HAL
open science

Learning Clustering-Based Linear Mappings for Quantization Noise Removal

Martin Alain, Christine Guillemot, Dominique Thoreau, Philippe Guillotel

To cite this version:

Martin Alain, Christine Guillemot, Dominique Thoreau, Philippe Guillotel. Learning Clustering-Based Linear Mappings for Quantization Noise Removal. IEEE International Conference on Image Processing (ICIP) 2016, Sep 2016, Phoenix, United States. 10.1109/ICIP.2016.7533151 . hal-01317625

HAL Id: hal-01317625

https://inria.hal.science/hal-01317625v1

Submitted on 18 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LEARNING CLUSTERING-BASED LINEAR MAPPINGS FOR QUANTIZATION NOISE REMOVAL

Martin Alain [1,2], Christine Guillemot [1], Dominique Thoreau [2] and Philippe Guillotel [2]

[1] *INRIA*
Campus de Beaulieu, 35042 Rennes Cedex France
firstname.lastname@irisa.fr

[2] *Technicolor Research and Innovation*
Av. des Champs Blancs, 35576 Cesson-Sévigné France
firstname.lastname@technicolor.com

ABSTRACT

This paper describes a novel scheme to reduce the quantization noise of compressed videos and improve the overall coding performances. The proposed scheme first consists in clustering noisy patches of the compressed sequence. Then, at the encoder side, linear mappings are learned for each cluster between the noisy patches and the corresponding source patches. The linear mappings are then transmitted to the decoder where they can be applied to perform de-noising. The method has been tested with the HEVC standard, leading to a bitrate saving of up to 9.63%.

Index Terms: clustering, linear mapping, regression, de-noising, HEVC.

1. INTRODUCTION

The main principles of modern video codecs, such as H.264 [1], HEVC [2] or VP9 [3], first reside in the reduction of spatial and temporal redundancies, through prediction tools. The prediction residue is then transformed to further reduce the inner correlations of the signal, and the transformed coefficients are quantized to remove non-perceptive information. The quantized transformed coefficients are finally entropy coded to remove the remaining statistical redundancies. The improvements achieved over the years in video compression have been made by carefully optimizing each of these steps. For example, Intra prediction methods based on sparse representations have been proposed in [4], and later based on Locally Linear Embedding (LLE) of patches [5]. The prediction methods based on LLE have then been extended to spatio-temporal prediction in [6]. Recent works also investigate new transforms or quantization designs [7][8]. Despite the obvious improvement brought by these techniques in terms of rate-distortion (RD) performances, the irreversible quantization step still results in a degradation of the reconstructed frames.

In this paper, we adopt a different perspective by proposing a novel and generic out-of-the-loop de-noising scheme, complementary to the aforementioned works. Contrary to most state-of-the-art de-noising methods, the proposed scheme does not rely on an additive white Gaussian noise (AWGN) model. In fact, it is known that the quantization noise fall outside these assumptions [9], and explicit modeling of this type of noise is difficult. Thus, even top performing de-noising techniques such as the block-matching 3D algorithm (BM3D) [10] have a limited efficiency in our context.

The proposed approach first consists in clustering patches extracted from the reconstructed sequence. This clustering step can thus be performed at both the encoder and the decoder side. At the encoder side, we then learn for each cluster a projection function between the degraded patches and the corresponding source patches from the input sequence. These projection functions are then transmitted to the decoder, where they can be applied for each cluster in order to enhance the degraded patches quality.

The rest of this paper is organized as follow. Section 2 reviews related work on image de-noising. Section 3 describes the proposed clustering-based quantization noise removal scheme. Experimental results are then discussed in Section 4.

2. RELATED WORK

Video codecs such as H.264 [1] and HEVC [2] use in-loop deblocking filters to address blocking artifacts [11], which are the most visible artifacts due to the block transform and quantization. Compared to previous video compression standards, HEVC also introduced a Sample Adaptive Offset filter in addition to the deblocking filter. HEVC is thus a challenging codec for noise removal applications, and for this reason it was chosen to perform the experiments (see Section 4).

However, these in-loop filters are designed with a minimal complexity, and thus have a limited efficiency. Note that a more efficient Adaptive Loop Filter was also considered in [12]. The coefficients of a local Wiener filter are learned at the encoder in order to minimize the error between the decoded and the original pixels, and transmitted to the decoder. While this method performs a local pixel-based filtering, the technique we propose exploits instead non-local patch-based similarities. In fact, image de-noising has been a very active research topic, and more complex and efficient methods have been proposed. The BM3D is a two step algorithm [10] and among the best de-noising methods [13], as it combines the strengths of several previous de-noising approaches. It thus exploits non-local similarities as in the Non-Local Mean (NLM) algorithm [14], as well as filtering in the transform domain, such as in [15] [16].

For the sake of conciseness, we briefly describe here the first step of the BM3D algorithm, which shows the main principles used in this algorithm. Each block of pixels in the noisy image is stacked into a 3D group along with its nearest neighbors, in the sense of the Euclidean distance. A 3D transform is then applied on each group, and the coefficients are filtered using hard thresholding. The de-noised estimates are then obtained by applying the inverse 3D transform. Advantageously, all the blocks of the 3D group are processed at once. Several estimates are thus obtained for each pixel, which are eventually aggregated to obtain the final estimate. In the second step of BM3D, the estimate from the first step is used to perform Wiener filtering in the 3D transform domain.

More sophisticated noise models than AWGN have been proposed, such as the signal dependent noise model [17]. However,

the main approaches to remove such noise eventually rely on techniques designed for AWGN. Such techniques in fact assume that the noisy image can be partitioned into segments or clusters for which the noise can be modeled by AWGN [18][19].

Recent efficient de-noising methods also rely on sparse representations [20][21][22] or the local learning of dictionaries through clustering [23][24]. Nonetheless, it was shown that these methods are usually outperformed by the BM3D algorithm [13], which we thus choose to perform some experiments in Section 4.

3. CLUSTERING-BASED QUANTIZATION NOISE REMOVAL FOR COMPRESSION

In this section we describe the proposed compression scheme improvement, which uses clustering-based linear mappings learning for noise removal.

3.1. Proposed scheme for compression

We denote $\mathbf{M}_d$ and $\mathbf{M}_s$ the matrices containing in their columns all the vectorized $M \times N$ patches (*i.e.* pixel blocks) extracted from the decoded sequence Y and the source sequence X respectively.

The main idea of the proposed scheme is represented in Fig. 1, and consists in the following steps:

- At the encoder side:
  - cluster the decoded patches $\mathbf{M}_d$
  - for each cluster c, learn a linear mapping $\mathbf{P}_c$ between the decoded patches $\mathbf{M}_d^c$ and the corresponding source patches $\mathbf{M}_s^c$ by minimizing $||\mathbf{M}_s^{cT} - \mathbf{M}_d^{cT}\mathbf{P}_c^T||^2$
  - encode the corresponding linear mappings (in matrix form) and transmit them to the decoder
- At the decoder side:
  - decode the linear mappings
  - cluster the decoded sequence $\mathbf{M}_d$, as it is performed at the encoder side
  - for each cluster c, apply the corresponding linear mapping $\mathbf{P}_c$ to the decoded sequence patches $\mathbf{M}_d^c$ to obtain the de-noised patches $\mathbf{M}_r^c$:

$$\mathbf{M}_r^c = \mathbf{P}_c \mathbf{M}_d^c \tag{1}$$

We can see that the proposed method does not rely on any assumption on the noise model. We can thus apply this scheme to different codecs, and expect it to be robust to new designs of transform or quantizations which could affect the quantization noise.

Furthermore, we can combine our method to existing de-noising techniques, which only partially remove the noise in our context. We thus propose an alternative two step scheme, where a blind de-noising algorithm is first performed on the decoded frames, before applying the proposed method (see Fig. 1). The matrix $\mathbf{M}_d$ then contains in its column the vectorized de-noised patches. We qualify this first pass de-noising as "blind" because no parameters should be transmitted between the encoder and the decoder side, so that there is no bit-rate increase. In practice, some parameters that have a negligible rate cost are sent.

We discuss more in details the different steps of the approach in the following sections.

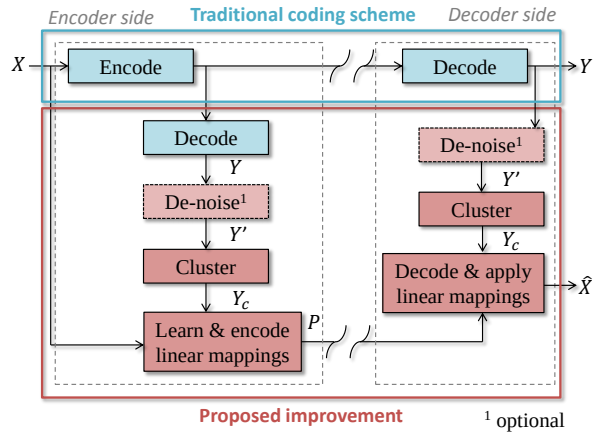

Fig. 1. Proposed compression scheme using noise removal based on clustering and linear mappings learning.

3.2. Clustering

The clustering is performed on $M \times N$ non-overlapping patches extracted from the decoded frames. Any clustering algorithm is eligible for the proposed method, as long as it can be performed similarly at the encoder and the decoder side. We choose the popular K-means algorithm [25]. Note that some tests performed with more advanced clustering methods, such as spectral clustering [26], showed that the de-noising performances were in the end very similar, despite the increased complexity.

Selecting the number of clusters K is crucial for the overall RD performances, as increasing K improves the de-noising performances, but also the bit-rate associated to the linear mappings. Empirical results show that 10 is a good order of magnitude for K (see Section 4). However, a more specific tuning of K depending on the video content and the target bit-rate can improve the final RD performances. To address this problem, we present below an adaptive method for selecting K, based on an rate-distortion optimization (RDO) criterion.

Instead of explicitly partitioning the data into K clusters, we propose a recursive binary partitioning, which proceeds as follow:

- At initialization, the full data set is considered as one cluster.
- Each cluster is then recursively split into 2 clusters if the RDO criterion (explained below) is satisfied, creating a binary tree structure.
- The procedure stops when no RDO criterion is satisfied, or a pre-defined maximum tree depth is reached.

The RDO criterion used to decide to further split a cluster is designed to balance the de-noising performances and the linear mappings rate cost. Given a cluster C_n and its corresponding mapping $\mathbf{P}_n$, we can allocate it a RD cost computed as:

$$J_n = D_n + \lambda R_n \tag{2}$$

The distortion D_n is computed as the Sum of Squared Error (SSE) between the patches of C_n de-noised with $\mathbf{P}_n$ and the corresponding source patches. The rate R_n is estimated as the number of bits of the encoded mapping $\mathbf{P}_n$. The Lagrangian parameter is computed as in the test model of HEVC:

$$\lambda = QP_{factor} \times 2^{((QP-12)/3)} \tag{3}$$

where QP is the quantization parameter and QP_{factor} is an adjustment factor, set to 1 in our experiments. Thus, to decide if C_n is

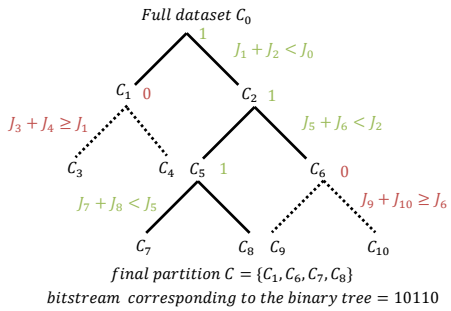

Fig. 2. Example of a cluster partition based on a binary tree structure and an RDO criterion. The clusters are recursively split to reach 4 clusters. The tree structure is described by only 5 bits.

further divided in two clusters C_{n+1} and C_{n+2}, we merely check the condition $J_{n+1} + J_{n+2} < J_n$. Note that this method is independent from the clustering algorithm. An example of binary tree structure obtained after recursive partitioning is given in Fig. 2.

To perform the exact same partitioning at the decoder, we then need to transmit the binary tree structure. In our experiment, the maximum binary tree depth is set to 4, which corresponds to a maximum of 16 clusters and requires a maximum of 15 bits to be encoded. Thus, the bit-rate allocated to the binary tree structure is negligible compared to the bit-rate of the encoded sequence.

3.3. Linear mappings learning and encoding

We first describe here the process to learn the linear mappings at the encoder, which are then used to perform de-noising at the decoder side (see Eq. 1). Considering multivariate linear regression, the problem is of searching for each cluster c the function $\mathbf{P}_c$ minimizing:

$$\mathbf{E} = ||\mathbf{M}_s^{c\,T} - \mathbf{M}_d^{c\,T}\mathbf{P}_c^T||^2 \tag{4}$$

which is of the form $||\mathbf{Y} - \mathbf{XB}||^2$ (corresponding to the linear regression model $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$). The minimization of Eq. (4) gives the least squares estimator

$$\mathbf{P}_c = \mathbf{M}_s^c\mathbf{M}_d^{c\,T}(\mathbf{M}_d^c\mathbf{M}_d^{c\,T})^{-1} \tag{5}$$

The linear mappings are thus obtained in a matrix form, which then need to be transmitted to the decoder. For that purpose, we consider the K linear mapping matrices of size $(MN) \times (MN)$ as a sequence of K images, and encode them using a video codec. An example of such matrices is given in Fig. 3, where we can observe redundancies within and between the different matrices.

The linear mappings matrices consist originally in floating points values, thus they need to be quantized to integer values before encoding. To avoid a loss in terms of de-noising performance due to the quantization, the matrices are quantized on 16 bits integer values. The matrices are then encoded with the Range Extension (RExt) of HEVC [27][28], which supports 16 bits input. The quantization parameter of HEVC was chosen very low ($QP = -30$) to limit the loss on the de-noising performances. Finally, in order to recover the floating point matrices, the minimum and maximum values of the original floating points matrices need to be sent to the decoder without loss (with a 32 bits precision).

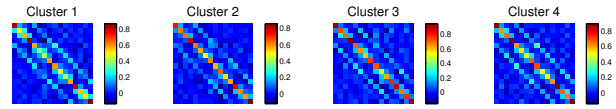

Fig. 3. Linear mapping matrices obtained on the first frame of the Kimono sequence encoded with HEVC at QP=22.

3.4. Oracle clustering for de-noising based on linear mappings learning

Finally, we describe in this section a so-called oracle clustering, which goal is to maximize the de-noising performances of the proposed scheme for a fixed number of clusters K. In other words, we want to minimizes the distortion at the decoder side while keeping a steady bit-rate associated to the linear mappings. For that purpose, the oracle clustering jointly learns the partitioning and the associated linear mappings in order to minimize the overall reconstruction error. Note that this clustering is denoted "oracle", as it relies on the knowledge of the source patches, and can not be directly use in our scheme as it is not reproducible at the decoder side. Nonetheless, it allows to evaluate an upper bound on the coding performances.

Formally, the problem corresponding to the oracle clustering is formulated as follows:

$$\min_C \sum_{i=1\ldots K} \sum_{x_d, x_s \in C_i} ||x_s - \mathbf{P}_i x_d||^2 \tag{6}$$

where x_s and x_d are corresponding source and decoded patches respectively, and $C = \{C_i\}_{i=1\ldots K}$. The problem of Eq. (6) can be solved using a greedy algorithm, described in Algorithm 1.

The initial set of linear mappings $\mathbf{P}_1^{(1)} \ldots \mathbf{P}_K^{(1)}$ is supposed known, e.g. obtained by first performing the K-means clustering on the coded/decoded patches and learning a linear mapping for each cluster.

Algorithm 1 Oracle clustering for linear mapping learning

Input: cluster number K, coded/decoded patches matrix $\mathbf{M}_d = \{x_d\}$, corresponding source patches matrix $\mathbf{M}_s = \{x_s\}$
Output: cluster set $C = \{C_i, 1 \leq i \leq K\}$, corresponding mapping matrices $\mathbf{P} = \{\mathbf{P}_i, 1 \leq i \leq K\}$
Repeat the two following steps until convergence:
Assignment step:
$\forall i = 1 \ldots K,$
$C_i^{(t)} = \{x_d, x_s : ||x_s - \mathbf{P}_i^{(t)} x_d||^2 \leq ||x_s - \mathbf{P}_j^{(t)} x_d||^2 \forall j, 1 \leq j \leq K\}$

Update step:

$$\forall i = 1 \ldots K, \mathbf{P}_i^{(t+1)} = \mathbf{M}_s^i \mathbf{M}_d^{i\,T}(\mathbf{M}_d^i \mathbf{M}_d^{i\,T})^{-1}$$

4. SIMULATIONS AND RESULTS

The sequences used in the experiment are presented in Table 1, and mainly consist in HEVC test sequences. The test sequences are processed per Group Of Pictures (GOP), where a GOP contains a number of frames equal to the frame rate. The sequences are encoded with the HEVC test model HM (ver 15.0) [29] using the Main profile in Random Access, with 4 values for the Quantization Parameter, $QP = 22, 27, 32, 37$. Note that the proposed method is applied on the luminance channel. The patch size is set to $M \times N = 4 \times 4$, which has been observed as offering a good trade-off between de-noising performances and rate cost for transmitting the linear projection matrices.

We present below rate-distortion performances for several experiments, computed using the Bjontegaard metric [30] with respect to the sequences encoded with HEVC. More detailed results are available at `http://www.irisa.fr/temics/demos/clusteringLinearMappingNoiseRemoval/clusteringLinearMappingNoiseRemoval.html`.

Table 1. Test sequences

Sequence	Resolution	Frames count	Frame rate
City	1280×720	600	60
Park Scene	1920×1080	240	24
Tennis	1920×1080	240	24
Kimono	1920×1080	240	24
Cactus	1920×1080	500	50
Terrace	1920×1080	600	60
Basket	1920×1080	500	50
Ducks	1920×1080	500	50
People On Street	2560×1600	150	30
Traffic	2560×1600	150	30

We show in Table 2 the RD performances of our method (without first pass de-noising) for each sequence, first performed with a fixed number of clusters $K = 10$ (left column) and then compared to the adaptive selection of K (center column). The results are averaged over all GOPs. We can see that almost all sequences benefit from the adaptive K. We show the selected value of K as well, averaged over the four QP values, which varies between about 5 and 16 depending on the sequence. The number of clusters is generally reaching higher values for sequences with a higher resolution and/or frame rate. The K value depends on the bit-rate as well, and we have in average over the sequences $K = 12.3, 11.1, 9.0, 8.0$ for $QP = 22, 27, 32, 37$ respectively.

In the right column, we also give the performances of the oracle clustering with $K = 10$ clusters, which show that the proposed method could potentially reach about 30% of bit-rate reduction in average. Note that these results merely represent an upper bound on the performances, as the oracle clustering can not be performed independently at the decoder side.

We then show in Table 3 the RD performances of our method, first using the one step scheme as above (left column), then compared to the two step scheme (right column), where the videos are first de-noised with the extension of BM3D to videos (VMB3D) [31]. The noise standard deviation parameter is estimated as the square root of the MSE computed on a GOP at the encoder side, and is then sent to the decoder to perform the exact same de-noising. We use the adaptive K selection for both schemes. Results are given for the first GOP of the sequences.

We can see that a substantial bit-rate reduction can be achieved when combining the proposed method with the VBM3D. In fact, the VBM3D only partially removes the quantization noise, which is further reduced when using the proposed method in the second step. All sequences benefit from the two step schemes, and in particular the *People On Street* sequence, which reaches almost 10% of bit-rate saving.

We also give the selected K values, and we can see that the number of clusters selected is higher when using the first pass de-noising. In fact, in the RDO decision, the improvement brought by the first pass de-noising reduces the distortion, and thus allows for a higher bit-rate cost, *i.e.* more linear mappings can be sent to the decoder.

It has been observed that the processing time increase brought by the proposed method (with a non-optimized Matlab implementation) compared to the HM software is around 110% at the encoder and around 4000% at the decoder side. In the current version, more than 95% of the proposed method running time is dedicated to the K-means clustering. We thus expect the complexity increase to be much more reduced with an optimized implementation, e.g. considering fast algorithms [32], or parallel implementation [33], possibly on GPU [34].

Table 2. RD performances of the proposed scheme averaged over all GOPs (Bjontegaard bit-rate gain with respect to HEVC)

	K-means			Oracle clustering
	$K = 10$	Adaptive K		$K = 10$
	BD-rate	BD-rate	K value	BD-rate
City	-1.81	**-2.24**	8.9	*-28.92*
Park Scene	0.38	**-0.47**	5.8	*-21.66*
Tennis	0.08	**-1.42**	4.9	*-38.20*
Kimono	0.43	**-0.94**	5.8	*-33.60*
Cactus	-1.08	**-1.15**	10.7	*-23.48*
Terrace	**-8.28**	-8.08	14.4	*-39.50*
Basket	-0.65	**-0.80**	8.8	*-28.99*
Ducks	-1.88	**-1.96**	15.8	*-22.07*
People On Street	-2.56	**-2.60**	16.0	*-23.26*
Traffic	-1.26	**-1.43**	10.0	*-25.50*
Average	-1.66	**-2.11**	10.1	*-28.52*

Table 3. RD performances of the proposed schemes for the first GOP (Bjontegaard bit-rate gain with respect to HEVC)

	Adaptive K		Two step scheme	
	BD-rate	K value	BD-rate	K value
City	-2.76	8.5	**-3.24**	12.3
Park Scene	-0.69	5.8	**-1.07**	7.5
Tennis	-0.90	6.0	**-7.77**	6.8
Kimono	-1.32	5.8	**-4.32**	5.8
Cactus	-1.12	12.0	**-6.18**	13.0
Terrace	-5.25	15.8	**-7.24**	15.8
Basket	-1.10	7.5	**-3.80**	9.0
Ducks	-2.39	15.3	**-4.13**	16.0
People On Street	-2.57	16.0	**-9.63**	15.5
Traffic	-1.44	10.3	**-5.43**	11.5
Average	-1.96	10.3	**-5.28**	11.3

5. CONCLUSION AND PERSPECTIVES

In this paper we introduced a novel generic scheme designed to de-noise compressed videos and improve the coding performances of existing codecs. The compressed video patches are first clustered, and linear mappings are then learned between the decoded and source patches at the encoder side. These linear mappings are then transmitted to the decoder to perform de-noising. We also propose an algorithm to optimally select the number of clusters in a RD sense. Experimental results thus show that our method can improve the coding performances of the HEVC standard. In addition, the proposed technique is also complementary to existing de-noising approaches, and by combining the different techniques, we can reach even higher bit-rate reduction, up to about 10%.

Furthermore, we showed through the oracle clustering that much higher coding gains could be reached, which calls for the investigation of more specific clustering techniques instead of the K-means algorithm. Other perspectives could be explored as well thanks to the genericity of the proposed scheme. For example, an extension to color videos is quite straightforward, by vectorizing the three components of the patches instead of the sole luminance. Moreover, extension to scalable applications can be considered, such as spatial scalability, for which the proposed scheme could be used to perform joint de-noising and super-resolution. In fact, a similar method has been proposed recently for super-resolution applications in [35]. Finally, LDR/HDR or color gamut scalabilities could be also investigated [36].

6. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[2] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje, "The latest open-source video codec VP9 - An overview and preliminary results", *Pict. Coding Symp.*, pp. 390–393, 2013.

[4] M. Türkan and C. Guillemot, "Sparse approximation with adaptive dictionary for image prediction", in *IEEE Int. Conf. Image Process.*, 2009, pp. 25–28.

[5] M. Türkan and C. Guillemot, "Image prediction based on neighbor embedding methods.", *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 1885–98, 2012.

[6] M. Alain, C. Guillemot, D. Thoreau, and P. Guillotel, "Inter-prediction methods based on linear embedding for video compression", *Signal Process. Image Commun.*, vol. 37, pp. 47–57, 2015.

[7] A. Arrufat, P. Philippe, and O. Deforges, "Mode-dependent transform competition for HEVC", in *IEEE Int. Conf. Image Process.*, 2015, pp. 1598–1602.

[8] J.-M. Valin and T. B. Terriberry, "Perceptual vector quantization for video coding", in *SPIE Vis. Inf. Process. Commun.*, 2015, vol. 9410.

[9] O. G. Guleryuz, "Linear, worst-case estimators for denoising quantization noise in transform coded images", *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2967–2986, 2006.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering", *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[11] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera, "HEVC Deblocking Filter", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, 2012.

[12] C.-Y. Chen, C.-Y. Tsai, Y.-W. Huang, T. Yamakage, I. S. Chong, et al., "The adaptive loop filtering techniques in the HEVC standard", in *Proc. SPIE*, 2012, p. 8499.

[13] P. Chatterjee and P. Milanfar, "Is denoising dead?", *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, 2010.

[14] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising", in *Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 60–65.

[15] D. L. Donoho, "De-noising by soft-thresholding", *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[16] S. P. Ghael, A. M. Sayeed, and R. G. Baraniuk, "Improved wavelet denoising via empirical Wiener filtering", in *Proc. SPIE*, 1997, pp. 389–399.

[17] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel, "Adaptive Noise Smoothing Filter for Images with Signal-Dependent Noise", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-7, no. 2, pp. 165–177, 1985.

[18] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 299–314, 2008.

[19] X. Liu, M. Tanaka, and M. Okutomi, "Signal dependent noise removal from a single image", in *IEEE Int. Conf. Image Process.*, 2014, pp. 2679–2683.

[20] M. Elad and M. Aharon, "Image denoising via sparse and redundant representation over learned dictionaries", *IEEE Transations Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[21] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[22] J. Sulam, B. Ophir, and M. Elad, "Image denoising through multi-scale learnt dictionaries", in *IEEE Int. Conf. Image Process.*, 2014, pp. 808–812.

[23] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries", *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, 2009.

[24] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering", in *Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 457–464.

[25] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations", *5th Berkeley Symp. Math. Stat. Probab. 1967*, pp. 281–297, 1967.

[26] S. Uw, A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm", *Adv. Neural Inf. Process. Syst. 14*, pp. 849–856, 2001.

[27] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)", *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 1001–1016, 2013.

[28] "HM RExt Software, Ver. HM 15.0 RExt 8.1. Available: https://hevc.hhi.fraunhofer.de/trac/hevc/browser".

[29] "HM Software, Ver. 15.0. Available: https://hevc.hhi.fraunhofer.de/trac/hevc/browser".

[30] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", *Doc. VCEG-M33, ITU-T VCEG Meet.*, 2001.

[31] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering", *Eur. Signal Process. Conf.*, pp. 145–149, 2007.

[32] G. Hamerly, "Making k-means even faster", *SIAM Int. Conf. data Min.*, pp. 130–140, 2010.

[33] Y. Zhang, Z. Xiong, J. Mao, and L. Ou, "The Study of Parallel K-Means Algorithm", *6th World Congr. Intell. Control Autom.*, pp. 5868–5871, 2006.

[34] S. Giuroiu, "CUDA K-Means Clustering", Available: https://github.com/serban/kmeans.

[35] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong, "Learning Multiple Linear Mappings for Efficient Single Image Super-Resolution", *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 846–861, 2015.

[36] P. Korshunov and T. Ebrahimi, "Context-dependent JPEG backward-compatible high-dynamic range image compression", *Opt. Eng.*, vol. 52, no. 10, pp. 102006, 2013.