of geometric transformations. This interpretation motivated the proposal of a generalized intra prediction framework, whereby the first-order linear filters are replaced by adaptive linear filters with sparsity constraints. In this context, we investigate the use of efficient sparse linear models, adaptively estimated for each block through the use of different algorithms, such as Matching Pursuit, Least Angle Regression, Lasso or Elastic Net.

The proposed intra prediction framework was implemented and evaluated within the state-of-the-art high efficiency video coding standard. Experiments demonstrated the advantage of this predictive solution, mainly in the presence of images with complex features and textured areas, achieving higher average bitrate savings than other related sparse representation methods proposed in the literature.

Index Terms—Intra image prediction, sparse linear prediction, least squares regression, least angle regression, lasso, geometric transformations

I. INTRODUCTION

State-of-the-art video compression standards are based on a hybrid approach that comprises three stages: a prediction step, transform-based residue coding and entropy coding. The prediction methods play an important role in image and video coding algorithms, as they provide an efficient solution to reduce signal energy based on the previously encoded samples. In the case of video coding, inter-prediction methods, such as motion-compensation, tend to provide the highest coding gains by exploiting the temporal similarities between the current and previously encoded frames. However, in some situations, intra prediction is the only available solution, as in the cases of still image coding applications [1] or for the first frame and refreshing frames of a video sequence.

The idea of intra prediction is to use previously encoded samples from spatial neighbouring blocks to predict the unknown samples. The directional prediction [2] is the main solution for intra prediction adopted in the current state-of-the-art H.264/AVC [3], [4] and High Efficiency Video Coding (HEVC) [5], [6] standards. Its principle consists in projecting the reconstructed samples at block boundaries along specific directions, providing an efficient representation of the directional structures and straight edges, often present in natural images. While H.264/AVC only supports 8 directional modes, HEVC exploits 33 prediction directions. In addition to directional modes, these standards use the DC and planar modes which provide efficient prediction of smooth areas.

Despite its advantages, directional intra prediction presents some issues, mostly in the presence of complex regions, such as textured areas. These issues are an intrinsic limitation of directional modes, because they only exploit the first line of samples at the top and left neighbourhoods of the block. In order to better predict the textured areas, alternative methods that reuse repeated texture patterns along the image have been proposed in literature. The most common solutions are based on block matching (BM) [7] and template matching (TM) [8] algorithms. While BM algorithm requires some kind of signalling to indicate the optimal matched block in the causal reconstructed area, TM provides an implicit way to derive the predictor block. In TM algorithm, a reference template is formed using the causal reconstructed samples in the neighbour-hood of the block to be predicted. A search procedure is performed by comparing the reference template with each equally shaped candidate template existing in a predefined causal search window. The block predictor is given by the block associated to the candidate template which produces the lowest matching error. Improved variations of TM have been proposed for H.264/AVC, e.g., using an average of multiple predictors [9], using adaptive illumination compensation methods [10], or being combined with BM algorithms [11].

A related class of algorithms which has been widely investigated in literature for efficient intra image prediction is based on sparse representation. The research of these methods has been motivated by the assumption that natural image signals are formed by few structural primitives. Most solutions involve a linear combination of few patterns chosen from a large
and redundant dictionary. Either static or adaptive dictionaries can be used for sparse prediction [12], [13], [14]. Adaptive dictionaries are frequently formed by patches that exist in a causal window in the reconstructed image [13]. However, dictionary learning methods have also been shown to provide efficient results [14]. To avoid the transmission of linear coefficients, sparse prediction methods typically determine a sparse linear combination of dictionary codewords that best approximates a causal template area, surrounding the block to be predicted. Then, the same linear combination of the co-located pixels is used to generate the predictor for the unknown block samples. Common solutions used to find sparse linear combinations of dictionary codewords are based on matching pursuit algorithms [15].

In the context of sparse representation, the neighbour-embedding methods have been proposed for data dimensionality reduction, namely the locally linear embedding (LLE) and the non-negative matrix factorization (NMF) algorithms [16]. The main purpose of these methods is to approximate the block as a linear combination of \( k \)-nearest neighbours (\( k \)-NN) in a causal neighbouring region of the block. For efficient RD performance, these methods first find an approximation to a known template defined in the block neighbourhood, and then use the same coefficients to predict the unknown block by linearly combining the co-located samples. This procedure might be aided by a block correspondence method [17]. Neighbour-embedding prediction approach has been shown to provide better performance than sparse prediction algorithms when implemented in the H.264/AVC standard [16].

Linear prediction methods based on least-squares optimization have been also successfully proposed for intra image coding [18], [19]. Although they were originally proposed for lossless image compression, efficient adaptive approaches based on least-squares approximations soon emerged for lossy image coding [20], [21], [22], [23], [24]. Their main advantage is the ability to embed image characteristics into the linear filter support, for a better prediction result.

In this paper, we present a new linear prediction framework based on sparse linear models and geometric transformations (GTs) for the HEVC standard. This research work has two main contributions, specifically the generalization of the well-known directional prediction as particular case of the proposed method, and a comprehensive evaluation of different algorithms for sparse model estimation. We show that sparse models estimated using greedy approaches such as matching pursuit-based algorithms are less efficient for prediction applications. Conversely, not-so-greedy approaches, such as Least-Angle Regression or Lasso methods, are able to provide better prediction performance. Experimental results demonstrate the advantage of the proposed method based on sparse linear models, specially in the presence of complex textures and repeated patterns, where directional modes often fail.

The remaining of this paper is organized as follows. Section II reviews the existing linear prediction methods. In Section III we describe an alternative interpretation for directional prediction which gives rise to the proposed generalized intra prediction framework. Section IV reviews several algorithms used to enforce the sparsity constraint of the proposed linear model. Section V presents the proposed algorithm based on generalized optimal sparse predictors for the HEVC standard. Experimental results are presented and discussed in Section VI, and conclusions are presented in Section VII.

II. LINEAR PREDICTION METHODS

Long before the emergence of the H.264/AVC standard, the use of linear prediction for intra image coding has been investigated in literature with successful results [19], [25]. Its main purpose is to represent each image sample as a linear combination of the previous encoded samples in its neighbourhood. Due to the non-stationary statistics of natural images, context-based adaptive linear prediction approaches tend to be more effective.

A. Context-based adaptive linear prediction

The superiority of context-based adaptive linear prediction has been discussed in [18] from the viewpoint of the edge-directed property. This property is related to the fact that image samples around edges have a major influence in the coefficient estimation process when using least-squares regression. Consequently, such methods are able to adaptively learn the orientation of most edges, offering an efficient prediction result without requiring explicit edge detection. In what follows, we describe the least-squares prediction (LSP) algorithm as proposed in [18].

Let \( X(\mathbf{n}) \) denote the target image sample to be predicted, where vector \( \mathbf{n} \) corresponds to the two-dimensional spatial coordinates in the image. The linear prediction of \( X(\mathbf{n}) \), using an \( N \)-th order Markov model, as illustrated in Figure 1 (for \( N = 10 \)) is given by:

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

where \( \mathbf{g}(i) \) represents the relative position of each causal neighbouring sample that takes part of the filter context, and \( a_i \) are the linear coefficients.

LSP adaptively estimates the optimal linear coefficients using a least-squares training procedure based on a training window (TW), defined in the previously encoded region of the image. In order to learn the most relevant local image features, the TW typically surrounds the target sample \( X(\mathbf{n}) \), as illustrated in the example of Figure 1, with size equal to \( M = 2(T + 1) \) samples. The main advantage of such least-squares training procedure is that the linear coefficients do not need to be explicitly signalled to the decoder. Since the TW...
data is available in both the encoder and decoder sides, the decoder is able to replicate the training procedure to estimate the same coefficients.

Consider the column vector of TW samples \( y = [X(n - h[1]) \ldots X(n - h(M))]^T \), where \( h(j) \) represents the relative position of each sample in the TW. Define also the matrix \( C \) whose element \((j, i)\) corresponds to \( X(n - h(j) - g(i)) \), that represents the \( j \)th filter context sample associated to the TW sample \( X(n - h(j)) \). The least-squares minimization problem can be written as \( \min_{a}(\|y - Ca\|^2) \), where \( a = [a_1 \ldots a_N]^T \) corresponds to the linear coefficients to estimate. A well-known closed-form solution for this problem [18] is:

\[
a = (C^TC)^{-1}(C^Ty).
\]  

(2)

B. Lossy image coding using LSP

Due to its pixel-by-pixel based operation, LSP has found high applicability in lossless image coding, with efficient results reported in literature. Nevertheless, extensions of LSP for lossy image coding have also emerged, namely in the context of the most recent block-based video coding algorithms. In [21], the LSP algorithm described in [18] has been adapted for the block-based Multidimensional Multiscale Parser (MMP) algorithm. In this approach, LSP maintains a pixel-by-pixel procedure, recomputing a new set of linear coefficients for each sample to be predicted on the target block. When the TW or the linear filter include sample positions within the target block, where the reconstructed values are not available, the corresponding predicted values are used.

In [20], LSP is proposed for H.264/AVC standard using different TWs and filter contexts, which depend on the available neighbouring blocks. The linear model is updated in a block-by-block basis, i.e., the same set of coefficients is used through the whole block. Thus, only one estimation procedure, based on a fixed TW, is required for each encoded block. This solution avoids the use of predicted samples in the least-squares training procedure, although prediction samples can still be used in the linear prediction process. The main advantage of block-by-block model estimation is the lower computational complexity, when compared to pixel-by-pixel estimation approaches.

The recursive use of predicted samples for image prediction yields a sort of error propagation, since predicted samples inherently incorporate an error that is not present in the reconstructed samples. In order to reduce such error propagation, the method in [26] proposes a line-based linear prediction scheme for the H.264/AVC standard, combined with 1D transform for residue coding. In this method, a new linear model is estimated for each line. Since the residual can be encoded at line-level using 1D transform, the training procedure only uses reconstructed samples. Line-based residue coding also minimizes the use of predicted samples during the linear prediction process.

In [23], an alternative linear prediction method based on three-tap recursive filters is proposed by replacing the directional prediction framework of H.264/AVC and VP8 encoders. These filters provide a low complexity solution for image prediction assuming a 2D non-separable Markov model. The coefficients are estimated using an offline approach based on a “k-mode” iterative technique that minimizes the prediction error (or rate-distortion cost) over the training data. An improved version of this method based on four-tap recursive filters is presented in [24] with successful results for VP9 and HEVC encoders.

III. GENERALIZING DIRECTIONAL PREDICTION

The directional prediction framework of HEVC standard is based on the planar, DC and 33 angular modes [2], as illustrated in Figure 2a. In its procedure, the angular modes propagate the values from the neighbouring samples of the first column and first row into the target block in some specific direction. In practice, this process can be viewed as a linear combination of causal samples. The neighbouring regions used in HEVC intra prediction process are illustrated in Figure 2b. Note that, when not available, the left-down and top-right regions are generated using a padding procedure. In this section, we analyse the sample propagation of HEVC angular modes and present an alternative two-stage interpretation for their design, based on the concepts of linear prediction and GT. Moreover, we describe a generalization of such interpretation of angular prediction, which is the basis of the new intra sparse prediction framework proposed in this paper.

A. Two-stage interpretation of directional prediction

An angular mode can be easily written as a set of linear predictors, whose filter contexts and weighting coefficients depend on the predicted sample position and mode direction. This interpretation, however, has some issues related to the dependency existing between the filter context and the predicted sample position. Such a dependency results in a more complex linear model formulation, which does not provide valuable insights for designing a generalized linear prediction method. This has motivated an alternative interpretation of angular modes, which at a first-stage only considers three simple linear models associated to the angular 10 (horizontal), angular 18 (diagonal) and angular 26 (vertical) modes. These angular modes have a common characteristic: under the linear prediction interpretation they may be reproduced using a filter with fixed context.

Unlike most angular modes, that use a linear combination of two reference samples for prediction of each sample, the three referred angular modes simply copy the reference neighbouring samples along the block, in either horizontal,
diagonal or vertical directions. Such copy of reference samples can be performed by first order linear filters for lossy image prediction (see Subsection II-B), which only depend on the mode direction and use a weighting coefficient equal to 1. Considering the notation of Equation 1, these linear models can be written as:

\[
\hat{X}(n) = \hat{X}(n - g),
\]

where \(\hat{X}(\cdot)\) corresponds to a previously reconstructed or predicted sample, depending on the coordinate \(n\) and relative position of the first order filter, \(g\), which is \((1,0), (1,1)\) or \((0,1)\), for angular modes 10, 18 and 26, respectively. These filter contexts are illustrated in Figure 3, being designated by \(F_H^1\), \(F_D^1\) and \(F_V^1\). The gray coloured samples represent the target prediction sample.

In order to reproduce the remaining angular modes of HEVC encoder, we propose these linear filters (first-stage) with an additional processing stage based on GTs (second-stage). The idea is to distort the linearly predicted block, so that all angular modes can be reproduced from the three linear filter from Figure 3. Different sets of GTs are available, depending on the used linear filter, as indicated in Table I. For instance, in the case of filter \(F_H^1\), the \(T_H(d)\) GT is used to reproduce any angular mode between 2 and 10, controlled by parameter \(d\). For linear filter \(F_D^1\), two sets of GTs are defined, namely \(T_{Dh}(d)\) and \(T_{Dv}(d)\), which generate angular modes 11 to 18 and angular modes 18 to 25, respectively. The proposed GTs are affine, i.e. they preserve collinearity and distance ratios, and parameter \(d\) is used to control the magnitude of transformation. When \(d = 0\), there is no transformation on the output block (corresponding to the angular modes 10, 18 or 26). The proposed transformations that generate all the HEVC angular modes are illustrated in Figure 4. For each linear filter, the linearly predicted output (labelled by LPB) and the transformed block (labelled by TB) are shown. For clarity, a representative edge is visible in the illustrated blocks.

The \(T_H(d)\) and \(T_V(d)\) transformations correspond to vertical and horizontal block skews, respectively. As observed in Figure 4, the straight lines generated by horizontal and vertical first-order linear filters (\(F_H^1\) and \(F_V^1\)) remain straight and also parallel after the block skew transformation. In practice, the \(T_H(d)\) (or \(T_V(d)\)) transformation simply displaces each vertical line (or horizontal line) of the block by an amount proportional to its distance to the horizontal block margin (or vertical block margin). Assuming two-dimensional transformations represented by:

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} = \begin{bmatrix}
    t_1 & t_2 & t_3 & t_4 \\
    1 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
    x \\
    y
\end{bmatrix},
\]

the proposed affine transformations can be written as:

\[
T_H(d) = \begin{bmatrix}
    1 & 0 & 0 & 0 \\
    d/32 & 1 & 0 & 0
\end{bmatrix},
\]

\[
T_V(d) = \begin{bmatrix}
    1 & 0 & -d/32 & 0 \\
    0 & 1 & 0 & 0
\end{bmatrix},
\]

where \(d\) is the parameter that defines the amount of skewing. Note that \(d\) is negative for \(T_V(d)\) transformation because the skewing is applied to the left direction as shown in Figure 4c. To ensure that the transformed block fills all samples of the target block, the linear filter step propagates both the left and left-down block neighbourhoods in case of \(F_H^1\), and both the top and top-right block neighbourhoods in case of \(F_V^1\). When fractional sample positions need to be derived, a linear combination of the two nearest integer samples is performed. The values of \(d\), used in GT, that correspond to HEVC angular modes are given in Table II.

Regarding \(T_{Dh}(d)\) and \(T_{Dv}(d)\) transformations, a slightly different approach is used. The linearly predicted block (LPB) is divided into two triangles, which are independently transformed. These transformations should preserve the continuity between the block and the left and top neighbourhoods, in order to follow the propagation effect of angular modes 11 to

---

**Table I**

<table>
<thead>
<tr>
<th>Filter</th>
<th>Neighbourhood</th>
<th>Transform</th>
<th>Angular modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F_H^1)</td>
<td>left, left-down</td>
<td>(T_H(d))</td>
<td>2 to 10</td>
</tr>
<tr>
<td>(F_D^1)</td>
<td>left, top-left, top</td>
<td>(T_{Dh}(d) / T_{Dv}(d))</td>
<td>11 to 18 / 18 to 25</td>
</tr>
<tr>
<td>(F_V^1)</td>
<td>top, top-right</td>
<td>(T_V(d))</td>
<td>26 to 34</td>
</tr>
</tbody>
</table>

**Table II**

<table>
<thead>
<tr>
<th>Value of (d)</th>
<th>0</th>
<th>2</th>
<th>5</th>
<th>9</th>
<th>13</th>
<th>17</th>
<th>21</th>
<th>26</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_H(d)) modes</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>(T_V(d)) modes</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
</tbody>
</table>
25. In the case of $T_{D_h}(d)$ transformations, the triangle below the block diagonal is scaled ($T_{D_h}(d)$), while the triangle above the diagonal is horizontally skewed ($T_{D_v}(d)$). These affine transformations are given by:

$$T_{D_h}(d) = \begin{bmatrix} d/32 & 0 \\ 0 & 1 \end{bmatrix}, \quad T_{D_v}(d) = \begin{bmatrix} 1 & d/32 \\ 0 & 1 \end{bmatrix}, \quad (6)$$

where the parameter $d$ defines the intensity of the transformations. Similarly, $T_{D_v}(d)$ uses scale and skew transformations on the triangles above and below the block diagonal, respectively. The difference is that the transformations are made along the vertical direction, instead of the horizontal one. The values of $d$ used in GT that result in HEVC angular modes are given in Table III.

Based on the presented first-order linear filters and GTs, it is possible to demonstrate that the proposed two-stage approach provides exactly the same prediction results as the HEVC angular prediction for most modes. The only exception are angular modes 11 to 25, whose interpolation process may present a slight difference. Unlike the proposed method, HEVC angular prediction does not consider all the reference samples of the left and top neighbouring regions to generate modes 11 to 25. This is because HEVC performs angular prediction based on a single line of reference samples (left or top neighbouring lines depending on the prediction mode), which is extended with a sub-set of samples from the other (perpendicular) line of reference samples. Implementation details of HEVC angular modes can be found in [2].

### B. Generalizing prediction using sparse linear predictors

The presented two-stage formulation of directional prediction demonstrates that first-order linear prediction models can be used to generate angular intra modes when combined with GTs in a second stage. An interesting approach would consist in replacing the first-order predictors by optimal linear predictors, that would be able to reproduce directional prediction as a particular case. This would require the optimal predictors to exploit a larger causal area than the first order filters. Unlike the traditional linear prediction methods, discussed in Section II, which tend to use small $N$th-order Markov models, based on closest neighbouring samples, we propose a more general method that is able to use predictors at larger distances, similarly to sparse prediction methods. This observation led us to a generalized intra prediction framework based on optimal sparse predictors, which is described in what follows.

The proposed method generalizes the first-order filters using higher-order ones defined in an augmented context area, as shown in Figure 5, and adaptively estimated in a causal TW. In the example of Figure 5, the context area size, $D_f = 5$, results in the filters $F_{H}^{22}$, $F_{D}^{22}$, and $F_{V}^{22}$, with the superscript indicating the maximum order of the filter. The use of three distinct filter shapes allows to select predictors either from regions in the left or top sides of $\hat{X}(n)$. Note that one filter cannot include predictors from both down-left and top-right regions, because they are not simultaneously available during the prediction process. The availability of these predictors depends on the prediction scanning order, which can be done from top-to-bottom or left-to-right. Thus, in order to efficiently use filter $F_{H}^{k}$, a top-to-bottom scanning order is employed. For filter $F_{V}^{k}$, the left-to-right scanning order is used. In the case of filter $F_{D}$, the selected scanning order is not relevant, since the predictors from the left-top region are always available in both cases. However, in the case of filters $F_{H}^{k}$ and $F_{V}^{k}$ there are still some situations in which predictors may not be available. For instance, the predictors belonging to the two bottom rows of filter $F_{H}^{22}$ (or the two rightmost columns of filter $F_{V}^{22}$) may not be available when predicting the two bottom rows (or the two rightmost columns) of the block. This is an issue when non-null coefficients are defined in these positions. To solve it, the nearest available reconstructed samples in the missing rows and columns are used.

Unlike the first-order filters, that use a fixed coefficient equal to 1 to reproduce angular modes (see Equation 3), the proposed augmented filters are locally estimated in a causal TW, based on least-squares regression problem with a sparsity constraint. The training procedure is performed once per Prediction Unit (PU) in HEVC encoder, and the estimated linear model is used to predict all the samples of the PU. For each type of filter context presented in Figure 5, a different TW is defined, as shown in Figure 6. In the proposed method, the size of the TW scales with the block size, but dimension $D_{TW}$ is fixed, being equal to 4. Note that GTs are applied after linear prediction and the training procedure is performed without knowledge of GTs. The idea of linear prediction is to learn the image features from the known reconstructed data to provide an efficient approximation of the block. On the other hand, the GTs allow to perform an explicit modification of the output of linear prediction, providing an additional mechanism to represent statistical changes that cannot be derived from the causal information, such as sudden change of the direction of an image edge in the unknown block.

Although the proposed linear models may use all the samples within the $D_f \times D_f$ context area, it is not expected that they are all relevant for the prediction process. The least-squares-based training procedure should attribute higher weighting coefficients for more important samples and lower

### Table III

<table>
<thead>
<tr>
<th>Value of $d$</th>
<th>4096</th>
<th>1638</th>
<th>910</th>
<th>630</th>
<th>482</th>
<th>390</th>
<th>315</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{D_h}(d)$ modes</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>$T_{D_v}(d)$ modes</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
</tr>
</tbody>
</table>

![Fig. 5. High-order linear filters for the proposed generalized intra prediction framework.](image-url)
Fig. 6. TW regions used to estimate the filters $F^k_{H}$ (a), $F^k_{D}$ (b) and $F^k_{V}$ (c), used in the proposed intra prediction framework.

weighting coefficients to the less relevant samples. However, when the number of filter coefficients (or predictors) is large in relation to the number of training samples (observations), over-fitting issues may occur. In these cases, the estimated models tend to memorize the training data rather than learn the underlying linear model. As a consequence, these models may present a poor predictive performance. This problem is aggravated when the filter size is larger than the TW size, resulting in an under-determined problem, with infinite solutions.

In order to overcome these problems, regularization constraints are typically added to the ordinary least-squares regression problem. We investigated the use of a sparsity-based regularization function for the least-squares problem. The classical LSP optimization problem (see Equation 2) using a sparsity constraint can be written as:

$$\arg\min_{a} \|y - Ca\|_2^2 \quad \text{subject to} \quad \|a\|_0 \leq t,$$  
\[ (7) \]

where $a$ is the sparse solution vector, $C$ is the correlation matrix computed as described in Section II, $y$ is the training data and $t$ is a bound on sparsity (number of non-zero coefficients). In general, searching the exact solution for this problem is NP-hard [27], and thus computationally intractable. To solve this problem, there are various algorithms that provide approximated solutions with feasible complexity. Some of these algorithms are reviewed in the next section.

IV. SPARSE MODEL ESTIMATION ALGORITHMS

In this section we review some algorithms to estimate approximate solutions to the constrained least-squares problem described by Equation 7. These methods include the iterative greedy matching pursuit-based algorithms [15], [28], as well as less greedy solutions, such as Least Angle Regression (LAR) [29], Lasso [30] and Elastic Net (EN) [31]. From this point onwards, we will use the notation of Equation 7, assuming that $y$ is centred to have zero mean, and the columns of $C$ are normalized to unit $l_2$-norm and zero mean.

A. Matching Pursuit algorithms

The Matching Pursuit (MP) [15] algorithm provides greedy variable selection, by searching the predictor (column of matrix $C$) that best fits the residual at each iteration and adding it to the model. The $l_2$-norm of the objective function error (i.e. the residual) is often used as the stopping criterion.

Since MP does not orthogonalize the residue in relation to the current model predictors, the same predictor can be chosen again in later iterations. Due to this issue, MP may potentially require a large number of iterations to converge. The Orthogonal Matching Pursuit (OMP) [28] is an improved version of MP which solves the predictor repetition and the convergence issue. OMP recalculates all the support coefficients at each iteration, by solving an ordinary least-squares (OLS) problem over the support augmented with the new predictor. This way, it projects the signal on the span of the predictors already chosen, ensuring that new residual is orthogonal to the already chosen predictors. Although OMP requires more operations than MP, it tends to converge in fewer iterations, due to the residue orthogonalization step.

B. Least Angle Regression

Due to their greedy nature, MP methods tend to choose only one predictor in the presence of a group of highly correlated predictors, excluding sometimes useful predictors. A less greedy version of the traditional forward selection methods is the LAR algorithm proposed in [29].

The LAR algorithm can be briefly described in what follows. Similarly to forward selection algorithms, LAR starts with all coefficients equal to zero and finds the predictor, $c_1$, which is most correlated with the response $y$. To find the second predictor, LAR moves along the direction of $c_1$, updating the residual along the way, until some other predictor, $c_2$, becomes as correlated as $c_1$ with the current residual. These two predictors form the most correlated set, that is named as active set. Then, LAR proceeds along a new direction that is equiangular to both predictors $c_1$ and $c_2$ (i.e. along the least angle direction), and it stops when a third predictor, $c_3$, has as much correlation as $c_1$ and $c_2$ with the current residual, adding it to the active set. This process continues by using the equiangular distance between the three found predictors, until a fourth predictor is obtained, and so on. LAR terminates its procedure when the number of desired predictors is achieved or when the full solution path is obtained. The LAR algorithm is very efficient, since it can solve the whole solution path (i.e. the sequence of all solutions from 0 up to $p$ predictors), at the cost of solving the OLS for the full set of $p$ predictors.

C. Lasso regression

The Least Absolute Shrinkage and Selection Operator, (Lasso) [30], [32] has been proposed in statistics literature for estimation of linear models, using an $l_1$-norm constraint on the coefficients’ vector. The Lasso constraint is a convex relaxation of the intractable $l_0$-norm problem described in Equation 7, providing approximate solutions to this problem with reasonable computational complexity. A useful unconstrained formulation of the Lasso problem for Equation 7, based on a Lagrangian multiplier $\lambda$, is given by:

$$\arg\min_{a} \frac{1}{2}\|y - Ca\|_2^2 + \lambda\|a\|_1.$$  
\[ (8) \]

Due to the nature of $l_1$-norm, Lasso enjoys two main features: model selection and coefficient shrinkage. The model selection feature guarantees that estimated solutions using
Lasso are sparse and thus it can provide accurate approximations to the original $l_0$-norm problem formulation. With $\lambda = 0$, the problem in Equation 8 becomes the classical OLS minimization problem. As $\lambda$ increases, Lasso tends to shrink the OLS coefficients towards zero, which results in decreased coefficients’ variance and a more biased solution. Consequently, the solution becomes more stable and the predicted values present lower variance, improving the overall prediction accuracy.

D. Elastic Net regression

The EN regression [31] has been proposed as an extension to the Lasso formulation, in order to improve its prediction performance for some scenarios, e.g. when the number of predictors ($p$) is much larger than the number of samples ($n$). In such a scenario, Lasso can select at most $n$ out of $p$ predictors, which is a problem when the number of relevant predictors is superior to $n$. Another interesting feature of EN is the ability to include groups of correlated predictors into the model, which is often desirable in prediction scenarios. In contrast to EN, Lasso lacks the grouping effect, selecting only one arbitrary predictor from a group of correlated ones.

The EN problem can be formulated as an extended version of Lasso, adding an $l_2$-norm component (Ridge penalty) to the $l_1$-norm penalty of Lasso. Considering the Lasso problem of Equation 8, its EN extension can be written as:

$$
\arg\min_a \frac{1}{2} ||y - Ca||^2_2 + \lambda_1 ||a||_1 + \lambda_2 ||a||^2_2 .
$$

Unlike the model estimation methods previously presented, the EN uses two parameters to adjust the influence of the used penalties: the sparsity parameter $\lambda_1$ and the grouping parameter $\lambda_2$. The EN problem can be solved using modified versions of standard algorithms for the resolution of Lasso problem. In such methods, the $\lambda_1$ parameter defines the early stopping condition, while $\lambda_2$ modifies the input dataset to the algorithms.

V. Adaptive Sparse Predictors for HEVC

The proposed two-stage intra prediction framework has been incorporated into HEVC standard by replacing the 33 angular intra prediction modes and keeping the original planar and DC modes. The linear filtering stage is implemented as described in Subsection III-B, using a variable maximum filter size (given by parameter $D_f$). For each encoded PU, the filter size should be explicitly signalled to the decoder. The filter size is important because it indicates the maximum distance of non-zero filter coefficients from the predicted sample. The use of the $D_f$ parameter can be comparable to a step penalty function, which gives equal importance (weight 1) to samples within a squared context area, discarding the outside samples (weight 0). As an alternative, more complex and smooth penalties might be used, for instance, smooth functions that exponentially decrease the importance weight of the coefficient when its distance to $X(n)$ increases.

In this implementation, we used the straightforward step-based penalty solution, where $D_f$ indicates the maximum filter size, as illustrated in Figure 5. Thus, the maximum filter size is adaptively selected and explicitly transmitted to the decoder. We minimize the amount of signalled information by using only two possible values for $D_f$, specifically 1 and 31, which are represented by a binary flag. A larger number of filter sizes ($D_f$ values) could be used, however, it would require additional signalling bits.

The use of $D_f = 1$ corresponds to the filter contexts shown in Figure 3 which provide similar results as the HEVC angular modes (using coefficient equal to 1). For $D_f = 31$, up to 960 non-zero coefficients are available, but only a few should be selected by the model estimation algorithms described in Section IV. The use of larger filter context sizes allows to include more predictors located at larger distances, being favourable when the image presents correlations between distant samples. In general, we observed that the use of larger filter contexts does not present a negative impact in coding performance, even when there are no correlations between distant samples. Thus, we selected a larger filter context size ($D_f = 31$) in order to be useful for a wide range of test images. The only drawback of the larger filter sizes is the increased computational complexity associated. In regard to the TW size we used $D_{TW} = 4$ (see Figure 6). The use of smaller TW sizes showed to produce an inferior coding performance, while the use of larger TWs did not have a significant impact in the coding results.

A point of concern for the estimation algorithms used to find the optimal linear models is the choice of an appropriate stopping condition. In the case of the Lasso problem, the stopping criterion is related with the $\lambda$ parameter. The most straightforward solution for this problem is to use a fixed number of non-zero predictors, similar to traditional LSP methods. In such an approach, which we call KNP ($k$ non-zero predictors), the optimization algorithm stops whenever $k$, the predefined number of predictors, is reached.

In addition to trivial KNP method, we investigated alternative early stopping criteria, based on adaptive solutions. Statistical methods, such as Akaike Information Criterion (AIC) [33], Bayesian Information Criterion (BIC) [34] or Mallows $C_p$ [35], have been used as stopping criteria for efficient model selection. However, in our scenario, where the number of predictors (maximum filter context size) is usually larger than the number of training samples (i.e. $p \gg n$), these methods are not recommended, because they turn into a measure of training error only. A popular solution to adaptively find the optimal number of non-zero coefficients, when only training data is available, is the Cross-Validation (CV) method [36]. However, because the CV has high computational complexity, we decided to investigate two other methods to find the optimal number of predictors under the evaluated model estimation algorithms.

The first adaptive solution is based on the Variation of the Training Error (VTE) at each iteration. The idea behind this method is to stop the optimization algorithm when the variation of mean squared error in the TW is inferior to a predefined threshold, $\rho_E$, in terms of percentage. The second used method is based on the Geometrical Stopping Criterion (GSC), originally proposed in [37] for the LAR algorithm.
GSC is based on the fact that the angle, $\theta_{j,n}$, between the residue and the predictor $c_j$ tends to 90° as the algorithm iteration $n$ increases. Thus, the stopping criterion is defined by $\Delta \theta_n \leq \sigma_{\theta}$, where $\theta_n = \theta_{1,n} \ldots \theta_{j,n} \ldots \theta_{p,n}$, $\Delta \theta_n = \max(\theta_n) - \min(\theta_n)$ and $\sigma_{\theta}$ is the standard deviation of the angles at first iteration. In practice, this method forces the algorithm to stop when the difference between the maximum and minimum angles is smaller than their standard deviation at the first iteration. Since this approach is static, we made a simple modification that allows for tuning, introducing the parameter $\rho_{\theta}$, yielding $\Delta \theta_n \leq \rho_{\theta}\sigma_{\theta}$.

The optimal tuning parameters for KNP, VTE and GSC stopping criteria, the parameters $k$, $\rho_{E}$ and $\rho_{G}$, respectively, are fixed for the whole image coding process. We obtained the optimal values through experimental tests (see Section VI) and we verified that they are approximately consistent for several images with different characteristics. Thus, the only encoded side information in the proposed intra prediction framework is the linear filter configuration ($F^k_H$, $F^k_D$ or $F^k_V$), the selected GT, and the maximum filter context size ($D_f$ value).

In order to encode these symbols we have made minimal changes to the HEVC bitstream syntax. The linear filter configuration and GT are encoded using the existing symbols and entropy coding functions originally developed for the angular modes, including the Most Probable Modes scheme and the Context-Adaptive Binary Arithmetic Coding (CABAC) [38]. This is possible because all the combinations between the three available filter context configurations and GTs result in a total of 33 modes, which for the particular case of first-order filters correspond to the HEVC angular modes (see Tables I, II and III). Regarding the filter context size, an additional bit, that indicates whether $D_f$ is 1 or 31, is also signalled. For the case of Planar and DC modes, this flag is omitted. The statistical redundancy of this binary flag is exploited using CABAC, based on 33 contexts that correspond to each available mode, being initialized with uniform distribution. A brief description of the proposed method is presented in Algorithm 1.

VI. EXPERIMENTAL RESULTS

The proposed intra prediction framework has been evaluated for still image coding using the HEVC reference software HM-14.0 version\(^1\). Four coding approaches using different predictor estimation algorithms have been used in the experiments: HEVC-OMP, HEVC-LAR, HEVC-LASSO and HEVC-EN. The used test images are organized in three sets. Set 1 (see Figure 7) includes test images with complex textured areas and repeated patterns, which tend to be better exploited by the proposed methods. Set 2 includes first frame of HEVC test sequences from classes B, C and E [39]. Set 3 corresponds to the test images proposed for the Image Compression Grand Challenge at ICIP 2016\(^2\), converted from RGB to YUV420 color space using Image Magick tools\(^3\). Only the luminance component has been considered in the experiments. In the following, we analyse the rate-distortion (RD) performance of these algorithms as well as their optimal parameters.

---

\(^1\)http://www02.smt.ufrj.br/~eduardo/hevc_gosp
\(^2\)http://jpeg.org/static/icip_challenge.zip
\(^3\)http://www.imagemagick.org

---

Algorithm 1 Proposed prediction algorithm (encoder).

**Input:** causal reconstructed image of current PU;

**Output:** predicted block $\hat{P}$ for current PU;

1: evaluate Planar and DC modes and compute RD costs;
2: use first-order filters: $F^3_H$, $F^3_D$ and $F^3_V$ (see Figure 3), to generate 3 first-stage output blocks;
3: for each available GT $t$ do
4: use GT $t$ over one of the previous generated first-stage output blocks (selected one depends on $t$ - see Table I);
5: save previous prediction result and compute RD cost;
6: end for
7: estimate sparse models (e.g. using LASSO/KNP) for 3 filter context configurations with $D_f = 31$ (see Figure 5);
8: use estimated $k$-sparse filters: $F^k_H$, $F^k_D$ and $F^k_V$, to generate the 3 first-stage output blocks;
9: for each available GT $t$ do
10: use GT $t$ over one of the previous generated first-stage output blocks (selected one depends on $t$ - see Table I);
11: save previous prediction result and compute RD cost;
12: end for
13: select prediction result with lower RD cost and set $\hat{P}$;
14: transmit the selected mode (Planar, DC or linear filter/GT) using the HEVC directional prediction signalling;
15: transmit an additional binary flag, indicating $D_f$ value.

---

Fig. 7. Test images of set 1: (a) Barbara ($512 \times 512$), (b) Barb2 ($720 \times 576$), (c) Bike ($512 \times 512$), (d) Snook ($720 \times 576$), (e) Wool ($720 \times 576$), (f) Houses ($512 \times 512$), (g) Pan0_qcif ($176 \times 144$) and (h) Roof ($512 \times 512$).

A. Effect of sparsity constraints

In this subsection, we discuss the influence of the sparsity constraints on the linear prediction models solved by OMP, LAR and Lasso algorithms. In order to evaluate the importance of the sparsity constraint, Barbara image was encoded using a modified version (explained below) of HEVC-OMP, HEVC-LAR and HEVC-LASSO with $QP=27$, and the KNP stopping method. Different sparsity levels, in the range of $k = 1, ..., 100$ were tested.

In Figure 8 we present some statistics about training and prediction error, as well as model usage frequency versus the number of non-zero predictors ($k$). These results only correspond to those blocks chosen by the encoder to be predicted by the proposed method using the high-order filter case with $D_f = 31$, as defined in Section V, that are the cases where the presented optimization algorithms are effectively used. In order to better interpret the error of linear prediction,
we disabled the GTs for the high-order filters.

The average training error of Figure 8 corresponds to the average MSE obtained in the TW for the blocks predicted using sparse linear predictors. Such training error gives the average modelling capabilities of the sparse linear predictors in the TWs, defined around left and top regions of the predicted block. The results of Figure 8a show that the average approximation error tends to decrease as the number of non-zero predictors ($k$-value) increases, regardless of the predictor estimation algorithm, as expected.

According to Figure 8a, the training error obtained by OMP algorithm tends to be lower than the one of LAR and Lasso, for the same sparsity value. This can be justified by the highly greedy behaviour of OMP, which provides a good approximation using less iterations, i.e. less predictors. The average prediction error on the block, shown in Figure 8b, presents a different behaviour than the one of the training error. The greedy OMP algorithm tends to provide better prediction performance (lower prediction MSE) for smaller $k$-values. Regarding LAR and Lasso algorithms, Figure 8b shows that these methods tend to provide better prediction performance for larger $k$-values, more specifically when $k > 10$.

It is important to note that, in Figure 8b lower average prediction error does not imply better RD performance. The presented results correspond to the average prediction error of the blocks chosen by the encoder, which may differ for each experiment. Thus, to better understand these results, the number of times that the sparse linear prediction methods were used (usage frequency) is illustrated in Figure 8c. We may observe that OMP-based prediction is more frequent when smaller $k$-values are used (with maximum at $k = 4$), decreasing its frequency as more predictors are considered. These results are somewhat in line with the average prediction errors of OMP algorithm, so we should expect a better overall RD performance when fewer predictors are used.

In regard to LAR and Lasso algorithms, the usage frequency results tend to improve as the number of non-zero predictors increases, until reaching a maximum around $k = 15$. For higher $k$-values the usage frequency of these methods decreases again, mainly for the LAR algorithm. The prediction errors, shown in Figure 8b, of LAR and Lasso methods also reach their minima close to $k = 15$, although they do not increase significantly for higher $k$-values. These observations suggest that there is an optimal number of non-zero coefficients, which provides the highest RD performance. When we compare the results for the three methods, we observe that Lasso provides the best prediction results and higher usage frequency. In the next subsection we analyse the RD performance of these methods using different sparsity levels and three different stopping criteria.

### B. Regularization parameters for optimal RD performance

Figure 9 presents the Bjontegaard Delta Bitrate (BDRATE) [40] results as a function of the number of non-zero predictors, using HEVC-OMP, HEVC-LAR and HEVC-LASSO encoders with KNP criterion and test set 1. BDRATE results were computed against the original HEVC standard. In the case of the greediest approach, the HEVC-OMP, one may see that bitrate savings tend to be more significant (i.e. higher negative BDRATE values) for small $k$-values. Less greedy approaches, specifically the HEVC-LAR and HEVC-LASSO schemes, present a different behaviour compared to HEVC-OMP. Their optimal RD performance is obtained with the number of predictors $k = 15$, that is rather consistent for all tested images. Moreover, it can be observed that the maximum bitrate savings using these schemes are significantly higher than using HEVC-OMP. These results support the observations presented in Subsection VI-A.

The results using the VTE stopping criterion are illustrated in Figure 10 for HEVC-OMP, HEVC-LAR and HEVC-LASSO schemes and test set 1, with BDRATE results computed against the original HEVC standard. The performance of HEVC-OMP approach increases as $\rho_E$ goes from 1% to 10%. Between 10% and close to 100% its performance is not consistent, as it may decrease or not, depending on the test image. Results for the HEVC-LAR and HEVC-LASSO schemes reveal that an optimal value for $\rho_E$, that works for all tested images, can be found. Regarding HEVC-LAR, the optimal VTE stopping criterion value is around 0.4%, while HEVC-LASSO performs better when $\rho_E$ is 0.1%.

Figure 11 presents the BDRATE results when the geometrical stopping criterion is used on HEVC-OMP, HEVC-LAR and HEVC-LASSO schemes with test set 1. In this test, HEVC-OMP seems to perform more efficiently on average for
$\rho_G = 0.5$. HEVC-LAR and HEVC-LASSO achieve the best results for most images at $\rho_G = 2$ and $\rho_G = 1$, respectively.

Another solution investigated in this paper is based on the EN method (HEVC-EN). As explained in Section V, EN can be reproduced using the Lasso algorithm with modified input matrices. The first parameter of EN, $\lambda_1$, is related with the stopping criterion of proposed iterative Lasso algorithm, while the second parameter, $\lambda_2$, associated to $l_2$-norm constraint, is incorporated in the modified input matrices. Figure 12 presents the BDRATE results of HEVC-EN for test set 1, using optimal stopping parameter for Lasso ($k = 15$ non-zero predictors) and a variable value for $\lambda_2$. In order to analyse the influence of $l_2$-norm constraint, BDRATE values have been computed relative to the Lasso results. Note that Lasso is a particular case of the EN method, specifically when $\lambda_2 = 0$.

From Figure 12, we may observe that adding $l_2$-norm constraint to Lasso problem does not provide consistent gains. For a few cases, the BDRATE performance may slightly increase, such as the Pan0_qcif image. However, other images present a decrease of RD performance or insignificant gain, independently of the $\lambda_2$ value. Such results lead us to conclude that the Lasso approach is preferable to the EN method. Furthermore, the EN requires an extra regularization parameter, $\lambda_2$, which adds complexity without consistent advantages.

In order to provide a clear comparison between the OMP, LAR and Lasso, Table IV shows the previously presented
For Lasso, we set choice for LAR has been $k$ for KNP, VTE and GSC methods, respectively. Similarly, the greedy OMP method presents the worst performance, being significantly inferior to LAR and Lasso methods. These results also reveal that the use of $l_1$-norm constraint (Lasso) tends to be more effective than LAR procedure for image prediction.

When we compare to the Ridge regression algorithm, we observe that sparse estimations methods tend to be more effective. However, the non-greedy OMP method shows to be inferior than the non-sparse Ridge regression approach.

Regarding the results for different stopping criteria, they depend on the optimization algorithm. For the case of OMP, there is no better stopping criterion, since different images reach optimal RD results using different stopping criteria. In respect to the not-so-greedy algorithms, LAR and Lasso, we observe more consistent results for all images. The VTE-based stopping criterion demonstrates an inferior performance than the one of traditional KNP and adaptive GSC methods. Both LAR and Lasso present the best average results using the traditional KNP solution. For few images, GSC provides a superior RD performance. However, such performance difference is insignificant in all cases (less than 0.1%).

The better performance of KNP stopping criterion over the adaptive VTE method can be explained by the poor relation between training and predictor errors. As can be seen in Figure 8, the training error tends to decrease monotonically as more predictors are added to the model. On the other hand, the prediction error has a different behaviour and it may increase or decrease with the inclusion of new predictors. This uncertain relationship between training and prediction errors does not guarantee that VTE stopping method is the best one.

### C. RD performance relative to other intra prediction methods

In this section we compare the proposed prediction framework based on the generalized sparse optimal predictors (GOSP) with some existing state-of-the-art image prediction methods. For this purpose, we used the best setup for the GOSP method, based on Lasso algorithm with KNP stopping criterion using $k = 15$ non-zero predictors. The intra prediction methods compared in these experiments include the traditional LSP algorithm, sparse-LSP (SLSP) [42], TM and neighbour-embedding methods. The traditional LSP has been used as described in Section II for block-based image prediction using $k = 10$ closest predictors. The recent SLSP method has a close relation with this work, using linear prediction with sparse models, estimated by the $k$-nearest neighbours method. The SLSP can be viewed as a generalization of the neighbour-embedding methods, which have been proposed in [16] for the H.264/AVC encoder. In this section, the LLE neighbour-embedding method [16] is also compared with our

---

### Table IV

<table>
<thead>
<tr>
<th>Image</th>
<th>OMP-KNP</th>
<th>OMP-VTE</th>
<th>OMP-GSC</th>
<th>LAR-KNP</th>
<th>LAR-VTE</th>
<th>LAR-GSC</th>
<th>LASSO-KNP</th>
<th>LASSO-VTE</th>
<th>LASSO-GSC</th>
<th>RIDGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara2</td>
<td>-1.76</td>
<td>-1.73</td>
<td>-1.82</td>
<td>-5.70</td>
<td>-4.81</td>
<td>-5.40</td>
<td>-6.73</td>
<td>-6.18</td>
<td>-6.18</td>
<td>-4.41</td>
</tr>
<tr>
<td>Bike</td>
<td>-6.05</td>
<td>-6.06</td>
<td>-6.08</td>
<td>-6.05</td>
<td>-5.58</td>
<td>-5.69</td>
<td>-6.90</td>
<td>-6.78</td>
<td>-6.92</td>
<td>-2.57</td>
</tr>
<tr>
<td>Roof</td>
<td>-0.87</td>
<td>-0.59</td>
<td>-0.76</td>
<td>-6.75</td>
<td>-5.71</td>
<td>-6.83</td>
<td>-7.59</td>
<td>-7.36</td>
<td>-7.45</td>
<td>-6.90</td>
</tr>
<tr>
<td>Houses</td>
<td>-2.67</td>
<td>-2.59</td>
<td>-2.56</td>
<td>-3.56</td>
<td>-3.00</td>
<td>-3.34</td>
<td>-4.48</td>
<td>-4.23</td>
<td>-4.40</td>
<td>-2.09</td>
</tr>
<tr>
<td>Average</td>
<td>-5.24</td>
<td>-5.22</td>
<td>-5.17</td>
<td>-9.81</td>
<td>-8.94</td>
<td>-9.41</td>
<td>-11.11</td>
<td>-10.78</td>
<td>-10.87</td>
<td>-7.80</td>
</tr>
</tbody>
</table>

**Fig. 12.** BDRATE results for test set 1 using HEVC-EN based on Elastic Net as function of parameter $\lambda_2$, with constant parameter $\lambda_1$ given by KNP with $k = 15$ non-zero predictors. HEVC-LASSO is used as reference method to compute BDRATE results.
approach. In addition to LLE, the TM algorithm, which is a particular case of LLE with a number of predictors equal to 1, has been evaluated. All these methods were implemented in the HEVC standard, by replacing the angular mode 3. Note that the proposed GOSP is the only method that generalizes intra directional prediction, and replaces all the angular modes of HEVC standard (excluding planar and DC modes). The configuration parameters used by these methods (e.g. TW size, number of predictors or maximum filter size) are the same as the ones used in [42].

Tables V, VI and VII present the BDRATE results of the proposed GOSP solution, as well as the comparing methods, all using the HEVC standard, for the compression of test images of sets 1, 2 and 3, respectively. BDRATE results were all computed relative to the performance of original HEVC implementations, using HM-14.0 software with intra main profile. The bitrate overhead given by GTs (5 bits) and the results given in Table VII led us to similar conclusions. In regard to the test images defined in set 3, the experimental results shown in Table VII were omitted because the BDRATE values are inferior to 1% for all images. In fact, also for some of the presented test images, the BDRATE results are not significant, as can been seen in Table VI. In the case of the LSP, TM, LLE and SLSP implementations, the small gains can be justified by the fact that HEVC angular intra modes are already performing efficiently. Regarding the proposed GOSP method, we observe that even replacing the angular modes of HEVC, its RD performance tends to be similar or better than the one of the HEVC standard. In many cases GOSP provides significant RD gains over other methods, as for instance, the Johnny and BasketballDrive images. There are several images for which SLSP presents a slight advantage over GOSP, but the advantage is always inferior to 1%. The justification for such a behaviour can be related with the fact that GOSP uses more bitrate to signal the prediction modes. Although sparse predictors are able to provide better prediction results, the bitrate overhead given by GTs (5 bits) and the maximum filter size $D_f$ (1 bit) is superior to the one of the compared HEVC-based methods, which use at most 5 bits to signal the angular modes including the added linear prediction technique, such as SLSP.

In regard to the test images defined in set 3, the experimental results given in Table VII led us to similar conclusions. The proposed prediction method shows to be more efficient than directional prediction for most images, providing bitrate savings up to 8.3%. However, one also observes that the LLE method [16] provides superior coding gains in particular for Microsoft-P10 and Microsoft-P26 images. Such advantage of LLE is due to the fact that this method is able to use more distant predictors than GOSP, using a larger causal window for predictor selection. Since these images present some strong correlations between samples at larger distances than the context filter size of GOSP method, they are better predicted by other methods able to exploit these correlations.

D. Statistical analysis

In order to demonstrate the advantage of the proposed method for image prediction, we analyse the blocks that are predicted using adaptive sparse predictors. Figure 13 illustrates the Barbara image highlighting the PUs predicted by adaptive
sparse models \( (D_f = 31) \), when encoded using the proposed HEVC and GOSP algorithm for \( QP=32 \). Non-highlighted regions correspond to PUs predicted either by planar, DC or first-order linear filters \( (D_f = 1) \). The highlighted regions clearly show that adaptive sparse models tend to be used in more complex textured regions with repeated patterns, namely the Barbara clothing, the table’s towel, as well as some edges present in the scene. These results demonstrate the advantage of sparse models when compared to the first-order linear filters for the compression of these regions.

In regard to the mode usage statistics for this experimental test (Barbara image and \( QP=32 \)), we observed about 11% of PUs predicted using planar mode, 5% using DC, 22% using first order filters and 62% using adaptive sparse filters. A detailed representation of the usage statistics of linear prediction modes is shown in Figure 14. In this plot, the bright and dark bars represent the frequency of occurrence of first order linear filters and adaptive sparse predictors, respectively, for each available GT and filter configuration (see Figures 3 and 5). The correspondence between the used mode signalling indexes, from 2 to 34, and the selected GT and filter configuration, is the one previously illustrated in Table I, with \( d \) values represented in Tables II and III.

The results of Figure 14 show that almost all GTs (modes 2 to 34) are used in the prediction process, for both the first order and the adaptive sparse models. In the case of the sparse models (dark bars), one observes that mode usage distribution has increased variance, with larger difference between the lowest and highest usage rate value. The 10, 18 and 26 modes, which correspond to the three filter configurations of Figure 5 without using GT (i.e. GT value \( d = 0 \)), are some of the most used prediction modes. This shows that in some cases the estimation procedure provides efficient sparse models for image prediction, not requiring additional GT, specifically in the presence of stationary image statistics. However, the use of other modes which use GTs with non-zero \( d \) value (mode different than 10, 18 and 26) is still significant, as can be apprehended from Figure 14. For some specific modes, such as 15, 16, 17, 19, 20, 21 and 27, the use of GTs over PUs predicted by adaptive sparse models is more frequent than PUs predicted by first order filters. These results demonstrate the effectiveness of GTs in improving the linear predicted block generated by adaptive sparse predictors in many situations.

### E. Computational complexity results

Regarding the computational complexity, the proposed method, inevitably, is much more complex than HEVC, because it needs to solve the Lasso problem and apply GTs. Although we did not focus this work on the research of computationally efficient implementations for the proposed method, we performed a detailed evaluation of the encoder and decoder running times (in seconds), as presented in Table VIII. These results are compared to the original HEVC running times, and related by the ratio columns of Table VIII.

The increase of the computational complexity of the encoders and decoders using GOSP are related to the usage rate of the proposed sparse models. On the encoder side, the sparse models are evaluated at all available blocks, and thus, the running times have a close relation with the image resolution. On the decoder side, the sparse models are only used in the blocks predicted by sparse models. The number of blocks predicted by these modes mainly depends on the image resolution and on the presence of complex textured areas with redundant structures that can be efficiently exploited by sparse models.

Despite its higher computational complexity, we believe that the proposed method could be largely improved using...
a more efficient algorithmic design as well as sub-optimal coding approaches. For instance, in smooth regions, sparse predictors usually are not important and thus they do not need to be evaluated in the encoder. Also, to avoid the complexity of adaptive model estimation methods, offline model estimation approaches could be investigated in the future to reduce the computational complexity of the model estimation algorithms.

VII. CONCLUSIONS

In this paper we developed a generalization of the intra prediction framework that unifies the angular HEVC prediction modes and least-squares prediction. It is based on sparse linear predictors followed by GTs. Upon this generalization, we proposed a new intra prediction framework to be incorporated into HEVC. We have investigated the performance of several sparse decomposition algorithms, such as OMP, Lasso, LAR and EN. The results obtained shed an interesting light over the intra prediction process, allowing a better understanding of the strengths and limitations of both angular HEVC-like and least-squares prediction from an image coding points of view, and show that sparsity is the key to unify them.

The use of LAR and LASSO methods for image prediction provide more efficient solutions than the greedy OMP method resulting, consequently, in state-of-the-art compression results when incorporated in HEVC. An interesting extension of this work, would be to apply the proposed method for inter prediction by augmenting the filter context to the previous encoded frames. Such an approach might provide an implicit solution for motion compensation, with additional ability to compensate non-rigid motion deformations using GTs.

REFERENCES

Luis F. R. Lucas (M’11) was born in Portugal in 1988. He received the Engineering and M.Sc. degrees in Electrical Engineering from Escola Superior de Tecnologia e Gestão de Instituto Politécnico de Leiria, Portugal, in 2000, and the Ph.D. degree from the Universidade Federal do Rio de Janeiro, Brazil, in 2016, in collaboration with the Instituto de Telecomunicações, Portugal.

Since 2009, he has been a researcher at Instituto de Telecomunicações, Portugal. His research interests include digital signal and image processing, namely image and video compression.

Nuno M. M. Rodrigues (M’99) graduated in electrical engineering in 1997, received the M.Sc. degree from the Universidade de Coimbra, Coimbra, Portugal, in 2000, and the Ph.D. degree from the Universidade Federal do Rio de Janeiro, Brazil, in 2009, in collaboration with the Universidade Federal do Rio de Janeiro, Brazil.

Since 1997, he has been with the Department of Electrical Engineering, Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria, Portugal. Since 2009, he has been with the Instituto de Telecomunicações, Portugal, where he is currently a Senior Researcher. He has coordinated and is currently involved as a researcher in various national and international funded projects. His current research interests include digital signal and image processing, namely, image and video compression, multiview and light field image and video compression, medical image compression and many-core programming.

Carla L. Pagliari (M’90, SM’06) received the Ph.D. degree in Electronic Systems Engineering from the University of Essex, Colchester, U.K., in 2000. She was with TV Globo, Rio de Janeiro, Brazil, from 1983 to 1985. From 1986 to 1992, she was a Researcher with the Brazilian Army Research and Development Institute, Rio de Janeiro, Brazil. Since 1993, she has been with the Department of Electrical Engineering, at the Military Institute of Engineering, Rio de Janeiro, Brazil.

She has experience in the area of Television Engineering and Image Processing, acting on the following topics: image/video coding, stereo/multi-view video coding, stereo systems, image/video synthesis, video streaming, and digital TV. She participated in the definition of the Brazilian Digital Television System and is a member of the Educational Board of the Brazilian Society of Television Engineering. She has projects in image processing for defence applications, image/video fusion of the infrared and visible-light signals, video over IP for applications in public safety, and digital TV. She acts as an associate editor of the Multidimensional Systems and Signal Processing (Springer) journal.

Sérgio M. M. de Faria (M’93, SM’08) was born in Portugal in 1965. He received the Engineering degree in Electronic Engineering from Universidade de Coimbra, Portugal, in 1988, the M.Sc. degree in Electrical Engineering from Universidade de Coimbra, Portugal, in 1992, and the Ph.D. degree in Electronics and Telecommunications from the University of Essex, England, in 1996.

He is Professor with the Department of Electrical Engineering, in Escola Superior de Tecnologia e Gestão of Instituto Politécnico de Leiria, Portugal, since 1990. He has collaborated in master courses with Faculty of Science and Technology and with Faculty of Economy of Universidade de Coimbra, Portugal. He is an Auditor with A3ES organization for Electrical and Electronic Engineering courses in Portugal. He is a Senior Researcher with Instituto de Telecomunicações, Portugal.

His research interests include 2D/3D image and video processing and coding, motion representation, and medical imaging. In this field, he has published edited 2 books and authored 8 book chapters, and over 130 peer reviewed papers. He has been participating and he is responsible for several, national and international (EU), funded projects.

He is an Area Editor of Signal Processing: Image Communication. He has been a Scientific and Program Committee Member of many international conferences. He is a reviewer for several international scientific journals and conferences (IEEE, IET and EURASIP). He is a Senior Member of the IEEE.