A Signal Adaptive Prediction Filter for Video Coding Using Directional Total Variation: Mathematical Framework and Parameter Selection

Jennifer Rasch, Victor Warno, Jonathan Pfaff, Caren Tischendorf, Detlev Marpe, *Fellow, IEEE*, Heiko Schwarz, and Thomas Wiegand, *Fellow, IEEE*

***Abstract*— In this paper we combine video compression and modern image processing methods. Iterative filter methods for prediction signals based on classic inpainting methods are introduced and extensive parameter tests are described. In order to construct an alternative prediction filter for video coding, techniques originally employed for inpainting are applied. Thereby, the structures of the underlying prediction were incorporated into the filter construction making it signal adaptive. The resulting optimization problem is solved using the so-called Alternating Direction Method of Multipliers (ADMM). The undertaken novel parameter tests are described and it is shown that they improve the coding efficiency of the tool. The suggested filter is embedded into a software based on HEVC with additional QTBT (Quadtree plus Binary Tree) and MTT (Multi-Type-Tree) block structure. Overall, the proposed filter method obtains average bitrate savings of 1.35% at an average encoder runtime increase of 28% and decoder runtime increase of 38%. UHD test sequences achieve bitrate savings of up to 3.66% for Random Access.**

***Index Terms*— High efficiency video coding (HEVC), intra prediction, inter prediction, PDEs, adaptive filtering, edge detection, linear filtering, nonlinear filtering, video compression, total variation, Alternating Direction Method of Multipliers (ADMM).**

I. Introduction

THE increasing demand for high resolution videos, together with limited transmission and memory capacity is still driving the research on video compression. As a core technique in state-of-the-art video codecs such as High Efficiency Video Coding (HEVC, [1]) a hybrid approach with block based architecture is used. The term "hybrid" refers to a combination of prediction from previous decoded frames (inter) or adjacent decoded blocks from the frame itself (intra prediction) together with transform coding of the resulting residual. Thus, the quality of the prediction signal has a large influence on the efficiency of video codecs. In this paper, a method first introduced in [2] based on state-of-the-art mathematical denoising techniques is elaborated in more detail. The central idea of this method is to construct a signal adaptive prediction filter that incorporates the structures of the underlying prediction signal: to realize this, a variational approach is used.

In the field of video coding attempts have been made to use such total variational methods coming from inpainting to replace intra prediction modes ([3], [4]). Due to their diffusion properties these methods have the disadvantage that they cannot prolong edges well and are therefore not suitable to replace angular prediction modes. This is the reason why in this paper instead of replacing the prediction, the prediction provided by the video codec is chosen as initial condition.

Fig. 1 shows a block diagram of a classic hybrid video encoder with enclosed decoder depicting the newly introduced additional prediction filter step that is applied on top of the selected prediction signal. So far, little work has been published on tools increasing the efficiency of the codec that operate at this particular location. Classically ([5]), linear filtering techniques have been applied to the reference samples before predicting to improve intra prediction. Furthermore after predicting, modern video codecs employ linear smoothing techniques for certain intra prediction modes, which are only applied on the block boundaries (boundary smoothing, see [5], [6]).

Recently [7], intra prediction modes have been proposed for video coding which are based on neural networks. The method uses the reconstructed samples to generate an affine linear combination of predefined image patterns as the prediction signal.

In case of inter, most methods to improve prediction operate directly at the generation of the prediction block ([8]): Weighted prediction approaches that superimpose different reference blocks can be applied to increase the accuracy of the current prediction, for example. Another classical method are so-called interpolation filters, that use fixed sets of filter coefficients to interpolate in between the samples of the reference block(s) to generate fractional samples for an accurate inter prediction. In [9], adaptive interpolation filter coefficients

Manuscript received December 5, 2019; revised August 21, 2020; accepted October 1, 2020. Date of publication October 19, 2020; date of current version October 23, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mireille Boutin. *(Corresponding author: Jennifer Rasch.)*

Jennifer Rasch, Victor Warno, Jonathan Pfaff, Detlev Marpe, Heiko Schwarz, and Thomas Wiegand are with the Video Coding & Analytics Department, Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: j.rasch@protonmail.com).

Caren Tischendorf is with the Department of Mathematics, Humboldt University of Berlin, 10117 Berlin, Germany.

Digital Object Identifier 10.1109/TIP.2020.3030590

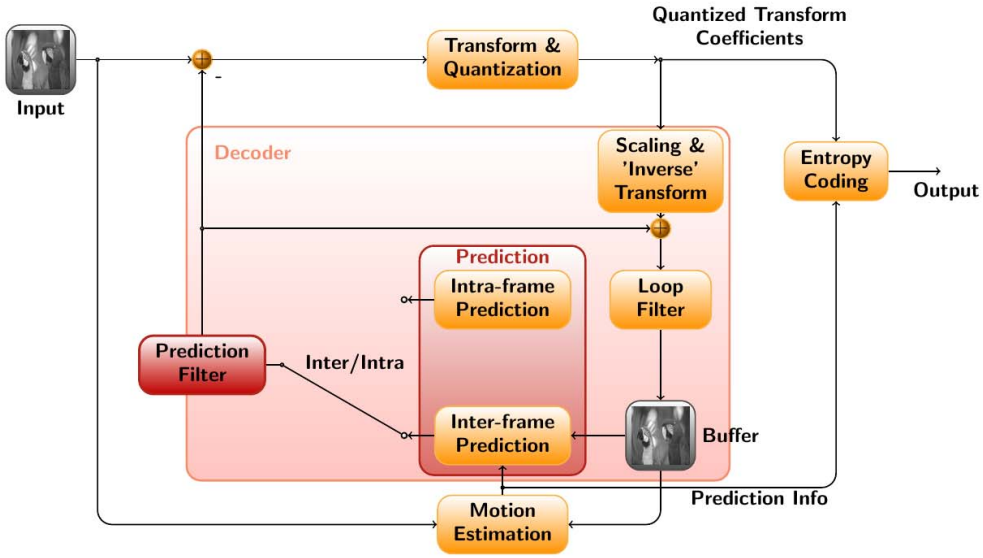

Fig. 1. Block Diagram of a hybrid video encoder with newly introduced prediction filter and enclosed decoder.

have been proposed which are fitted and signaled frame by frame.

Only fairly recent, post-prediction methods have been suggested for the new video compression standard using affine linear models to compensate local illumination changes [10], [11]. As the methods established in [12] and used here, this tool operates on the prediction signal.

The methods used here as the ones described in [12] are signal adaptive in the sense that they are signaled block-wise. Thereby, they fit into the framework of classic video coding tools. But on the other hand, in opposition to other tools used in video coding, they are unique in the sense that they are based on nonlinear mathematical models and are therefore highly (i.e. partly sample-wise) signal adaptive.

In order to overcome the disadvantage of uniform inpainting attenuating edges, these models base on the idea of using some kind of a priori knowledge about the structures in the image. Rudin, Osher and Fatemi [13] developed a method to smooth images preserving sharp edges using constraints that involve the statistics of noise. Based on this, Chan and Shen [14] applied total variation methods to fill in missing parts of damaged images recovering edges based on the information from surrounding areas. This approach has been further developed by [15]. In [16] the authors develop a method to reconstruct images. In [17] such a method using structural a priori knowledge is applied to increase the spatial resolution of an image. All these ideas are based on similar mathematical methods.

Now, in this paper, as in [2], they are used to develop a filtering process preserving sharp edges for the prediction signal in video coding. The resulting optimization problem can be solved using the Alternating Direction Method of Multipliers (ADMM) [18, p. 13] which is why this filter is referred to as ADMM filter in the following.

The diffusion filter method described in [7], [12], [19], [20] is based on similar ideas as the one employed here. In fact, in can be shown [21], that in a continuous setting for a certain choice of parameters, the problems are equivalent. However, the ADMM filter which is employed here uses an optimization formulation with a penalty approach which is solved using the ADMM method. Numerically, this leads to a different solution. For a more detailed comparison, see [21]. In contrast to the diffusion filter method, the ADMM filter does not require testing (and signaling) of variable configurations.

This paper is organized as follows. In section II an overview of the current state-of-the-art in video coding is given. section III explains how the prediction filter for video coding is constructed based on an optimization approach as in [2]. In section III-A and III-B it is explained how structural a priori knowledge can be incorporated into the problem formulation. Additional to [2], section III-C motivates the selected solving method Alternating Direction Method of Multipliers (ADMM). In section IV the finite discretization and the ADMM solving method are described in more detail than in [2]. In contrast to [2] which was implemented into HEVC, here, the chosen filter approach is implemented into a software based on HEVC which uses a QTBT with MTT block structure. The main addition to [2] in this paper are the extensive parameter tests in order to improve the filter efficiency in section V. Eventually, in section VI, the bitrate savings for the ADMM filter are reported. In this paper the bitrate savings -also referred to as rate-distortion (RD) gains- are generally measured in terms of Bjøntegaard delta (BD) bitrate [22].

II. State of the Art Video Coding

The standardization of video coding technology plays a major role in the broad adoption and growing popularity of video technology. The main purpose of a video coding standard is to define the interface between encoder and decoder to ensure interoperability among a wide range of devices. Video coding standards are designed to provide a maximal

degree of freedom for the manufacturers to adapt the encoder to specific applications. More details can be found in [23].

All modern video codecs use the concept of hybrid coding. The term 'hybrid' goes as far back as to the 1980s [24] and describes a combination of two fundamental concepts of video coding, predictive coding and transform coding. The basic architecture of a hybrid video encoder together with an enclosed decoder is shown in Fig. 1, additionally depicting the new prediction filter step proposed here. The flow of the encoder is depicted using continuous lines.

For the sake of simplicity, here, we will describe the encoder only. Note that the standard does not imply a specific encoding approach. The following is to be read as an example of a particular encoder such as can be found in [25].

The video signal is divided into pictures and the pictures are split into blocks. Typically, the picture is divided into macroblocks of a fixed size consisting of a luma and two corresponding chroma components. In HEVC [1], the picture is divided into coding tree units (CTUs) of a size selected using a configuration parameter in the encoder. A CTU consists of a luma coding tree block (CTB) and two corresponding chroma CTBs. A CTB may contain only one coding unit (CU) or may be split into several CUs using a quadtree decomposition.

Assuming there are already reconstructed blocks in the picture storage, a prediction for the current block is formed using preceding blocks. If those blocks are spatially adjacent, the resulting prediction is called intra prediction. If they are taken from already reconstructed pictures, i.e., are temporally preceding, one speaks of inter prediction. The decision if intra or inter prediction is used is taken on CU level. To optimize the usage of already reconstructed pictures in the inter case a motion estimation is performed. The residual between the prediction and the original block is calculated, transformed and quantized using a certain quantization parameter (QP). The resulting coefficients are fed into the entropy coder. The entropy coder typically uses either variable-length coding (VLC) (e.g., Huffman codes [26]) or arithmetic coding (e.g., context-based adaptive binary arithmetic coding - CABAC [27]). To obtain the reconstructed samples, the quantized transform coefficients are rescaled and retransformed. Note that due to the quantization process, a loss of information takes place. Therefore, the reconstruction differs from the original. The reconstructed blocks are loop filtered and stored in the picture buffer where they are used to predict the following blocks.

III. Adaptive Filtering Using the Alternating Direction Method of Multipliers: A Variational Optimization Approach

In this section, it is shown how optimization methods coming from inpainting using variational approaches can be used to establish a novel filter method for prediction signals in video coding.

Let f be the initial prediction provided by the underlying video codec and extended by the reconstructed samples on the upper and left side and u its filtered version. In a continuous setting, extended prediction f can be seen as a function $f\colon R \subset \mathbb{R}^2 \to \mathbb{R}$ where $R := \Omega \cup \Gamma$ is a set consisting of a rectangular domain $\Omega \subset \mathbb{R}^2$ and the left and upper boundary Γ of Ω. Additionally, we introduce the notion of S, an operator that restricts the spatial domain of function $u\colon R \subset \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ to the reconstructed area such that

$$Su = u_{|\Gamma}.$$

In a continuous setting this notion is defined using the Trace Theorem ([28]). In order to make sure our filtered solution is close to the reconstructed samples on the sides, we want $\|Su - f_{|\Gamma}\|^2$ to be small. Using a penalty approach, we introduce a penalty term (also referred to as prior) $\mathcal{J}$ that is to be constructed in such a way that it results in a large value for undesirable solutions u. Now, the optimization problem reads as follows: The filtered image u shall solve

$$\operatorname*{argmin}_u \{\frac{1}{2}\|Su - f_{|\Gamma}\|^2 + \alpha \mathcal{J}(u)\} \tag{1}$$

for scalar-valued function $\mathcal{J}(u)$ and parameter $\alpha \in \mathbb{R}$, $\alpha > 0$ with boundary condition

$$\frac{\partial}{\partial \nu} u(p,t) = 0 \quad \forall p \in \partial R, \forall t \in \mathbb{R} \tag{2}$$

where ν denotes the normalized vector perpendicular to the boundary pointing outwards (outer normal) and ∂R the boundary of set R. The boundary condition specifies the values that the solution needs to take along the boundary of the domain. The gradient here is to be understood in a weak sense and $\|\cdot\|$ denotes the L^2 norm. The time parameter t relates to the strength of the filtering. Starting for $t = 0$ with $u(\cdot, 0) = f(\cdot)$, for $t > 0$ solution u is getting filtered. For notational convenience, parameter t is neglected in the following.

A. Overall Smoothing

A common way of smoothing images is to penalize large gradients. To do so, one chooses

$$\mathcal{J}(u) = \int_\Omega |\nabla u(x)|\, dx, \tag{3}$$

where $|\cdot|$ denotes the euclidean norm $|x| = \sqrt{x^T x}$. This choice of prior $\mathcal{J}$ is called total variation. However, since this formulation does not incorporate any a priori knowledge about the image structure, this choice leads to a strong overall smoothing. For predictions containing strong edges this might not be desirable. Furthermore, it is known that total variation can lead to a piecewise constant solution which is also called "staircasing effect" [29]. These disadvantages can be avoided using the following choice of prior.

B. Prior Information on Direction of Edges

It is reasonable to assume that the filtered solution u shares not only the location but also the direction of edges with the initial image f. Smoothing along edges should be preferred over smoothing in perpendicular direction. In order to realize this, $\mathcal{J}$ can be chosen as

$$\mathcal{J}(u) = \int_\Omega |D\nabla u(x)|\, dx. \tag{4}$$

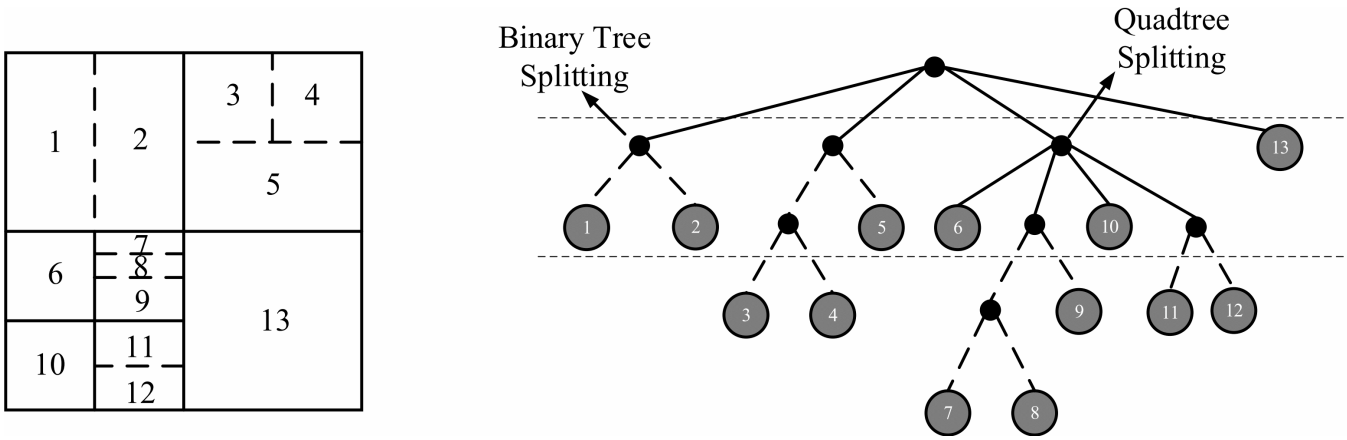

Fig. 2. Illustration of a QTBT structure, taken from [33], ©2017 IEEE.

Here, for fixed, given f, we set $\xi = \frac{\nabla f}{\|\nabla f\|_\eta}$, where $\|\cdot\|_\eta^2 = \|\cdot\|^2 + \eta^2$ and define operator $D\colon L^2(\Omega, \mathbb{R}^2) \to L^2(\Omega, \mathbb{R}^2)$ as

$$Dv = v - \langle \xi, v\rangle \xi \text{ for } v,\ \xi,\ \nabla f \in L^2(\Omega, \mathbb{R}^2).$$

The edge parameter η is related to the size of an edge and has to be chosen accordingly (for details, see section V). The operator D is called anisotropic diffusion tensor. Now, eq. (4) can be reformulated as

$$\mathcal{J}(u) = \int_\Omega |\nabla u(x) - \langle \xi,\ \nabla u\rangle \xi(x)\ | \, dx, \tag{5}$$

where $\langle \cdot,\ \cdot \rangle$ denotes a scalar product. This choice of prior $\mathcal{J}$ is called directional total variation. Note that due to the linearity of D, function $\mathcal{J}$ as chosen above is convex. Here, $\mathcal{J}(u)$ penalizes smoothing into directions not parallel to $\nabla f(x)$, especially those perpendicular to $\nabla f(x)$. In the following, the argument x is omitted. The expression $\nabla u - \langle \xi,\ \nabla u\rangle \xi$ reduces to $(1 - |\xi|^2)\nabla u$ in regions where ∇u is collinear to ξ (i.e., $\exists \zeta > 0$ s.t. $\xi = \zeta \nabla u$) and to ∇u where ∇u is orthogonal to ξ. Thus, gradients that are collinear to ξ are favored (since $\mathcal{J}(u)$ becomes smaller) as long as $|\xi| \neq 0$. The strength of the penalization depends on the norm $|\nabla u|$.

C. Theoretical Foundations

For the choice of total and directional variation, in the continuous case the existence of a minimizer can be shown for eq. (1) -using standard theorems found for example in [30]–[32]. However, the uniqueness of the minimizer is not given in general since the functional in eq. (1) is not strictly convex. As is explained in [18], the ADMM method was developed to yield convergence without strong assumptions like strict convexity. Therefore, the ADMM solving method is a feasible choice and it is shown in the latter that the chosen approach achieves impressive results in the suggested application to video coding here.

IV. Finite Discretization and Implementation

Let $R := \Omega \cup \Gamma$ be a rectangular set in $\mathbb{R}^2$ of dimension $(N + \ell) \times (W + \ell)$ where $N \times W$ coincides with the size of the current block and ℓ denotes the width of the extension on the left and upper side. Here, Ω denotes the rectangular set of dimension $N \times W$ and its extended area shall be denoted by Γ. Let the initial image f consist of the given prediction block of dimension $N \times W$ extended using the ℓ reconstructed samples on the left and upper side. For simplicity, it can be assumed that $\ell = 1$. Assume that S is a mask that restricts the domain of function u to the reconstructed area such that $Su = u_{|\Gamma}$. Now, solve

$$\underset{u}{\operatorname{argmin}}\{\frac{1}{2}\|Su - f_{|\Gamma}\|^2 + \alpha \mathcal{J}(u)\} \tag{6}$$

with

$$\mathcal{J}(u) = \sum_n |D_n \nabla u(p_n)|. \tag{7}$$

using boundary condition

$$\frac{\partial}{\partial \nu} u(p) = 0 \quad \forall p \in \partial R.$$

Here, set

$$D_n = I - \xi_n \xi_n^T \text{ and } \xi_n = \frac{\nabla f(p_n)}{|\nabla f(p_n)|_\eta},$$

where p_n ranges over the number of image points in Ω and $\nabla f(p_n) \in \mathbb{R}^2$, $|\cdot|_\eta^2 = |\cdot|^2 + \eta^2$ and $I \in \mathbb{R}^{2\times 2}$ denotes the identity. Eq. (6) is solved using the ADMM method (as described in section IV-C) and then the initial prediction f in the codec is replaced by its filtered descendant u. This filter method is referred to as ADMM filter in this paper.

A. Implementation

The ADMM filter is embedded into a software based on HEVC [34] that includes an additional QTBT block structure and MTT partitioning: That means that the quadtree structure of HEVC is replaced by a Quadtree plus Binary Tree (QTBT, [35]) block structure. An example for a QTBT partitioning is shown in Fig. 2. The CTUs are firstly divided in a quadtree manner and then further partitioned using a binary tree structure. QTBT allows more flexibility in the shape of the CU structure which can be rectangularly shaped now instead of

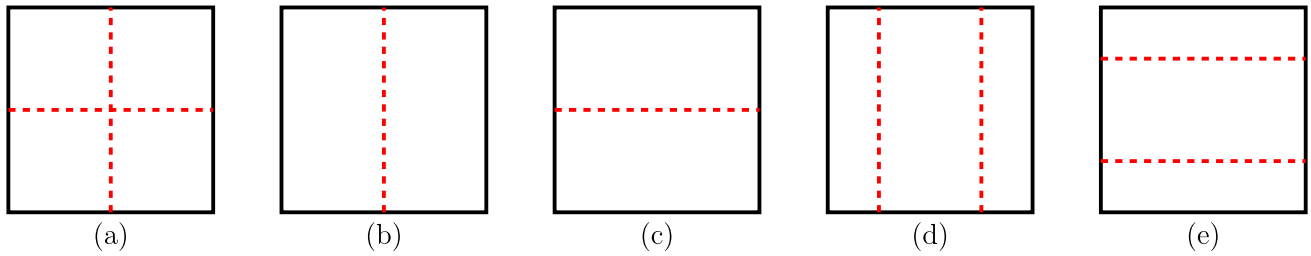

Fig. 3. Multi-Type-Tree structure, (a) quad-tree partitioning (b) vertical binary-tree partitioning (c) horizontal binary-tree partitioning (d) vertical center-side triple-tree partitioning (e) horizontal center-side triple-tree partitioning.

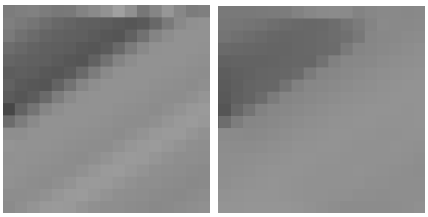

Fig. 4. Left: Original intra prediction. Right: Intra prediction after ADMM filtering.

only squared. In order to better capture objects in the center of blocks, instead of the binary partitioning the so-called Multi-Type-Tree (MTT) partitioning ([36]) is used here. In addition to quad-tree splitting and binary vertical and horizontal splitting, MTT introduces horizontal and vertical center-side triple-tree partitionings as depicted in Fig. 3. All other non-HEVC tools in [34] are turned off.

The filter is tested at coding unit (CU) level where the shape of the block depends on the outcome of the MTT block partitioning. It is applied on the luminance signal and uses the provided intra or inter prediction as described above. Parameter ℓ denoting the required width of the outer reconstructed signal is set to one. For more details on how the the MTT partitioning can influence gains and complexity of a filter method we refer to [20]. The choice of the required parameters for the ADMM filter is described in section V. In contrast to the diffusion filter, the ADMM filter tests only one filter configuration on the CU block which is to be en- or disabled. The information whether the ADMM filter is applied or not is sent at CU level. The filtering is restricted to larger blocks, i.e., it is not applied for blocks with an area ≤ 32.

B. Example

Fig. 4 demonstrates the impact of the suggested ADMM filter for an angular intra prediction block. On the left hand side, the original intra prediction is depicted. On the right, the result of applying the ADMM filter is shown. The ADMM filter smooths and denoises the upper area and the bottom left corner of the prediction block while the edges of the underlying prediction remain sharp.

C. Alternating Direction Method of Multipliers

There exist several optimization algorithms for convex problems in discrete spaces. The Alternating Direction Method of Multipliers is a fusion of dual ascent, the augmented Lagrangian method and the method of multipliers which combines their advantages: It is intended to blend the decomposability of the dual ascent with the convergence properties of the method of multipliers. More details on this can be found in [18].

In general, ADMM can be used to solve problems of the following form

$$\begin{aligned} &\min F(u) + G(z) \\ &\text{subject to } Au + Bz = c, \end{aligned} \tag{8}$$

where $u \in \mathbb{R}^o$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{w \times o}$, $B \in \mathbb{R}^{w \times m}$ and $c \in \mathbb{R}^w$. It is assumed here that F and G are convex. It can be seen that a splitting of variables took place, i.e., the objective function was separated across variables u and z. The associated augmented Lagrangian reads as

$$\mathcal{L}(u, z, v) = F(u) + G(z) + \langle v, Au + Bz + c \rangle + \frac{\rho}{2}|Au + Bz - c|^2.$$

Then, ADMM consists of the following iteration steps

$$u_{k+1} := \underset{u}{\operatorname{argmin}} \mathcal{L}_\rho(u, z_k, v_k) \tag{9}$$

$$z_{k+1} := \underset{z}{\operatorname{argmin}} \mathcal{L}_\rho(u_{k+1}, z, v_k) \tag{10}$$

$$v_{k+1} := v_k + \rho(Au_{k+1} + Bz_{k+1} - c), \tag{11}$$

where $\rho > 0$. Clearly, ADMM features the main steps of dual ascent and the method of multipliers: it consists of an u-minimizing step (9), a z-minimizing step (10) of the Lagrangian and an update step of the Lagrange multiplier (11). Similarly as in the method of multipliers, the step size in eq. (11) is chosen as penalty parameter ρ. The name "alternating direction" stems from the fact that the variables u and z are updated in an alternating fashion.

To make use of the Alternating Direction Method of Multipliers (ADMM), one recasts the optimization problem eq. (6) onto

$$\begin{aligned} &\underset{u}{\operatorname{argmin}} \{\frac{1}{2}|Sz - f_{|\Gamma}|^2 + \alpha \mathcal{J}(u)\} \\ &\text{subject to } u = z. \end{aligned} \tag{12}$$

The associated augmented Lagrangian (in scaled form, see [18]) reads as follows

$$\mathcal{L}_\rho(u, z) = \frac{1}{2}|Sz - f_{|\Gamma}|^2 + \alpha \mathcal{J}(u) + \frac{\rho}{2}\{|u - z + v|^2 - |v|^2\},$$

where penalty parameter $\rho \in \mathbb{R}, \rho > 0$ and υ is the Lagrange multiplier corresponding to constraint $u = z$. To solve eq. (12),, one aims to find $\underset{u,z}{\operatorname{argmin}}\mathcal{L}_\rho(u, z)$. To do so, a splitting of the variables u, z is performed and the minimization problem is separated into

$$\min_u\{\alpha\mathcal{J}(u) + \frac{\rho}{2}|u - z + \upsilon|^2\}, \tag{13}$$

and

$$\min_z\{\frac{1}{2}|Sz - f_{|\Gamma}|^2 + \frac{\rho}{2}|u - z + \upsilon|^2\}. \tag{14}$$

To solve eq. (14),, calculate the derivative with respect to z and set it to zero

$$0 \overset{!}{=} S^*(Sz - f_{|\Gamma}) - \rho(u - z + \upsilon),$$

$$\text{which leads to} z \overset{!}{=} (S^*S + \rho I)^{-1}[S^* f_{|\Gamma} + \rho(u + \upsilon)].$$

Here, S^* denotes the adjoint of the subsampling matrix S and I denotes the identity matrix. Because of the special structure of S, the matrix $(S^*S + \rho I)$ is invertible (more details see subsection IV-E). This is used in algorithm 1 to calculate z_k. The extended prediction block f is used as initialization for z_0. In algorithm 1, the Lagrange multiplier υ is updated into the direction of steepest ascent $u - z$, and ρ serves as step size.

Algorithm 1 Alternating Direction Method of Multipliers

Input:

f	Extended prediction block
$\alpha \geq 0$	Penalty weight
S	Subsampling mask

Output:

$u_{K_{ADMM}}$	Approximate minimizer

procedure ADMM(f,α,S)
  $\rho \leftarrow 0.5$, $z_0 \leftarrow f$, $v_0 \leftarrow 0$, $\varrho_0 \leftarrow 0$, $K_{ADMM} \leftarrow 5$ ▷ Initialization
  for $k = 1 : K_{ADMM}$ **do**
    $u_k, \varrho_k \leftarrow \text{prox}^{\mathcal{J}}_{\alpha/\rho}(z_{k-1} - v_{k-1})$
    ▷ Apply algorithm 2 for $y := z_{k-1} - v_{k-1}$, $\beta := \frac{\alpha}{\rho}$, $\varrho = \varrho_{k-1}$
    $z_k \leftarrow (S^*S + \rho I)^{-1}[S^* f_{|\Gamma} + \rho(u_k + v_{k-1})]$
    $v_k \leftarrow v_{k-1} + \rho(u_k - z_k)$
  end for
  return $u_{K_{ADMM}}$
end procedure

D. Gradient Projection Method

To numerically solve eq. (13) as was done in [16], the problem is reformulated as

$$\text{prox}^{\mathcal{J}}_\beta(y) := \underset{u}{\operatorname{argmin}}\{\frac{1}{2}|u - y|^2 + \beta\mathcal{J}(u)\}, \tag{15}$$

where parameter $\beta \in \mathbb{R}, \beta > 0$. The operator defined in eq. (15) is called proximal operator [18].

In algorithm 1, it is required to solve eq. (15) for u_k. Eq. (4) can be dualized such that

$$\mathcal{J}(u) = \sum_n |D_n\nabla u(p_n)| = \sup_{\varrho\in\mathbb{U}}\langle -\text{div}D^*\varrho,\, u\rangle, \tag{16}$$

where $\mathbb{U} := \{\varrho \in (\mathbb{R}^2)^{nrP} \mid \varrho_n \in \mathbb{R}^2, |\varrho_n| \leq 1\}$ denotes the unit ball with $nrP = (N + \ell) \cdot (W + \ell)$ denoting the number of points in a block. This can be easily understood in the 1D case, where $|\upsilon| = \upsilon \cdot \varrho$ for $\varrho = \text{sign}(\upsilon)$ is equivalent to $|\upsilon| = \sup_{\varrho\in[-1,1]} \varrho \cdot \upsilon$ for some $\upsilon \in \mathbb{R}$. In eq. (16), the adjoint of $D = (D_n)_n$ denoted by $D^* = (D_n^*)_n$ and div is to be understood pointwise. Substituting eq. (16) into eq. (15), the latter can be rewritten as

$$\min_u \sup_{\varrho\in\mathbb{U}}\{\frac{1}{2}|u - y|^2 + \beta\langle -\text{div}D^*\varrho,\, u\rangle\} \tag{17}$$

As the function is convex in u and concave in ϱ, the order of the minimum and supremum can be exchanged ([37, Corollary 37.3.2]). Therefore, it holds

$$\sup_{\varrho\in\mathbb{U}} \min_u\{\frac{1}{2}|u - y|^2 + \beta\langle -\text{div}D^*\varrho,\, u\rangle\}. \tag{18}$$

To solve the inner minimization, eq. (18) is to be derived with respect to u and set to zero. This leads to

$$u_* = y + \beta\text{div}D^*\varrho,$$

which solves the inner minimization of eq. (18). To find the suprema, the gradient of

$$\sup_{\varrho\in\mathbb{U}}\{\frac{1}{2}|u_* - y|^2 + \beta\langle -\text{div}D^*\varrho,\, u_*\rangle\}$$

with respect to ϱ is computed. This leads to the derivation term $\beta D\nabla(u_*)$ which denotes the steepest ascent of ρ. Thus, ϱ is updated in every iteration step by

$$\varrho_k = \mathbf{P}_{\mathbb{U}}(\varrho_{k-1} + s\beta D\nabla(u_*)),$$

where $s > 0$ serves as step size and $\mathbf{P}_{\mathbb{U}}$ denotes the orthogonal projection onto the unit ball. Eventually, eq. (15) is solved by iteratively computing the dual variable

$$\varrho_k = \mathbf{P}_{\mathbb{U}}(\varrho_{k-1} + s\beta D\nabla(y + \beta\text{div}D^*\varrho_{k-1}))$$

inserting the iterative version of u_*. Then, after K_{prox} iterations u is approximated by

$$u_{K_{prox}} = y + \beta\text{div}D^*\varrho_{K_{prox}}.$$

This is realized in algorithm 2.

E. Finite Discretization

In a discrete setting, as mentioned in the beginning, the directional total variation eq. (4) coincides with

$$\mathcal{J}(u) = \sum_n |D_n\nabla u(p_n)|,$$

where

$$D_n = I - \xi_n\xi_n^T \text{ and } \xi_n = \frac{\nabla f(p_n)}{|\nabla f(p_n)|_\eta},$$

Algorithm 2 Gradient Projection Method

Input:

y	Proximal point
$\beta \geq 0$	Regularization parameter
D	Anisotropic diffusion tensor
ϱ	Dual variable

Output:

$u_{K_{prox}}$	Approximate minimizer
$\varrho_{K_{prox}}$	Dual variable

procedure PROX(y,β,D,ϱ)
  $s \leftarrow 1/(8\beta^2)$ ▷ Initialization of step size
  $\varrho_0 \leftarrow \varrho$, $K_{prox} \leftarrow 2$
  for $k = 1 : K_{prox}$ **do**
    $g_k \leftarrow \beta D\nabla(y + \beta \text{ div } D^*\varrho_{k-1})$ ▷ Gradient
    $\varrho_k \leftarrow \mathbf{P}_{\mathbb{U}}(\varrho_{k-1} + sg_k)$ ▷ Update dual variable
  end for
  $u_{K_{prox}} \leftarrow y + \beta \text{ div } D^*\varrho_{K_{prox}}$ ▷ Approx. minimizer
end procedure

where p_n ranges over the number of image points and $\nabla f(p_n) \in \mathbb{R}^2$, $|\cdot|_\eta^2 = |\cdot|^2 + \eta^2$ and $I \in \mathbb{R}^{2\times 2}$ denotes the identity. Now, recall the definition for forward and backward differences,

$$\mathcal{D}_{x_i}^+ u(p_n, t) := u(p_n + he_i, t) - u(p_n, t) \text{ (forward difference)}$$

and

$$\mathcal{D}_{x_i}^- u(p_n, t) := u(p_n, t) - u(p_n - he_i, t) \text{ (backward difference)}$$

for a sample $p_n = (x_1, x_2)^T \in \mathbb{R}^2$, $t \in \mathbb{R}$, parameter $h > 0$ and unit vector $e_i \in \mathbb{R}^2$.

Discretize $\nabla f \nabla f^T$ with backward differences $\mathcal{D}_{x_1}^- f$ and $\mathcal{D}_{x_2}^- f$. On the top respectively left border where the reconstructed samples end, a mirrored (Neumann) boundary condition is used. Then, the gradient matrix $\nabla f \nabla f^T$ can be calculated

$$\nabla f \nabla f^T = \begin{pmatrix} (\mathcal{D}_{x_1}^- f)^2 & \mathcal{D}_{x_1}^- f \mathcal{D}_{x_2}^- f \\ \mathcal{D}_{x_1}^- f \mathcal{D}_{x_2}^- f & (\mathcal{D}_{x_2}^- f)^2 \end{pmatrix}.$$

For every p_n inside the extended prediction block, there is a anisotropic diffusion tensor D_n such that

$$D_n = \frac{1}{(\mathcal{D}_{x_1}^- f)^2 + (\mathcal{D}_{x_2}^- f)^2 + \eta^2} \times \begin{pmatrix} 1 - (\mathcal{D}_{x_1}^- f)^2 & \mathcal{D}_{x_1}^- f \mathcal{D}_{x_2}^- f \\ \mathcal{D}_{x_1}^- f \mathcal{D}_{x_2}^- f & 1 - (\mathcal{D}_{x_2}^- f)^2 \end{pmatrix}$$

Since D_n is real and symmetric, it holds that $D_n = D_n^*$, i.e., D_n is self adjoint. After applying D^* to the current dual variable ϱ_{k-1}, its divergence has to be calculated. This is realized using forward differences, i.e.,

$$\text{div} = \mathcal{D}_{x_1}^+ + \mathcal{D}_{x_2}^+,$$

using a mirrored boundary condition on the right respectively lower boundary.

To solve eq. (14) discretely, one has to take a closer look at the specific structure of mask S. By definition, S restricts the domain of function u to the reconstructed area, i.e. it holds that $Su = u_{|\Gamma}$. Mask S is a sparse matrix of dimension $nrRecBdry \times nrP$ where $nrRecBdry = N + \ell + W + \ell - 1$ that possesses unit rows corresponding to the reconstructed boundary indices. The matrix is sparse and is zero everywhere except on the diagonal. The matrix S^* is of dimension $nrP \times nrRecBdry$. The matrix S^*S is of dimension $nrP \times nrP$. It is a sparse matrix with unit rows at the indices corresponding to the reconstructed boundary and zero rows otherwise. Since S^*S is a diagonal matrix, the term $(S^*S + \rho I)$ is invertible. Because of its special structure, multiplying by the term $(S^*S + \rho I)^{-1}$ leads to dividing the inner points of the block by ρ and the top and left boundary by $1 + \rho$. This way, $(S^*S + \rho I)^{-1}$ can be implemented efficiently. Applying S^* to $f_{|\Gamma}$ leads to an array of dimension nrP which is zero everywhere except at the indices corresponding to the reconstructed boundary.

V. Parameter Tests

In the numerical implementation of the ADMM filter, several parameters have to be chosen. It will be shown that the right choice of parameter (sets) have a strong influence on the RD gains of the filter. The parameters to be chosen consist of the number of iterations of the gradient projection method K_{prox} in algorithm 2, the number of iterations of the ADMM solving method K_{ADMM} in algorithm 1, penalty weight $\alpha > 0$, parameter $\rho > 0$ in the Lagrangian and edge parameter $\eta \geq 0$ related to the weighting of the gradient directions in penalty functional $\mathcal{J}$. Since testing the complete parameter space would result in a prohibitive amount of computations, the iteration parameters K_{prox} and K_{ADMM} were tested separately while the other parameters were set fixed. Setting $K_{prox} = 2$ and $K_{ADMM} = 5$ lead to a reasonable trade-off between RD gain and complexity [38].

Furthermore, the setting of penalty weight α in terms of block sizes was investigated. Since the penalty weight α is linked to the stepsize $s = \frac{1}{8\beta^2} = \frac{\rho^2}{8\alpha^2}$ in the computation of the proximal operator, choosing it too large will result in a very slow convergence, choosing it too small will make the method instable. Fig. 5 shows the number of blocks where weight α was chosen on the z-axis. The corresponding choice of the weight α is depicted on the x-axis, distinguished by the size of the quadratic blocks on the y-axis of height respectively width $\{8, 16, 32\}$. To collect the data for this figure, an encode-only test was set up where different weights α were tested and the best choice was saved. The costs did not include a signaling for the weight α. As can be observed in Fig. 5, the weight α was mostly chosen as $\alpha = 0.01$ for CU blocks of size 8 and as $\alpha = 0.05$ for larger block sizes in the setting of HEVC partitioning. This choice serves as a base for the following parameter tests.

Edge parameter η influences the effect of prior

$$\mathcal{J}(u) = \sum_n |D_n \nabla u(p_n)| = \sum_n |\nabla u(p_n) - \langle \xi_n, \nabla u(p_n) \rangle \xi_n|,$$

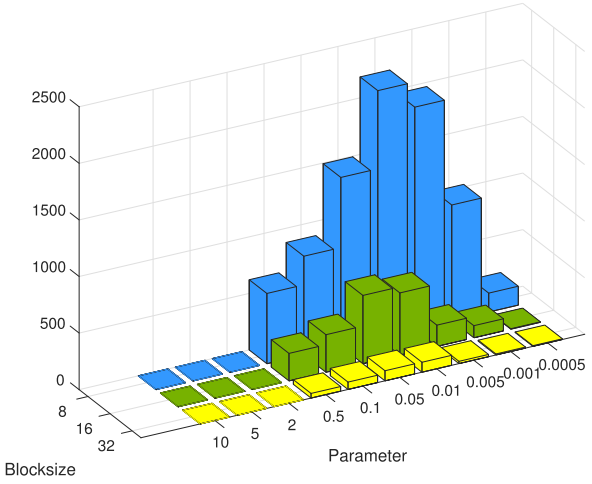

Fig. 5. Count of the chosen penalty weight $\alpha \in [0.0005, \ldots, 10]$ on the z-axis for quadratic blocks of size *Blocksize*, sequence ParkScene, QP22.

since

$$D_n = I - \xi_n \xi_n^T \text{ and } \xi_n = \frac{\nabla f(p_n)}{|\nabla f(p_n)|_\eta},$$

where $|\cdot|_\eta^2 = |\cdot|^2 + \eta^2$, $\eta \geq 0$ and $I \in \mathbb{R}^{2\times 2}$ denotes the identity. Recall that functional $\mathcal{J}$ serves as a penalty term, i.e. if $\mathcal{J}(u)$ is large, u is unlikely to be chosen as a solution.

Since there is no straightforward way of how to choose the parameter space (α, η, ρ), the parameters were empirically tested [38] in a software based on HEVC with additional QTBT block structure [34]. Penalty weight α was tested to be $\alpha \in \{0.005, 0.01, 0.05\}$, edge parameter $\eta \in \{0, 0.01, 1, 2\}$ and parameter $\rho \in \{0.1, 0.3, 0.5\}$. In the encoder all resulting combinations (α, η, ρ) in the parameter space were simulated. In the cases where the ADMM filter improved the original, the RD cost of the original (without ADMM filter) and the improved cost with ADMM filter were saved. Then, the cases where the ADMM filter was chosen using a specific parameter combination (number of occurence of the parameter combination) was saved together with the corresponding block size. No additional costs for the parameters were simulated.

To evaluate this encode-only test, the chosen CU blocks were separated into those that had at least one side equal to 8 and those with larger sides. Note that the ADMM filter (taken the diffusion filter as a model) was configured to be only applied to CU blocks of area larger than 32, thus cases of width and height equal to 4 were not considered. A finer separation into block size types or QPs did not prove to be effective in the sense of RD improvements. Furthermore, the cases were separated into intra and inter blocks. For each class of block type, the occurrence of a certain parameter combination $(\alpha_i, \eta_j, \rho_k)$ for $i, k \in \{1, 2, 3\}$, $j \in \{1, 2, 3, 4\}$ was multiplied by the corresponding percentage in cost improvement. Cost improvement is referring here to the difference between the cost of the unfiltered original and the ADMM filtered case. The multiplication of the number of occurences of a specific parameter combination and the cost improvement percentage is referred to as "improvement weight" in the following.

TABLE I

BEST PARAMETER COMBINATIONS FOR INTRA BLOCKS WITH BOTH SIDES LARGER THAN 8

Parameter Combination (α, η, ρ)	Improvement Weight
$(0.005, 1, 0.5)$	1714463
$(0.005, 0, 0.1)$	1575839
$(0.005, 0.1, 0.1)$	1311580
$(0.005, 1, 0.1)$	1159914
$(0.005, 1, 0.3)$	891617

TABLE II

BEST PARAMETER COMBINATIONS FOR INTRA BLOCKS WITH AT LEAST ONE SIDE EQUAL TO 8

Parameter Combination (α, η, ρ)	Improvement Weight
$(0.005, 1, 0.5)$	2404376
$(0.005, 0, 0.1)$	1697149
$(0.005, 0.01, 0.1)$	1376329

TABLE III

ALL INTRA COMPARISON OF $(\alpha, \eta, \rho) = (0.01, 1, 0.3)$ FOR ALL BLOCK TYPES (Y LEFT HAND SIDE) AND AS IN EQ. 19 (Y_{impr} RIGHT HAND SIDE), 1 FRAME, $QP \in \{22, 27, 32, 37\}$, MEASURED IN BD RATE

Resolution	Sequence name	Y	Y_{impr}
4K Panorama	ChairliftRide	-0.65%	-0.86%
4K HDR	PeopleInShoppingCenter	-0.72%	-0.86%
4K UHD	Nebuta	-0.60%	-0.69%
	Drums100	-1.48%	-1.49%
	Tango	-2.64%	-2.79%
	Rollercoaster	-1.53%	-1.62%
	Crosswalk	-2.52%	-2.67%
	FoodMarket	-4.57%	-4.60%
HD	Cactus	-0.57%	-0.68%
	Overall	-1.70%	-1.81%

The idea is here that the larger the weight, the larger its presumable improvement impact is on the test set.

Table I shows the best five parameter combinations (i.e., the one with the largest improvement weights) for intra blocks with both sides larger than 8 sorted by improvement weight. Here, the four best parameter combinations seem to be quite similar with respect to their improvement weights. Empirical tests lead to the choice of combination $(0.005, 1, 0.1)$ which is highlighted in gray. Table II shows the best three parameter combinations for intra blocks of at least one side equal to 8 sorted by improvement weight. Here, the results indicate that the best case here by far is the parameter combination $(0.005, 1, 0.5)$ (highlighted in gray). As empirical tests confirmed this, the parameters for intra blocks were set as

$$(\alpha, \eta, \rho)_{\text{intra}} = \begin{cases} (0.005, 1, 0.5) & \text{if one block size } = 8 \\ (0.005, 1, 0.1) & \text{else.} \end{cases} \tag{19}$$

In Table III, a comparison of tests with two different parameter combinations are shown in case of an AI configuration for one frame. The corresponding configuration files are taken from [39]. The results in the left columns were obtained choosing $(\alpha, \eta, \rho) = (0.01, 1, 0.3)$ for all block types. The results in the right columns were obtained choosing the parameter

TABLE IV

BEST PARAMETER COMBINATIONS FOR INTER BLOCKS WITH BOTH SIDES LARGER THAN 8

Parameter Combination (α, η, ρ)	Improvement Weight
$(0.005, 1, 0.5)$	556406
$(0.005, 2, 0.5)$	349788
$(0.005, 0.01, 0.5)$	132608

TABLE V

BEST PARAMETER COMBINATIONS FOR INTER BLOCKS WITH AT LEAST ONE SIDE EQUAL TO 8

Parameter Combination (α, η, ρ)	Improvement Weight
$(0.005, 1, 0.5)$	606032
$(0.005, 2, 0.5)$	401423
$(0.005, 0.01, 0.5)$	155779

TABLE VI

RANDOM ACCESS COMPARISON OF $(\alpha, \eta, \rho) = (0.01, 1, 0.3)$ FOR ALL BLOCK TYPES (Y LEFT HAND SIDE) AND AS IN EQ. 20 (Y_{impr} RIGHT HAND SIDE), 17 FRAMES, $QP \in \{22, 27, 32, 37\}$, MEASURED IN BD RATE

Resolution	Sequence name	Y	Y_{impr}
4K Panorama	Trolley	-0.30%	-0.59%
4K HDR	SunsetBeach	-0.43%	-0.75%
4K UHD	Nebuta	-0.91%	-2.72%
	Drums100	-0.97%	-1.40%
	Tango	-1.65%	-2.00%
	Rollercoaster	-1.32%	-1.15%
	Crosswalk	-2.03%	-1.14%
	FoodMarket	-2.03%	-2.43%
HD	BQTerrace	-0.17%	-0.34%
Overall		-1.09%	-1.39%

combinations as in eq. 19 resulting from the parameter tests described above. It can be seen that the results are improved by overall 0.11%.

In tables IV and V, the best parameter combinations in terms of improvement weight for RA and inter blocks are depicted. Here, both cases seem to indicate that the parameter combination $(\alpha, \eta, \rho) = (0.005, 1, 0.5)$ obtains the best results and is therefore highlighted in gray.

In Table VI, the results for choosing $(\alpha, \eta, \rho) = (0.01, 1, 0.3)$ for all block types is compared to choosing eq. 19 for intra blocks and $(0.005, 1, 0.5)$ for all inter blocks, i.e.,

$$(\alpha, \eta, \rho) = \begin{cases} (0.005, 1, 0.5) & \text{inter block,} \\ (0.005, 1, 0.5) & \text{intra block, if one block size } = 8 \\ (0.005, 1, 0.1) & \text{intra block, else.} \end{cases} \tag{20}$$

As can be seen here, the choice of the right parameter set improves the results by overall by 0.3%.

VI. RESULTS

In Table VII, Table VIII, Table IX and Table X the bitrate savings for the ADMM filter in All Intra and Random Access configuration are depicted for full sequences for $QP \in \{22, 27, 32, 37\}$ and for $QP \in \{27, 32, 37, 42\}$. The tool has been implemented into a software based on HEVC [34] and configured using configuation files found in [39]. In particular, QTBT with MTT block structure is turned on and large CTUs of size 128 are used. In case of Intra, there exists an implicit split such that the largest CUs are of size 64. The minimal CU size is 8×8 and the minimal TU size is 4×4.

It can be observed in tables VII, VIII, IX and X that the ADMM filter works best for sequences of high resolution such as UHD sequences. For AI, the RD gains are almost the same

TABLE VII

ALL INTRA, FULL SEQUENCES, $QP \in \{22, 27, 32, 37\}$, MEASURED IN BD RATE WITH ADMM FILTER

Resolution	Sequence name	$QP \in \{22, 27, 32, 37\}$		
		Y	U	V
4K Panorama	ChairliftRide	-0.82%	-1.32%	-1.99%
4K HDR	PeopleInShoppingCenter	-0.90%	-1.04%	-1.26%
4K UHD	Nebuta	-0.83%	-0.68%	-0.64%
	Drums100	-1.48%	-1.82%	-1.71%
	Tango	-2.52%	-2.63%	-1.74%
	Rollercoaster	-2.04%	-1.49%	-1.81%
	Crosswalk	-3.01%	-1.88%	-2.00%
	FoodMarket	-1.51%	-1.23%	-1.38%
HD	Cactus	-0.73%	-0.77%	-1.20%
Overall		-1.54%	-1.43 %	-1.53%
Enc Time		187%		
Dec Time		184%		

TABLE VIII

ALL INTRA, FULL SEQUENCES, $QP \in \{27, 32, 37, 42\}$, MEASURED IN BD RATE WITH ADMM FILTER

Resolution	Sequence name	$QP \in \{27, 32, 37, 42\}$		
		Y	U	V
4K Panorama	ChairliftRide	-0.82%	-1.74%	-2.94%
4K HDR	PeopleInShoppingCenter	-0.69%	-1.43%	-1.73%
4K UHD	Nebuta	-0.93%	-1.22%	-1.26%
	Drums100	-1.47%	-2.16%	-2.14%
	Tango	-2.69%	-3.02%	-1.72%
	Rollercoaster	-2.12%	-1.80%	-2.48%
	Crosswalk	-3.24%	-2.34%	-2.39%
	FoodMarket	-1.62%	-1.62%	-1.85%
HD	Cactus	-0.64%	-1.27%	-1.52%
Overall		-1.58%	-1.85 %	-2.00%
Enc Time		190%		
Dec Time		198%		

TABLE IX

RANDOM ACCESS, FULL SEQUENCES, $QP \in \{22, 27, 32, 37\}$, MEASURED IN BD RATE WITH ADMM FILTER

Resolution	Sequence name	$QP \in \{22, 27, 32, 37\}$		
		Y	U	V
4K Panorama	Trolley	-0.39%	-1.07%	-1.95%
4K HDR	SunsetBeach	-0.78%	-1.51%	-3.41%
4K UHD	Nebuta	-3.66%	-6.72%	-7.84%
	Drums100	-1.20%	-1.93%	-1.62%
	Tango	-1.87%	-4.38%	-2.94%
	Rollercoaster	-1.14%	-1.35%	-1.69%
	Crosswalk	-1.70%	-1.97%	-2.25%
	FoodMarket	-1.05%	-1.48%	-1.74%
HD	BQTerrace	-0.34%	-1.18%	-1.71%
Overall		-1.35%	-2.40 %	-2.79%
Enc Time		128%		
Dec Time		138%		

TABLE X

RANDOM ACCESS, FULL SEQUENCES, $QP \in \{27, 32, 37, 42\}$, MEASURED IN BD RATE WITH ADMM FILTER

Resolution	Sequence name	$QP \in \{27, 32, 37, 42\}$		
		Y	U	V
4K Panorama	Trolley	-0.44%	-1.51%	-2.43%
4K HDR	SunsetBeach	-0.92%	-2.14%	-4.79%
4K UHD	Nebuta	-4.21%	-6.59%	-7.35%
	Drums100	-0.83%	-1.80%	-1.73%
	Tango	-1.09%	-2.67%	-1.16%
	Rollercoaster	-0.91%	-1.41%	-1.82%
	Crosswalk	-1.32%	-1.84%	-1.51%
	FoodMarket	-0.89%	-1.69%	-1.94%
HD	BQTerrace	-0.18%	-1.17%	-1.51%
Overall		-1.20%	-2.31 %	-2.69%
Enc Time		129%		
Dec Time		131%		

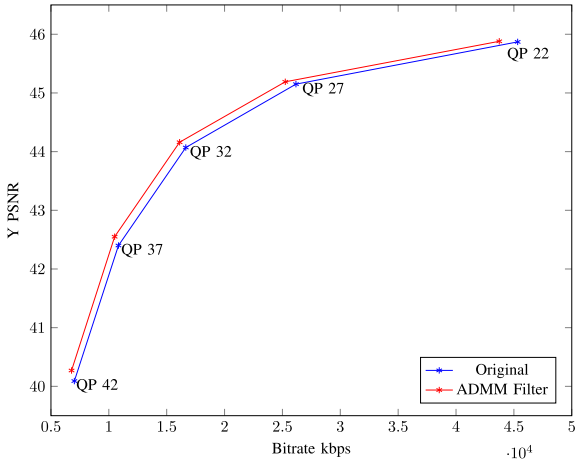

Fig. 6. RD Plot for test sequence Rollercoaster.

for $QP \in \{22, 27, 32, 37\}$ and for $QP \in \{27, 32, 37, 42\}$. In case of RA, the results for $QP \in \{22, 27, 32, 37\}$ are slightly better.

In Fig. 6 the rate-distortion plots for test sequence Rollercoaster for five QPs 22, 27, 32, 37 and 42 are depicted. There is a visible distance for all five QPs which corresponds to the considerable RD gain that was achieved.

VII. CONCLUSION

In order to construct a prediction filter mathematical techniques originally used for inpainting were applied using a variational approach. Thereby, the structures of the initial prediction provided by the video codec could be incorporated into the filter construction which resulted in a signal adaptive tool. The established optimization problem was solved using the so-called Alternating Direction Method of Multipliers (ADMM). A motivation for the selected solving method was given and its discretization was described in detail. Using a sequence of sophisticated parameter tests involving the magnitude of cost improvement, a suitable parameter set for the ADMM filter was determined which improved the bitrate savings substantially. The ADMM filter was embedded into a software based on HEVC with additional QTBT with MTT block structure and selected in a block-wise manner.

Overall, the ADMM filter method obtained bitrate savings of 1.54% for AI for 87% encoding and 84% decoding complexity increase. In case of RA, bitrate savings of 1.35% were accomplished at a encoding complexity increase of 28% and decoding complexity increase of 38%. For certain UHD sequences, it achieved bitrate savings of up to 3.01% for AI and 3.66% for RA.

REFERENCES

[1] 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, Dec. 2012.

[2] J. Rasch *et al.*, "A signal adaptive diffusion filter for video coding using directional total variation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 2570–2574.

[3] Y. Zhang and Y. Lin, "Improving HEVC intra prediction with PDE-based inpainting," in *Proc. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA), Asia–Pacific*, Dec. 2014, pp. 1–5.

[4] D. Doshkov, P. Ndjiki-Nya, H. Lakshman, M. Koppel, and T. Wiegand, "Towards efficient intra prediction based on image inpainting methods," in *Proc. 28th Picture Coding Symp.*, Dec. 2010, pp. 470–473.

[5] J. Lainema and W. Han, "Intra-picture prediction in HEVC," in *High Efficiency Video Coding (HEVC)*, V. Sze, M. Budagavi, and G. J. Sullivan, Eds. Springer, 2014, pp. 91–112.

[6] A. Minezawa, K. Sugimoto, and S. Sekiguchi, *An Improved Intra Vertical and Horizontal Prediction*, document JCTVC-F172. Torino: Joint Collaborative Team on Video Coding (JCT-VC), Jul. 2011.

[7] J. Pfaff *et al.*, "Video compression using generalized binary partitioning, trellis coded quantization, perceptually optimized encoding, and advanced prediction and transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1281–1295, May 2020.

[8] B. Bross, P. Helle, H. Lakshman, and K. Ugur, "Inter-Picture Prediction in HEVC," in *High Efficiency Video Coding (HEVC)*, V. Sze, M. Budagavi, and G. J. Sullivan, Eds. Springer, 2014, pp. 113–140.

[9] T. Wedi, "Adaptive interpolation filters and high-resolution displacements for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 484–491, Apr. 2006.

[10] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose, "Global brightness-variation compensation for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 988–1000, Dec. 1998.

[11] J. López, J. H. Kim, A. Ortega, and G. Chen, "Block-based illumination compensation and search techniques for multiview video coding," in *Proc. Picture Coding Symp.*, 2004, pp. 1–6.

[12] J. Rasch *et al.*, "A signal adaptive diffusion filter for video coding," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 131–133.

[13] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D: Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, Nov. 1992.

[14] J. Shen and T. F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, Jan. 2002.

[15] M. Burger, L. He, and C.-B. Schönlieb, "Cahn–Hilliard inpainting and a generalization for grayvalue images," *SIAM J. Imag. Sci.*, vol. 2, no. 4, pp. 1129–1167, Nov. 2009.

[16] M. J. Ehrhardt and M. M. Betcke, "Multicontrast MRI reconstruction with structure-guided total variation," *SIAM J. Imag. Sci.*, vol. 9, no. 3, pp. 1084–1106, Jan. 2016.

[17] L. Bungert, D. A. Coomes, M. J. Ehrhardt, J. Rasch, R. Reisenhofer, and C.-B. Schönlieb, "Blind image fusion for hyperspectral imaging with the directional total variation," *Inverse Problems*, vol. 34, no. 4, Apr. 2018, Art. no. 044003.

[18] S. N. E. P. B. Boyd Parikh Chu and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–112, 2010.

[19] J. Rasch *et al.*, "A signal adaptive diffusion filter for video coding: Improved parameter selection," *APSIPA Trans. Signal Inf. Process.*, vol. 8, 2019.

[20] J. Rasch *et al.*, "A signal adaptive diffusion filter for video coding: Mathematical framework and complexity reductions," *Signal Process., Image Commun.*, vol. 85, Jul. 2020, Art. no. 115861.

[21] J. Rasch, J. Pfaff, H. Schwarz, D. Marpe, and T. Wiegand, "Comparison between the diffusion and the ADMM filter and combined results," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, Nov. 2019, pp. 1–5.
[22] G. Bjøntegaard, *Calculation of Average PSNR Differences between RD-Curves*, document VCEG-M33, ITU-T SG16/Q6, 2001.
[23] T. Wiegand and H. Schwarz, *Video Coding: Part II Fundamentals Source Video Coding Foundations and Trends in Signal Processing*, vol. 10. Now Foundations and Trends, 2016.
[24] A. N. Netravali and J. O. Limb, "Picture Coding: A Review," *Proc. IEEE*, vol. 68, no. 3, pp. 366–412, 1980.
[25] *Reference Software for High Efficiency Video Coding*, ITU-T and ISO/IEC, 2014. [Online]. Available: https://hevc.hhi.fraunhofer.de/trac/hevc/browser
[26] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.
[27] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
[28] R. Adams and J. Fournier, *Sobolev Spaces* (Pure and Applied Mathematics), no. 140. New York, NY, USA: Academic, 2003.
[29] V. Caselles, A. Chambolle, and M. Novaga, "The discontinuity set of solutions of the TV denoising problem and some extensions," *Multiscale Model. Simul.*, vol. 6, no. 3, pp. 879–894, Jan. 2007.
[30] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock. (2009). *An Introduction to Total Variation for Image Analysis*. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00437581
[31] L. C. Evans, *Partial Differential Equations* (Graduate Studies in Mathematics). Providence, RI, USA: American Mathematical Society, 2010, vol. 19.
[32] A. M. Khludnev and J. Sokolowski, *Model. Control Solid Mech.* (International Series of Numerical Mathematics), vol. 122. Basel, Switzerland: Birkhäuser, 1997.
[33] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Effective quadtree plus binary tree block partition decision for future video coding," in *Proc. Data Compress. Conf. (DCC)*, Apr. 2017, pp. 23–32.
[34] A. Wieckowski *et al.*, "NextSoftware: An alternative implementation of the Joint Exploration Model (JEM)," 8th Meeting, doxument JVET-H0084. Macao, CN: Joint Video Exploration Team (JVET), Oct. 2017.
[35] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 7 (JEM 7)," 7th Meeting, document JVET-G1001. Torino, IT: Joint Video Exploration Team (JVET), Jul. 2017.
[36] X. Li *et al.*, "Multi-type-tree," 4th Meeting, document JVET-D0117. Chengdu, CN: Joint Video Exploration Team (JVET), Oct. 2016.
[37] R. T. Rockafellar, *Convex Analysis* (Princeton Landmarks in Mathematics and Physics). Princeton, NJ, USA: Princeton Univ. Press, 1970.
[38] V. Warno, "Constrained Optimization Problems using Alternating Direction Methods of Multipliers: Application in Video Coding to improve Intra-Prediction," M.S. thesis, Humboldt Univ. zu Berlin, Fraunhofer Heinrich-Hertz-Inst., Berlin, Germany, 2018.
[39] *Reference Software for Versatile Video Coding*. ITU-T and ISO/IEC, 2018, [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_VVCSoftware_VTM/

Jennifer Rasch received the Diploma degree in mathematics from the Humboldt University of Berlin, Germany, in 2012. From 2014 to 2020, she was as a Research Associate with the Video Coding & Analytics Group, Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, and actively participated in the standardization process of VVC. Her Ph.D. thesis at the Technical University of Berlin was recognized by the prestigious ARD/ZDF "Women & Media Technology" award 2020.

Victor Warno received the M.Sc. degree in mathematics from the Humboldt University of Berlin in 2018. His master thesis was a joint collaboration with Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin. Since 2018, he has been a Software Developer with Senacor Technologies, Berlin.

Jonathan Pfaff received the Diploma and Dr.rer.nat. degrees in mathematics from Bonn University in 2010 and 2012, respectively. After a postdoctoral research stay at Stanford University, he joined the Video Coding & Analytics Department, Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany, in 2015. He has contributed to the efforts of the Joint Video Experts Team (JVET) in developing the versatile video coding standard since 2018.

Caren Tischendorf received the Dipl.-Math. and Dr.rer.nat. degrees from the Humboldt University of Berlin, Germany, in 1992 and 1996, respectively. From 2002 to 2004, she was the Head of the junior research group Numerical Analysis, DFG Research Center MATHEON, Berlin. She was a Professor in Mathematics/Numerical Analysis with the University of Cologne from 2006 to 2012. She is currently a Professor with the Department of Mathematics, Humboldt University of Berlin.

Detlev Marpe (Fellow, IEEE) received the Dipl.-Math. degree (Hons.) from the Technical University of Berlin, Germany, in 1990, and the Dr.-Ing. degree from the University of Rostock, Germany, in 2004. He joined the Fraunhofer Heinrich Hertz Institute, Berlin, in 1999, where he is currently the Head of the Video Coding & Analytics Department and of the Image and Video Coding Research Group. He was a major Technical Contributor to the entire process of the development of the H.264/MPEG-4 Advanced Video Coding (AVC) standard and the H.265/MPEG High Efficiency Video Coding (HEVC) standard.

Heiko Schwarz received the Dipl.-Ing. degree in electrical engineering and the Dr.-Ing. degree from the University of Rostock, Germany, in 1996 and 2000, respectively. In 1999, he joined the Fraunhofer Heinrich Hertz Institute, Berlin, Germany. Since then, he has contributed successfully to the standardization activities of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group. Since 2010, he has been heading the research group "Image and Video Coding" at the Fraunhofer Heinrich Hertz Institute. In October 2017, he became a Professor at FU Berlin.

Thomas Wiegand (Fellow, IEEE) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg in 1995, and the Dr.-Ing. degree from the University of Erlangen-Nuremberg in 2000. He served as a Consultant for several start-up ventures. He has been an active participant in standardization for video coding multimedia with many successful submissions to ITU-T and ISO/IEC. He is an Associated Rapporteur of ITU-T VCEG. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, and is jointly heading the Fraunhofer Heinrich Hertz Institute, Berlin.