Sparse Coding based Frequency Adaptive Loop Filtering for Video Coding

Jens Schneider
Institut für Nachrichtentechnik
RWTH Aachen University, Germany
schneider@ient.rwth-aachen.de

Max Bläser
Institut für Nachrichtentechnik
RWTH Aachen University, Germany
blaeser@ient.rwth-aachen.de

Mathias Wien
Institut für Nachrichtentechnik
RWTH Aachen University, Germany
wien@ient.rwth-aachen.de

ABSTRACT

In-loop filtering is an important task in video coding, as it refines both the reconstructed signal for display and the pictures used for inter-prediction. In order to remove coding artifacts, machine learning based methods are assumed to be beneficial, as they utilize some prior knowledge on the characteristics of raw images. In this contribution, a dictionary learning / sparse coding based in-loop filter and a frequency adaptation model based on the l_p-ball-energy in the spectral domain is proposed. Thereby the dictionary is trained on raw data and the algorithms are controlled mainly by the parameter for the sparsity. The frequency adaption model results in further improvement of the sparse coding based loop filter. Experimental results show that the proposed method results in coding gains up to -4.6 % at peak and -1.74 % on average against HEVC in a Random Access coding configuration.

CCS CONCEPTS

• **Information systems** → *Multimedia information systems*;

KEYWORDS

Loop Filter, Frequency Adaption, Sparse Coding

ACM Reference Format:
Jens Schneider, Max Bläser, and Mathias Wien. 2018. Sparse Coding based Frequency Adaptive Loop Filtering for Video Coding. In *Proceedings of Packet Video'18*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3210424.3210427

1 INTRODUCTION

Loop filtering is a crucial step in state of the art video coding as it affects both the perceived quality of the reconstructed picture and the quality of pictures used for inter-prediction. Therefore, in the latest video coding standard High Efficiency Video Coding (HEVC) [3], a deblocking filter [11] and Sample Adaptive Offset (SAO) [7] are specified, leading to both visual improvements and improved objective coding performance. Moreover, adaptive loop filtering based on Wiener-filtering was investigated during the standardization phase of HEVC and is again part of the Joint Exploration Model (JEM) [4]. In-loop filtering can also be understood as an inverse image restoration problem aiming at removing the coding noise. Recently, Convolutional Neural Networks (CNN) based image processing evolved to the state of the art for many applications. Therefore, training CNNs for the purpose of in-loop filtering was suggested in [5] and [12], showing significant objective coding gains. However, the CNN models in these methods need to be trained according to certain rates as the model learns the relationship between the reconstructed images obtained by the coded representation and the original images. This directly includes coded data into the training process and results in several models for different rates.

Another promising approach for solving inverse problems is given by the Sparse-Land model including the ideas of dictionary learning and sparse coding [6]. It was shown that the sparsity of images represented in trained dictionaries is a useful property for different applications such as super-resolution and denoising. In consequence, a sparse coding based deblocking approach for JPEG was proposed in [8] and more recently a soft decoding approach was proposed [9]. However, these methods are compared to JPEG in the context of still image coding and no results regarding video coding are provided.

A sparse coding based frequency adaptive in-loop filtering scheme is proposed in this paper. The proposed method utilizes a trained dictionary for the removal of coding noise, which is controlled by signaling additional frequency adaptation parameters in the slice segment header of NAL units, as specified in HEVC. The implementation was based on the SPArse Modeling Software [10] and the HEVC reference software HM version 16.9.

The remainder of the paper is organized as follows: Section 2 shortly reviews the theory behind sparse image representation with a trained dictionary. In section 3 a comparison between the influence of coding noise and Gaussian noise on sparse coding is presented. Additionally, our method of Sparse Coding based frequency adaptive In-Loop Filtering (SCALF) is detailed. In section 4 the simulation setup and results are presented and section 5 concludes the paper.

2 SPARSE IMAGE REPRESENTATION

The concept of sparse image representation relies on the transformation from pixel domain to a domain where the signal is assumed to be sparse. This transformation is often chosen to be overcomplete in order to introduce even more sparsity. Moreover, the transformation acts on image patches so that a single patch is represented as

$$\boldsymbol{x} \approx \boldsymbol{D}\boldsymbol{\alpha} \tag{1}$$

Packet Video'18, June 12–15, 2018, Amsterdam, Netherlands

ACM ISBN 978-1-4503-5773-9/18/06…$15.00
https://doi.org/10.1145/3210424.3210427

with $\boldsymbol{x}$ representing the vectorized image patch, $\boldsymbol{D}$ representing the transform basis and $\boldsymbol{\alpha}$ representing the sparse coefficient vector. The transform basis $\boldsymbol{D}$ is also referred to as dictionary and its columns are referred to as atoms. Generally, it is possible to choose a dictionary for the representation such as the discrete cosine transform (DCT) basis functions or to train a dictionary as in the case of Karhunen-Loève (KLT) basis functions. However, these bases will not allow for a sparse representation of the signal, as they are not overcomplete and not optimized for sparsity. To overcome these issues it is beneficial to train a dictionary according to the Sparse-Land model [6]:

$$\boldsymbol{D} = \arg\min_{\boldsymbol{D}} \sum_{i=1}^{n} \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \,. \tag{2}$$

Usually n raw image patches extracted from a training set are used as training data $\boldsymbol{x}_i$. These patches are often preprocessed by mean subtraction (centering) and normalization to unit variance. Note that the penalty according to the l_1-norm introduces the sparsity to the minimization problem and the parameter λ has to be set in advance to control the influence of this penalty. Once the dictionary is trained, a sparse representation for an image patch $\boldsymbol{x}$ can be found solving

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \,. \tag{3}$$

Equation (3) can be solved with e.g. the LARS algorithm [6]. However, the obtained sparsity can vary for different patches as the formulation only penalizes the sparsity and does not constrain it.

In general, there is no need to find the sparse representation using (3). As the sparse representation should be a tradeoff between a small l_2-norm error between the representation and the original and a low number of nonzero coefficients, it is also possible to find a sparse representation solving the constrained minimization problem

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\alpha}\|_0 = L \tag{4}$$

with L representing the number of non zero coefficients in $\boldsymbol{\alpha}$. The solution of (4) can be obtained using the Orthogonal Matching Pursuit (OMP) algorithm and the sparsity of $\boldsymbol{\alpha}$ will be L for every patch. When the sparse representation is found, the image patches can be reconstructed according to (1) and the combination of the patches results in the reconstructed image.

3 FREQUENCY ADAPTIVE LOOP FILTERING

3.1 Sparse Coding based In-Loop Filtering

The general scheme of sparse coding based in-loop filtering follows the steps shown to be effective in the case of removing white Gaussian noise in [6]:

- Extract overlapping patches $\boldsymbol{x}_{\text{rec}}^{\text{HEVC}}$ from the image to be filtered $\boldsymbol{I}_{\text{rec}}^{\text{HEVC}}$ and center these patches,
- choose a suitable parameter L and find sparse codes $\boldsymbol{\alpha}$ in the dictionary $\boldsymbol{D}$ for all extracted patches $\boldsymbol{x}_{\text{rec}}^{\text{HEVC}}$ using (4),
- reconstruct the patches calculating $\boldsymbol{x}_{\text{rec}}^{\text{SC}} \approx \boldsymbol{D}\boldsymbol{\alpha}$,
- combine the reconstructed patches $\boldsymbol{x}_{\text{rec}}^{\text{SC}}$ back to an image $\boldsymbol{I}_{\text{rec}}^{\text{SC}}$ via averaging in overlapping areas.

This procedure introduces the prior knowledge about the characteristics of image signals to the result of the in-loop filtering process, since the dictionary $\boldsymbol{D}$ was trained with raw image data. As reported in our previous work [13], introducing more sparsity at lower rates is beneficial in the case of video coding, therefore this approach is also followed in this paper. From the perspective of the procedure described above, it is not easy to see why a higher level of sparsity should result in better denoising performance at lower rates. In order to motivate this assumption, consider the following minimization problem as a variation of (4):

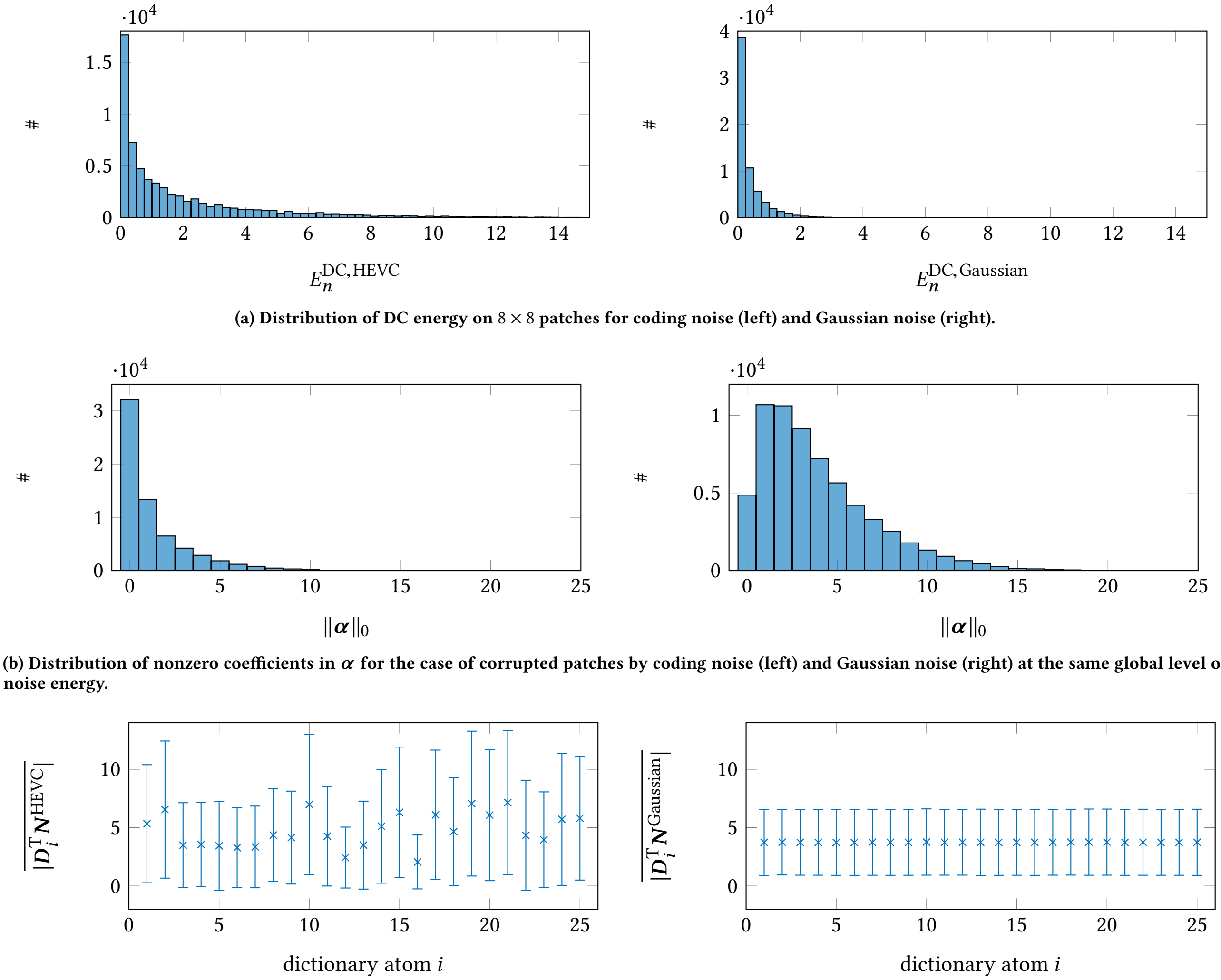
$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 < \varepsilon \,. \tag{5}$$

We obtain the sparsest solution for $\boldsymbol{\alpha}$ with this formulation such that the squared l_2-error between the sparse representation and the original signal becomes smaller than a threshold $\varepsilon > 0$. Obviously, it is reasonable to set the threshold ε close to the energy of the noise in a patch. Smaller values of ε would lead to a representation of the noise itself in the dictionary whereas for larger values of ε the result will be less accurate in terms of l_2-error. Since OMP only adds one atom of the dictionary to the support of $\boldsymbol{\alpha}$ in every iteration and stops when the l_2-error becomes smaller than ε, the support will be smaller when the stopping criterion is met after a lower number of iterations. Exactly this happens when ε is larger which is the case for higher energies of the noise, i.e. at lower bitrates. However, there is no guarantee for this technique of denoising to work in case of coding noise.

Regarding sparse coding there are some major differences between coding noise and Gaussian noise, which are developed in the following for one example sequence corrupted by intra coding noise. For all results shown in this section, a dictionary with $K = 512$ atoms trained with centered and normalized raw image patches was used. The training was performed according to (2) with $\lambda = 0.15$. For the comparison of the impact of Gaussian noise and coding noise on the sparse coding, zero mean Gaussian noise was generated such that it meets the same global energy of the coding noise for the first frame of the Traffic sequence at QP = 37. This results in a noise variance of $\sigma^2 = 22$ which is rather a low noise energy with respect to a common variance of $\sigma^2 = 400$ used e.g. in [6].

Difference in the local DC energy

Different from Gaussian noise, coding noise has a higher energy in the local mean of the noise. Generally, the mean of a patch cannot be predicted by sparse coding, as the input patches and the dictionary atoms are centered, i.e. they have zero mean. Therefore, there is hardly any chance to remove the DC noise in both cases. Figure 1a shows the histogram of the DC energy for 8 × 8 patches extracted from the image of coding noise and the image of Gaussian noise at the same global energy. It can be observed that the distribution of the DC energy is wider for the coding noise, meaning that there is more unremovable energy than in the case of Gaussian noise. Ideally, these histograms should show a peak at zero DC energy, so that there is no part of the noise which cannot be removed. However, the noise in realistic scenarios does not have locally zero mean but the distribution for the Gaussian noise is closer to the ideal peak than the distribution of the coding noise. Therefore, it can be assumed that removing Gaussian noise with sparse coding based denoising methods has a higher chance of success than removing



(a) Distribution of DC energy on 8×8 patches for coding noise (left) and Gaussian noise (right).

(b) Distribution of nonzero coefficients in $\boldsymbol{\alpha}$ for the case of corrupted patches by coding noise (left) and Gaussian noise (right) at the same global level of noise energy.

(c) Average and standard deviation (bars) of the absolute correlation between the noise and the first 25 atoms of the dictionary $\boldsymbol{D}$. The Matrix $\boldsymbol{N}$ denotes the vectorized noise patches. The statistics are shown for coding noise (left) and Gaussian noise (right).

Figure 1: Visualization of the different characteristics of coding noise and Gaussian noise with respect to sparse coding. The coding noise patches were extracted from the difference of the first frame of the raw and the coded $(\mathrm{QP} = 37)$ Traffic sequence. The noise has approximately the same energy at the global level.

coding noise. An integral measure, which can be derived from these histograms, is the average DC energy of the noise per 8×8 patch $\overline{E}_n^{8\times8}$. The values corresponding to the histograms shown in Figure 1a are $\overline{E}_n^{\text{DC},8\times8,\text{HEVC}} = 2.03$ and $\overline{E}_n^{\text{DC},8\times8,\text{Gaussian}} = 0.35$, again showing that the DC energy is lower for Gaussian noise than for coding noise.

Difference in the local energy

The property of different distributions of the energy cannot only be observed for the local DC of the noise. There is also a difference in the energy distribution of the centered noisy patches itself leading to a different behavior of the sparse coding. Solving (5) with OMP, it can be seen that OMP will not find any atom for the sparse representation and return an empty support, if $\|\boldsymbol{x}\|_2^2 < \varepsilon$. This would directly lead to a reconstructed patch only containing the average value of the input patch but no AC component, as $\boldsymbol{D\alpha} = \boldsymbol{0}$ in that case. Obviously, this is not desired, as the raw image patch most probably possesses some image-like characteristics and was not constant over the whole patch. However, with the formulation in (4) it is ensured that OMP will find L atoms for the support of $\boldsymbol{\alpha}$. From the considerations above it can be concluded that none of these atoms is reliable with respect to the noise level. The number of nonzero coefficients for 8×8 patches extracted from images corrupted with coding noise (left) and Gaussian (right) is depicted in Figure 1b. The sparse codes were calculated using (5) with $\varepsilon = N\sigma^2$ and $N = 64$. In other words, the stopping criterion for OMP was set to the expected value of the noise energy in the patch. From the displayed histograms it becomes clear, that OMP tends to find more empty supports for patches corrupted with coding noise than with Gaussian noise. Evaluating the sparsity by an integral measure of the average supports gives according to Figure 1b $\overline{\|\boldsymbol{\alpha}\|_0} = 1.3$ (left) and $\overline{\|\boldsymbol{\alpha}\|_0} = 3.9$ (right), respectively.

Difference in the correlation with the dictionary

A third difference between coding noise and Gaussian noise with respect to sparse coding lies in the correlation of the noise with the atoms of the dictionary. Conceptually, this correlation leads to errors in the support found by OMP. From the mathematical point of view, errors in the support refer to the error which is observable if a signal with known sparse codes is encoded with OMP. The conditions for exact recovery of the OMP algorithm are addressed e.g. in [14]. However, from the signal processing point of view it is often sufficient to ask, whether OMP can find the same support with an input of noisy image patches as it would find with an input of raw image patches. It can be directly concluded from the results in Figure 1b that this is generally not the case, since otherwise the distributions would be the same. A condition for the recovery of the same support for raw and noisy images is, that the coding noise in any patch must not be correlated with any atom of the dictionary. Put differently, the distribution of the correlation between all the extracted patches and any dictionary atom should show a unit pulse at 0. Obviously, this cannot be the case, due to the following reasons: The dictionary should cover the entire space $\mathbb{R}^N$, in order to achieve accurate representations of natural images, since images cannot be represented lossless in lower dimensions than they are already represented in. If that is the case, the noise cannot be orthogonal to all the atoms, because there is no orthogonal space left any more. However, it can be assumed that the closer the distribution of the correlations between noise and dictionary atoms is to the unit pulse, the better OMP will recover the support. What does "closer" mean in this context? The distribution of the unit pulse has zero mean and zero variance. Therefore, distributions with small mean and small variance can be considered to be close to the unit pulse distribution. Figure 1c shows the average and the standard deviation of the absolute correlation between the noise and the first 25 atoms of the dictionary $\boldsymbol{D}$. In fact, the dictionary had $K = 512$ atoms in this experiment, but for the visualization it would not make sense to depict the correlations for all the 512 atoms. The absolute value of the correlation was used for these statistics, since locally negative correlation between the noise and the atoms influences the support in the same manner as positive correlation does. From the figure it can be observed that the average absolute correlations and its standard deviation is higher for the coding noise than for the Gaussian noise in most cases. This means that the distribution of the correlation between Gaussian noise and dictionary is closer to the unit pulse distribution than the distribution for the coding noise. Again, it makes sense to find some global measure for this observation. Calculating the mean of the measures depicted in Figure 1c we get pairs $(\mu, \overline{\sigma})$ describing the mean absolute correlation of all the dictionary atoms with the noise and its standard deviation. For the results shown, those values are $\left(\mu^{\text{HEVC}}, \overline{\sigma}^{\text{HEVC}}\right) = (5.6,\ 5.1)$ and $\left(\mu^{\text{gaussian}}, \overline{\sigma}^{\text{gaussian}}\right) = (3.7,\ 2.8)$ indicating that the distribution of the correlation between the dictionary atoms and the Gaussian noise is globally more promising for the recovery of the correct support. Moreover, the statistics for the Gaussian noise show that there is almost no variation with respect to the different dictionary atoms. This can be also be interpreted such that the OMP algorithm stays globally unbiased for an input of image patches containing additional Gaussian noise. Intuitively, the observations from Figure 1c can also be explained by looking at the characteristics of the coding noise from a more high level perspective. Since the coding noise arises from the quantization of the DCT-coefficients of the residual signal, its characteristics are closer to natural images than the characteristics of Gaussian noise. Therefore, it is reasonable that the coding noise is more correlated with dictionary atoms which are designed for the sparse representation of natural image patches. Thus, coding noise can be considered to be more harmful for sparse coding applications than Gaussian noise.

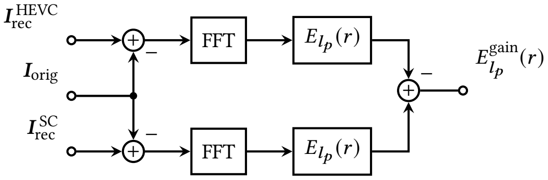

$I_{\text{rec}}^{\text{HEVC}}$
I_{orig}
$I_{\text{rec}}^{\text{SC}}$
+
−
FFT
$E_{l_p}(r)$
$E_{l_p}^{\text{gain}}(r)$

Figure 2: Analysis of l_p-ball energies of the both the coding error obtained by HEVC and sparse coding based in-loop filtering

To conclude this section, there are at least three reasons for a worse performance of sparse coding based denoising for coding noise removal: The DC energy of the coding noise is higher, the coding noise is not distributed uniformly over the patches, and the correlations between the noise and the dictionary atoms is higher for coding noise than for Gaussian noise. Nevertheless, the question whether sparse coding based denoising can be used as in-loop filter is answered in section 4. In order to further improve the filtering step, we designed a method for frequency adaption of the filtering process, which is detailed in the following.

3.2 l_p-Ball-Energy Analysis

The performance of the sparse coding based in-loop filter can be analyzed by the error signal between the raw image $\boldsymbol{I}_{\text{orig}}$ and the HEVC reconstruction $\boldsymbol{I}_{\text{rec}}^{\text{HEVC}}$ and the sparse coding reconstruction $\boldsymbol{I}_{\text{rec}}^{\text{SC}}$ respectively. The analysis is performed in the spectral domain of the error signals so that the frequencies at which the sparse coding approach is beneficial become visible. For this, the measure of the l_p-ball-energy $E_{l_p}(r)$ of a 2-dimensional signal is introduced. Conceptually, this measure represents the energy of the signal along the boundary of an l_p-ball with the generalized radius r in the frequency domain. In fact, for the cases with $p \neq 2$ the radius does not describe the distance of every point on the ball from its center but still leads to a valid parametrization of the ball. Thus, the term for the energy reads:

$$E_{l_p}(r) = \sum_{(f_1,f_2)\in l_p\text{-ball}(r)} |\boldsymbol{S}(f_1, f_2)|^2 . \tag{6}$$

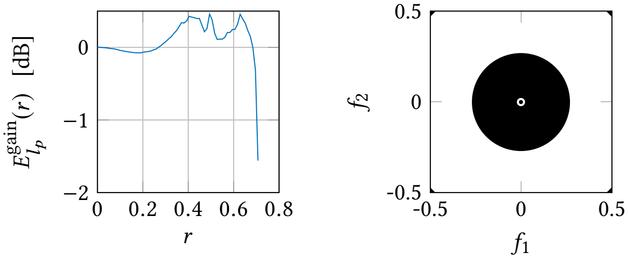


Figure 3: Visualization of $E_{l_p}^{\text{gain}}(r)$ (left) and the corresponding filter mask M (right) for an l_2-ball.

The parametrization is given by

$$r = \left(|f_1|^p + |f_2|^p\right)^{\frac{1}{p}} \ . \tag{7}$$

Note that in the cases $p < \infty$ the radius can be greater than $r = 0.5$ and parts of the ball's boundary still lie in the baseband of the spectrum. In these cases the energy along the boundary is calculated in the baseband only and the periodic copies of the spectrum are neglected. Generally, $r \in \left(0, 2^{\frac{1}{p}-1}\right)$ defines the range of r for which at least parts of the boundary lie in the baseband of the spectrum. Generally speaking, the parameter r parametrizes the frequency-axes and $E_{l_p}(r)$ measures energy at positions r of this parametrization.

Figure 2 shows a block diagram of the analysis in the l_p-ball-energy domain. The output $E_{l_p}^{\text{gain}}(r)$ indicates at which radius r which reconstructed signal is closer to the original in terms of error energy. This scheme can be utilized at the encoder side in order to decide at which radius r the HEVC reconstruction should be used and where it is beneficial to use the sparse coding in-loop filtering. Therefore, the output of the analysis is a perfect filter in the FFT-domain indicating which reconstruction signal should be used. This filter can also be interpreted as a binary mask $\boldsymbol{M}(r)$ combining the spectra of the HEVC reconstruction and the reconstruction obtained by the sparse coding based in-loop filter. The measure $E_{l_p}^{\text{gain}}(r)$ and the resulting filter mask $\boldsymbol{M}(r)$ in the FFT-domain are visualized in Fig. 3. The mask is obtained, such that $\boldsymbol{M}\left(E_{l_p}^{\text{gain}}(r) > 0\right) = 1$ and $\boldsymbol{M}\left(E_{l_p}^{\text{gain}}(r) \leq 0\right) = 0$. Consequently, the spectra of the both the reconstructions $\boldsymbol{S}_{\text{rec}}^{\text{HEVC}}$ and $\boldsymbol{S}_{\text{rec}}^{\text{SC}}$ can be combined according to the mask:

$$\boldsymbol{S}_{\text{rec}}^{\text{SCALF}} = \boldsymbol{M} \circ \boldsymbol{S}_{\text{rec}}^{\text{SC}} + \hat{\boldsymbol{M}} \circ \boldsymbol{S}_{\text{rec}}^{\text{HEVC}} \tag{8}$$

where $\circ$ denotes the Hadamard-product and $\hat{\cdot}$ denotes the logical complement. The inverse Fourier transform of $\boldsymbol{S}_{\text{rec}}^{\text{SCALF}}$ yields the final reconstruction result $\boldsymbol{I}_{\text{rec}}^{\text{SCALF}}$.

3.3 Signaling of Parameters

Since the decoder is not able to calculate the l_p-ball-energy of the error signals between the raw image $\boldsymbol{I}_{\text{orig}}$ and the reconstructions $\boldsymbol{I}_{\text{rec}}^{\text{HEVC}}$ and $\boldsymbol{I}_{\text{rec}}^{\text{SC}}$, some parameters have to be coded into the bitstream. For this proposal the following parameters

- SCALFenabled flag,
- ShapeIdx indicating the parameter $p \in \{1, 2, \infty\}$ for the shape of the l_p-ball,
- NumRadiusIdc indicating the total number of indices to the radius vector,
- RadiusIdc containing the indices to the radius vector,
- MaskStartVal holding the value of the filter mask at $r = 0$

are signalled in the slice segment header. All parameters are binarized with fixed length binary codes. The influence of the coded parameters on the bitrate is assumed to be low, as they are transmitted only once per picture.

In order to find a suitable set of parameters, the encoder tests whether the sparse coding based filter is beneficial for a picture and signals the SCALFenabled flag, if the filter should be applied. Moreover, it is tested which l_p-ball is most beneficial for the frequency separation and the index ShapeIdx to the parameter p is signaled. Thereby only the cases for $p \in \{1, 2, \infty\}$ are considered in oder to reduce the complexity of the encoder. The radius r of the ball is quantized into 64 bins and the indices RadiusIdc to the values where the filter mask $\boldsymbol{M}$ changes its value are signaled. The value MaskStartVal of the filter at $r = 0$ is also transmitted. In other words, the vector of indices specifying at which radius r the binary filter mask $\boldsymbol{M}$ changes its value is coded with a run-length code.

4 SIMULATION SETUP AND RESULTS

4.1 Simulation Setup

In the following, the parameters used for SCALF and the simulation setup for the video coding are detailed. For all results reported in this paper, a dictionary with $K = 512$ atoms was used. The patch size was chosen to be $s_{\text{p}} = 8 \times 8$ pixels, overlapping by 7 pixels for both training and testing. The dictionary was trained offline with 91 training images commonly used for training [15]. All the patches were preprocessed by mean subtraction and normalization to unit variance for training purposes. The training was performed according to (2) with parameter $\lambda = 0.15$. Note that no coded representation of the patches was used for training.

For the sparse coding in the encoder and the decoder (4) was used. The parameter L was chosen depended on the bitrate as mentioned above. We chose to set the number of nonzero coefficients L according to the Quantization Parameter (QP) such that $L = -\text{QP} + 42$. This linear relationship was found to be appropriate for the application in a preanalysis. Note that it is possible to test for several parameters L at the encoder side and signal the best suitable for every picture. However, this procedure would result in a further enlarged complexity of the encoder, as the sparse coding has to be performed multiple times in this case. For testing, patches were only centered, as normalization does not influence the result OMP delivers [6].

SCALF is integrated into the reference software HM-16.9 right after the deblocking filter is applied and before the SAO is invoked. The reason for this order of application is that SAO still has the ability to refine the output of SCALF in areas where it fails. The investigated rate points were chosen to be at QP $\in \{22, 27, 32, 37\}$ which is according to the common testing conditions for HEVC [2]. A comparison between three codecs was performed, namely HEVC, HEVC including SCLF (without l_p-ball frequency adaption) and

Table 1: BD-rate savings against HM-16.9 for different coding configurations and different in-loop filters.

	AI		RA	
seq.	SCLF	SCALF	SCLF	SCALF
BQTerrace	−0.15 %	−0.24 %	−0.56 %	−0.85 %
BasketballDrive	−0.45 %	−0.43 %	−0.79 %	−0.88 %
Cactus	−1.16 %	−1.2 %	−0.96 %	−1.23 %
Kimono	−0.85 %	−0.86 %	−0.81 %	−0.97 %
ParkScene	−1.43 %	−1.5 %	−0.25 %	−0.44 %
PeopleOnStreet	−2.78 %	−2.86 %	−4.23 %	−4.6 %
Traffic	−1.84 %	−1.91 %	−2.37 %	−3.2 %
AVG	−1.24 %	−1.29 %	−1.42 %	−1.74 %

HEVC including SCALF with l_p-ball frequency adaption. For the case of SCLF only the SCALFenabled flag is transmitted and the filter is operated according to the description in Section 3.1.

4.2 Results

Table 1 shows the coding results in terms of BD-statistics [1] for an All-Intra (AI) and a Random Access (RA) coding configuration. The gains are reported in terms of the average rate savings, as these values are better interpretable. For the AI coding configuration the average rate savings are −1.24 % on average for the SCLF and −1.29 % for SCALF including the frequency adaption. Consequently, the benefit of the frequency adaption is limited in case of an AI coding scenario. Note that for the sequence BasketballDrive the rate savings for SCALF are even worse than for the SCLF. This can be related to the fact, that the filtering is performed before the SAO is operated. Therefore, the SAO might perform worse, when the frequency adaption is applied resulting in lower gains. For the RA coding configuration the average rate savings are −1.42 % for SCLF and −1.74 % for SCALF respectively. The additional coding gain provided by the frequency adaption is in the range of −0.3 %. Note that for the sequence Traffic, the additional gain is even −0.8 %. Therefore, it can be concluded that the frequency adaption model has a significant impact on the coding performance for a RA coding scenario. Generally, it can be observed that the rate savings are higher in case of RA coding and that the improvements introduced by the frequency adaption are also higher for the RA case. The explanation of the first observation is that the SCALF improves the reconstructed signal and possible prediction signals, if it is applied in a configuration using inter prediction. Therefore, it can also lead to rate reduction due to lower residual coding cost. The ability to save rate is limited in the AI scenario, since only the signaling of SAO parameters is influenced by the sparse coding based loop-filters. Reducing the costs for residual coding is not possible for AI. In summary, the results show that the coding performance of HEVC can be improved when sparse coding based loop-filters are applied. Additionally, the proposed model for frequency adaption results in further coding gains for RA coding.

5 CONCLUSION

In this paper a sparse coding based in-loop filter for video coding is presented. Moreover, a model for frequency adaption is shown to be beneficial for this application. Thereby the proposed method of frequency adaption is generalized such that it can be used for the analysis and improvement of other machine learning based in-loop filtering techniques, since the l_p-ball-energy measure provides clues on the frequency range a loop filter performs well. The proposed in-loop filters clearly results in improvements over HEVC and the frequency adaption model leads to additional gains in RA coding scenarios. A major benefit with respect to CNN based in-loop filtering methods is that no training on coded data has to be performed and there is no need to train a different model for different rates. Additionally, the training of a dictionary is less complex than the training of a CNN and no GPU acceleration is needed. Nevertheless there is still potential for improvements in the proposed method. For example, the design of the binary filtering mask can be improved further so that it is not a mask but a filter with continuous frequency response or an optimum parameter L indicating the sparsity for a every picture can be coded into the bitstream.

In conclusion, this paper shows that there is still room for improvement in the in-loop filter of state of the art video coding standards.

REFERENCES

[1] Gisle Bjontegaard. 2001. *Calculation of average PSNR differences between RD-curves.* Technical Report Doc. VCEG-M33. ITU-T SG16/Q6 VCEG, Austin, USA.

[2] Frank Bossen. 2013. *Common test conditions and software reference configurations.* Technical Report. JCT-VC, 12th Meeting, Geneva.

[3] B. Bross, W. Han, J. Ohm, G. Sullivan, Y. Wang, and T. Wiegand. 2013. High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS and Last Call). JCTVC-L1003. (Jan. 2013).

[4] Jianle Chen, Elena Alshina, Gary J. Sullivan, Jens-Rainer Ohm, and Jill Boyce. 2016. *Algorithm Description of Joint Exploration Test Model 3.* Technical Report. JVET, 3rd Meeting, Geneva.

[5] Yuanying Dai, Dong Liu, and Feng Wu. 2016. A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding. *CoRR* abs/1608.06690 (2016). arXiv:1608.06690 http://arxiv.org/abs/1608.06690

[6] Michael Elad. 2010. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing* (1st ed.). Springer Publishing Company, Incorporated.

[7] C. M. Fu, E. Alshina, A. Alshin, Y. W. Huang, C. Y. Chen, C. Y. Tsai, C. W. Hsu, S. M. Lei, J. H. Park, and W. J. Han. 2012. Sample Adaptive Offset in the HEVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (Dec 2012), 1755–1764. https://doi.org/10.1109/TCSVT.2012.2221529

[8] Cheolkon Jung, Licheng Jiao, Hongtao Qi, and Tian Sun. 2012. Image deblocking via sparse representation. *Signal Processing: Image Communication* 27, 6 (2012), 663 – 677. https://doi.org/10.1016/j.image.2012.03.002

[9] X. Liu, X. Wu, J. Zhou, and D. Zhao. 2016. Data-Driven Soft Decoding of Compressed Images in Dual Transform-Pixel Domain. *IEEE Transactions on Image Processing* 25, 4 (April 2016), 1649–1659. https://doi.org/10.1109/TIP.2016.2526910

[10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning.* ACM, 689–696.

[11] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera. 2012. HEVC Deblocking Filter. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (Dec 2012), 1746–1754. https://doi.org/10.1109/TCSVT.2012.2223053

[12] W. S. Park and M. Kim. 2016. CNN-based in-loop filtering for coding efficiency improvement. In *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP).* 1–5. https://doi.org/10.1109/IVMSPW.2016.7528223

[13] Jens Schneider, Johannes Sauer, and Mathias Wien. 2017. Dictionary Learning based High Frequency Inter-Layer prediction for Scalable HEVC. In *Proc. of IEEE Visual Communications and Image Processing VCIP '17.* IEEE, Piscataway, St. Petersburg, USA.

[14] J. Wang and B. Shim. 2012. On the Recovery Limit of Sparse Signals Using Orthogonal Matching Pursuit. *IEEE Transactions on Signal Processing* 60, 9 (Sept 2012), 4973–4976. https://doi.org/10.1109/TSP.2012.2203124

[15] Roman Zeyde, Michael Elad, and Matan Protter. 2010. On single image scale-up using sparse-representations. In *International conference on curves and surfaces.* Springer, 711–730.