A convex optimization framework for video quality and resolution enhancement from multiple descriptions

Andrei Purica, Benoit Boyadjis, Béatrice Pesquet-Popescu, Frédéric Dufaux, Cyril Bergeron

A convex optimization framework for video quality and resolution enhancement from multiple descriptions

Andrei Purica*, Benoît Boyadjis**, Béatrice Pesquet-Popescu*, Frédéric Dufaux‡, Cyril Bergeron*
*LTCI, Telecom Paristech, Universite Paris-Saclay – 46 rue Barrault, Paris, France
* Thales Communications and Security, OPS/HTE/STR/MMP – Gennevilliers, France
‡ L2S, CNRS, CentraleSupelec, Univ. Paris Sud – 3 rue Joliot Curie, Gif-sur-Yvette, France
*first.last@telecom-paristech.fr – *first.last@thalesgroup.com – ‡first.last@l2s.centralesupelec.fr

***Abstract*—Transmission and compression technologies advancement over the past decade led to a shift of multimedia content towards cloud systems. Multiple copies of the same video are available through numerous distribution systems. Different compression levels, algorithms and resolutions are used to match the requirements of particular applications. As 4k display technologies are rapidly adopted, resolution enhancement algorithms are of vital importance. Current solutions do not take into account the particularities of different video encoders, while video reconstruction methods from compressed sources do not provide resolution enhancement. In this paper, we propose a multi source compressed video enhancement framework where each description can have a different compression level and resolution. Using a variational formulation based on a modern proximal dual splitting algorithm, we efficiently combine multiple descriptions of the same video. Two applications are proposed: combining two compressed Low Resolution (LR) descriptions of a video sequence into a High Resolution (HR) description and enhancing a compressed HR video using a LR compressed description. Tests are performed over multiple video sequences encoded with High Efficiency Video Coding (HEVC), at different compression levels and resolutions obtained through multiple down-sampling methods.**

***Index Terms*—video reconstruction, super-resolution, HEVC**

I. Introduction and state of the art

The continuous evolution of transmission systems, storage and video compression technology in the past decade provides the end user with easy access to video content. A varied number of distribution methods exist, from the classical DVD's to online streaming on the world wide web. High resolution, studio quality video sequences are usually down-sampled and compressed in order to match the requirements of certain applications and the limitations imposed by transmission and storage technologies. Large video databases such as YouTube or Netflix provide multiple resolutions and different encodings of the same video in order to account for user bandwidths and displays. This situation creates a lot of potential for resolution enhancement and compression artifact reduction techniques from single and multiple sources.

The ability to efficiently combine multiple descriptions of the same video and exploit the information variety can be a useful tool in several scenarios. Video transmission systems that rely on scalable encoders, such as Scalable High Efficiency Video Coding (SHVC)[1], generate multiple representations of a video sequence at a different quality level or resolution. The information variety between these representations can be used to obtain a higher quality video. Similar scenarios can be encountered when enhancing older videos that are only available in compressed form or when working with MultiView plus Depth (MVD) [2] video transmission systems where the problem of image alignment is achieved through depth computed disparity.

Super-resolution (SR) algorithms are post-processing techniques that infer a spatially High Resolution (HR) estimate from one or more Low Resolution (LR) images. Currently, SR is an active research field; a review of SR algorithms is available in [3], while performance comparisons can be found in [4], [5], [6]. In general, SR algorithms can be divided in single-frame (SF-SR) or multi-frame (MF-SR) approaches. The later exploits the motion between successive LR frames in order to extract unique information from each representation. In the case of 3D video, adjacent views can also be used [7]. The problem formulation in most cases assumes the availability of a high number of descriptions (5-30) which are subjected to different pixel shift operations (rotations, translations), blurring and sub-sampling. Some of the most popular MF-SR algorithms rely on a Bayesian probabilistic formulation and employ various SR priors such as smoothness with Total Variation (TV) [8] or the Simultaneous Auto Regressive (SAR) image model [9], l_1 based priors [10] or non-stationary image prior combinations [11]. This type of MF-SR approaches is best suited to tackle the problem of image acquisition, where a high number of descriptions is available with simple motion and a similar blurring.

In the recent work of Liu and Sun [12], the Bayesian approach is extended to videos. In this scenario, the descriptions are consecutive frames of a video sequence. As noted by the authors, this problem is inherently more difficult as real world videos have complex motion rather than a simple parametric form. The paper proposes a practical SR framework where optical flow [13], blur kernel [14] and noise levels [15] are simultaneously estimated. Gains of up to 3 dBs are reported over bicubic up-sampling, when super-resolving using 15 forward and backward frames. The degradation was synthetically

added as Gaussian blurring, sub-sampling and Gaussian white noise; tests were performed on real world video sequences. However, as reported by the authors it can take 2 hours to super-resolve one frame. In the work of Ma *et al.* [16], motion blur is taken into consideration and improvements over [12] are reported on real world sequences in a similar set-up where 30 frames are used in the computation. Another video SR method that incorporates blur estimation was developed by Faramarzi *et al.*[17]. A sequential approach is used for motion estimation using optical flow between current and past frames while the frame deblurring and blur estimation is performed using an iterative multi-scale approach with a Huber-norm based cost function minimized using the conjugate gradient method.

Segall *et al.* investigate the problem of SR on compressed video in [18]. They show that compression artifacts complicate the SR reconstruction and suggest that a model of compression should be employed. In [19], a Bayesian maximum a posteriori probability formulation is proposed that takes into account the quantization in frequency domain. The method is shown to provide improvements over spatial domain methods on frequency quantized images, when exact motion information is known. In [20], the H.264/AVC compression process is fused with the Bayesian SR approach, and gains ranging from 0.4 to 2.7 dBs over bicubic upsampling are reported. A more recent maximum a posteriori based SR method for compressed video with a multichannel image prior was proposed by Belekos *et al.* in [21]. Wang *et al.* [22] tackle the problem of compressed video enhancement from a different perspective. The authors propose a practical framework that enhances the quality of video by combining different encodings of the same sequence. In this scenario real world videos are encoded using MPEG-2 in two configurations. The proposed algorithm is able to combine the two decoded videos. Gains of up to 1.5 dB are reported, however, no resolution enhancement is performed.

Another class of SR methods is based on learning techniques and two of the most successful approaches are based on either dictionary learning [23] [24] or Convolutional Neural Networks (CNN) [25][26]. Dong *et al.*[27] propose an image super resolution method by coupling the sparse representation model (SRM) [28] with the nonlocal autoregressive model. Each LR patch can be represented using additional information from neighboring patches. In [29] and [30], Dai *et al.* extend the SRM to video and propose a dictionary based learning method that takes into account multiple frames. Optical flow is used to obtain correspondences between up-sampled LR patches in consecutive frames and the dictionary is trained using all corresponding LR patches and the HR patch. In the recent work of Kappeler *et al.* [31] video SR is achieved by means of CNN using 3 or 5 consecutive frames to super resolve the middle one. The authors also extend this approach to compressed video in [32]. They perform an extensive test of SR algorithms on a real world sequence (Myanmar at 960×540 resolution). The LR descriptions are created with ffmpeg and then compressed with MPEG's H.264/AVC encoder (4 different compression levels were tested). The proposed method shows gains of up to 4 dB over bicubic interpolation, albeit the algorithm was trained on the same sequence and 14 hour were needed for 3 frames input training and up to 1 minute to super resolve due to the motion compensation (motion information was computed before training with optical flow).

Nowadays, videos are mainly available in compressed form. Furthermore, different observations of a video sequence are usually available with a different compression level and resolution. To the best of our knowledge, there are no algorithms tackling the problem of super-resolution and compressed video enhancement from multiple descriptions of the same video. In this paper, we build on our previous work [33], and inspired by the method in [22], we propose a practical framework that is able to reconstruct and enhance a video sequence starting from multiple sources. More precisely: 1) we extend our model to take into account multiple videos, 2) we model the down-sampling process to account for any polyphase filter in a manner consistent with our choice of convex optimization method, 3) we integrate temporal prediction in our model 4) we integrate HEVC compression model and finally 5) we implement the framework using a modern and efficient proximal dual-splitting algorithm [34] such that we can combine observations with different compression levels, down-sampling methods or resolutions. Furthermore, this approach provides a great degree of flexibility. The framework can be applied to various applications ranging from image SR to enhancing the quality and suppressing compression artifacts of a video, and even combining multiple video streams, as shown in our proposed applications.

The effectiveness of the method is shown on multiple video sequences using HEVC [35] video compression standard. Furthermore, a generic Matlab implementation of a Video Coder (VC) is used to perform preliminary tests on specific scenarios. Two practical applications are proposed and evaluated against two of the best performing SR learning based methods [23], [31] and bicubic up-sampling: combining two LR HEVC compressed streams of the same video into a HR representation and improving the quality of a HR HEVC compressed sequence from a LR one. The proposed framework can also be used as a refinement method on the output of other SR techniques.

The rest of this paper is organized as follows. Section II states the problem and presents the mathematical model. Section III describes in detail the convex optimization method based on the mathematical foundation of [34]. In Section IV, we explain how the proposed framework adapts to HEVC compressed video content. Experimental results and an extended discussion on the proposed method's performance is available in Section V. Finally, Section VI concludes the paper.

II. Modeling the SR problem in the compressed domain

A. Problem statement

We depict a model of the super resolution problem in Fig. 1. Starting from an original video sequence, we consider different degradation models which consist of down-sampling (L) and compressing the source with a video coder.

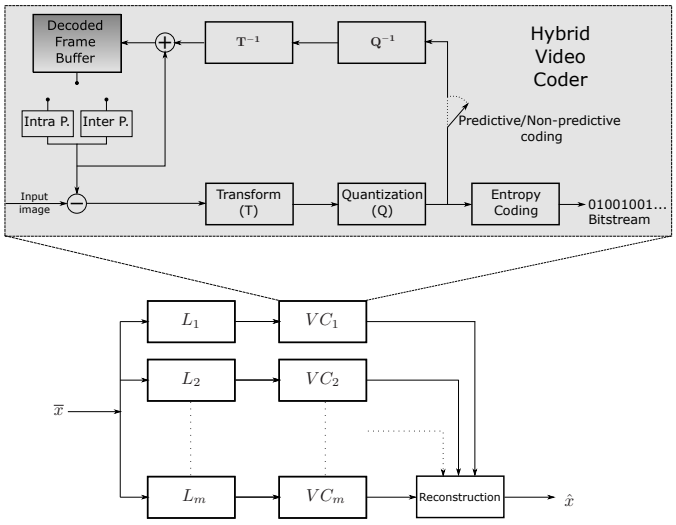

Fig. 1. A generic model for multiple video sources sub-sampling, compression and reconstruction.

1) From pixels to transform coefficients: Since our work features SR from multiple observations, let us consider a set of M encoded data streams - providing views of the same scene - stemming from different video coders. Each video coder has its own configuration (resolution, bitrate, etc.).

We denote by $\overline{x} = [\overline{x_1}, ..., \overline{x_K}]$, with $\forall i \in [1, K], \overline{x_i} \in \mathbb{R}^N$ the original HR sequence. In a compression scheme with no prediction, for every $m \in \{1, \dots, M\}$, the m-th coder generates a vector of coefficients $z_{m,i} \in \mathbb{R}^{P_{m,i}}$ which corresponds to a quantized version of the output of a linear transform $T_{m,i}$ applied to $L_{m,i}\overline{x_i}$, where $L_{m,i}$ is the down-sampling operator (see Sec.II-A3) used with the m-th encoder for frame i. More specifically we have $\forall i \in [1, K]$:

$$\overline{y}_{m,i} = T_{m,i} L_{m,i} \overline{x_i} \tag{1}$$

$$z_{m,i} = \mathcal{Q}_{m,i}(\overline{y}_{m,i}), \tag{2}$$

where $\overline{y}_{m,i}$ and $\mathcal{Q}_{m,i}$ are the transform coefficients of the down-sampled frame and the vector quantizer operator for the m-th encoder and frame i.

The above formulation does not account for the hybrid nature of video coders. Indeed, video coders do not apply directly the transform to pixel blocks, but to a residual obtained by differentiating the observation with a prediction. We therefore denote by $\widetilde{x_{m,i}}$ the predicted image of the m-th encoder for image i. Eq. (1) can thus be rewritten as:

$$\overline{y}_{m,i} = T_{m,i}(L_{m,i}\overline{x_i} - \widetilde{x_{m,i}}). \tag{3}$$

Obtaining the predicted image $\widetilde{x_{m,i}}$ typically depends on the video coder used, and more details about its computation will be given later on in this section.

2) Modeling the quantization process: For the sake of clarity, we remove indexes related to the coder and the image being processed (m and i in the previous section). Let us assume that $\mathcal{Q}$ performs a scalar quantization with n_Q quantization levels $r_1, \dots, r_{n_Q}$ and decisions levels $d_0, \dots, d_{n_Q}$ such that $d_0 < \dots < d_{n_Q}$ as shown in Fig. 2.

Fig. 2. Quantization model: interval limits and reconstruction values.

With these notations, the relation between a quantized coefficient $z^{(k)}$ and the original coefficient $\overline{y}^{(k)}$ follow the subsequent quantization rule:

$$\forall j \in [1, n_Q] \qquad z^{(k)} = r_j \quad \Leftrightarrow \quad \overline{y}^{(k)} \in \mathcal{I}_j, \tag{4}$$

where $\mathcal{I}_j$ is the interval defined as

$$\mathcal{I}_j = \begin{cases} [d_{j-1}, d_j[& \text{if } j < n_Q \\ [d_{n_Q-1}, d_{n_Q}] & \text{if } j = n_Q. \end{cases} \tag{5}$$

We now denote by $(j^{(k)})_{1 \le k \le P}$ the quantization index selected for $z^{(k)}$. Then, it can be deduced that $\overline{y}$ belongs to the following closed convex set

$$C = \Big\{ y = (y^{(k)})_{1 \le k \le P} \in \mathbb{R}^P | \\ (\forall k \in \{1, \dots, P\}) \;\; d_{j^k - 1} \le y^{(k)} \le d_{j^{(k)}} \Big\}, \tag{6}$$

Note that the closure of $\mathcal{I}_{j^{(k)}}$ instead of $\mathcal{I}_{j^{(k)}}$ itself has been considered in order to make C closed (*i.e.* $d_{j^{(k)}} \in C$). The projection onto C can then be straightforwardly defined:

$$\forall y = (y^{(k)})_{1 \le k \le P} \in \mathbb{R}^P, \mathcal{P}_C(y) = (p^{(k)})_{1 \le k \le P}, \tag{7}$$

where $(\forall k \in \{1, \dots, P\})$

$$p^{(k)} = \begin{cases} d_{j^{(k)}-1} & \text{if } y^{(k)} < d_{j^{(k)}-1} \\ d_{j^{(k)}} & \text{if } y^{(k)} > d_{j^{(k)}} \\ y^{(k)} & \text{otherwise.} \end{cases} \tag{8}$$

3) Modeling the re-sampling process: In the present work, $L_{m,i}$ corresponds to a down-sampling process (with or without prefiltering), but it could also account for a registration error, after some suitable linearization. Thus, we model the down-sampling process to account for the most common methods used in video coding: a polyphase filter followed by a decimation. This model accounts for the down-sampling procedure used in the scalable video coding extensions of H.264/AVC (SVC [36]) or HEVC (SHVC [1]) video standards. In the following formulation we use the Fourier transform to express the frequency response of a filter of impulse response $\widetilde{l}[r]$:

$$\widetilde{L}(f) = \sum_{r=0}^{R-1} \widetilde{l}[r] e^{-i2\pi f r}, \tag{9}$$

where R is the kernel size of the filter. Note that other transforms can be used depending on the coding method that is modeled. For example, wavelet transforms can be used to model JPEG 2000 compression [37]. If we consider the polyphase components of the filter as:

$$\widetilde{e_g}[a] = \widetilde{l}[aG + g], \tag{10}$$

where G is the number of phases or components, the filter can now be expressed as a sum of phase components as:

$$\widetilde{L}(f) = \sum_{g=0}^{G-1} \sum_{a=0}^{A-1} \widetilde{e_g}(a) e^{-i2\pi f(aG+g)}, \tag{11}$$

where A is the number of taps for each phase (*i.e.* the kernel size of a single phase filter). Each polyphase component (L_g) can easily be obtained by fixing g in the above equation and summing over a. For convenience, the above formulation assumes that R is a multiple of G; if not, $\widetilde{l}[r]$ can be extended by zero-padding.

In Fig. 3, we depict a simple example of downsampling and upsampling with a factor of $1/2$ and 2 respectively. Here, x represents four adjacent pixels in an image row. U denotes an image expansion with zeros, while D is a decimation. More precisely, the downsampling process uses only 1 phase (0.5), thus, the image is expanded by a factor of 2. The same operation will also be applied on the filter in order to match the zero values in the image. Once the filter is applied, decimation is used to extract the pixels at positions 1.5 and 3.5. Note that the decimation process needs to account for both the phase decimation and the initial expansion of the image. The LR representation is denoted by y, while the downsampling operator L is defined as:

$$L(x) = D_L(U_L(\widetilde{e}_{g_2}) * U_L(x)). \tag{12}$$

The up-sampling process defined by H follows the same logic and is defined as:

$$H(x) = D_H(U_H(\widetilde{e}_{g_1,g_3}) * (U_H(y))). \tag{13}$$

However, in this case two phases are involved, g_1 and g_3. The image is expanded with zeros by a factor of 3 and the filter $\widetilde{e}_{g_1,g_2}$ is defined as:

$$\widetilde{e}_{g_1,g_2} = [w_1^{g_2}, w_1^{g_1}, w_2^{g_2}, w_2^{g_1} ..., w_A^{g_2}, w_A^{g_1}]. \tag{14}$$

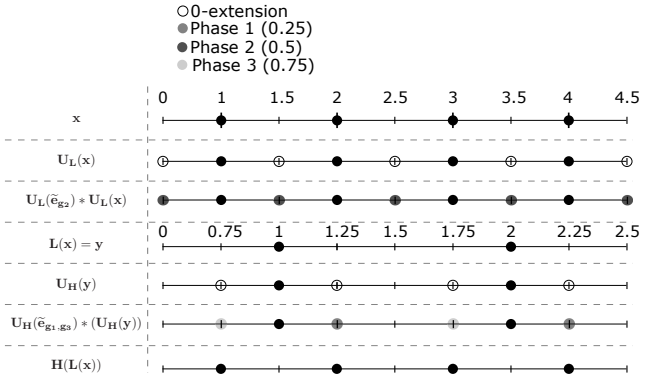

Fig. 3. Down-sampling and up-sampling operators.

In this case the filter is expanded by inserting one zero value in-between consecutive pairs $w_a^{g_2}, w_a^{g_1}$ such that phases 1 and 3 can be computed in a single application of the filter. The decimation process will be used to remove the original pixels of y. We aim to perform a single matrix multiplication in the transform domain in order to easily model the adjoint operator and reduce computational time. Thus, the final filter will be a 2-D version of the current one and the adjoint operator, required for our solver, is easily expressed as the Hermitian transpose (*i.e.* the complex conjugate of the transpose). The weights are determined using any popular interpolation method, such as: Lanczos filter [38], bicubic [39] or filters proposed for SVC [36] or SHVC [40].

B. Modeling the SR process

1) A data-fidelity measure in the compressed domain: We propose to evaluate the fidelity of an observation in the transform domain. In absence of additional clues, reconstruction levels represent the best quality reference (which minimizes the error) for the solution in each transform domain. We opt for the reasonable choice of minimizing the sum of distances between the projections of the sub-sampled solution onto the transform bases and the corresponding quantized transforms observed in the compressed bitstream, according to a suitable metric ϕ_m. Eventually, to account for the different quality of each encoded version, we use an additional parameter α_m :

$$J_{\text{DF}}(\mathbf{x}) = \sum_{i=1}^{K} \sum_{m=1}^{M} \alpha_m \phi_m \left(T_{m,i}(L_{m,i}\widehat{x}_i - \widetilde{x_{m,i}}) - z_{m,i} \right). \tag{15}$$

where $\widehat{x}_i$ is the reconstruction of frame i.

2) Exploitation of available data: The above objective function measures a distance to reconstruction levels in the compressed domain. We propose to strengthen the modeling of the SR problem using all available information in the compressed bitstream. In particular, we know the reconstruction levels and the associated quantization intervals for each quantized coefficient in the bitstream. Since quantization constraints are in the form of a closed convex set $C_{m,i}$ (See Eq. 6), the latter constraints can be directly used in the formulation of the optimization problem. Therefore we enforce the following admissibility condition to the solution:

$$\text{Find } \widehat{x} : \forall m \in [1, M], \forall i \in [1, K], T_{m,i}(L_{m,i}\widehat{x}_i - \widetilde{x}_i) \in C_{m,i}. \tag{16}$$

3) A Priori *knowledge:* Encompassing *a priori* information into the reconstruction problem is a common choice in the literature. We first enforce the solution to have pixel values belonging to a specific *range*, typically known given the application domain. This condition can be expressed as follows:

$$\text{Find } \widehat{x} : \forall i \in [1, K], x_{\text{min}}^{(i)} \leq \widehat{x}^{(i)} \leq x_{\text{max}}^{(i)}. \tag{17}$$

Moreover, a typically adopted choice is to enforce the smoothness of the solution by limiting its discontinuities according to a suitable metric. This is necessary in order to deal with the noise and artifacts introduced at the compression stage. We opt here for the classical Total Variation (TV) [41] to measure the discontinuities of the solution. In order to avoid over-smoothing, the TV will not be introduced in the minimization criterion, but rather limited by means of an additional constraint:

$$\text{find } \widehat{x} : \forall i \in [1, K], \text{TV}(\widehat{x_i}) \leq \eta_i. \tag{18}$$

Obviously, the choice of the bound η_i is critical, and its computation will be detailed in the experiments section.

For the super-resolution case, the sole TV constraint may be insufficient. In particular, we compute data fidelity w.r.t. the reconstruction levels only using the LR (and transformed) versions of the solution. As a matter of fact, among all the possible solutions providing the desired minimum distance, there is still no guarantee that unlikely ones will not be picked. Among them, some may be particularly noisy, in which case

even the activation of the TV constraint can only lead to poor results.

To cope with this problem, we propose to balance the minimization criterion with an additional *super-resolution prior*. To this aim, let us consider a set of up-sampling operators $H_{m,i}$, which can be chosen as to optimally adapt/compensate the corresponding sub-sampling operators $L_{m,i}$. The super-resolution prior is here defined as the distance of the solution $\widehat{x}$ from its subsequently sub-sampled and up-sampled version, according to a suitable metric ψ_m per description, namely:

$$J_{\text{SR}}(\mathbf{x}) = \sum_{i=1}^{K}\sum_{m=1}^{M} \psi_m \left((\text{Id} - H_{m,i}L_{m,i})\widehat{x_i} \right), \tag{19}$$

where Id is the identity matrix. In this way, a preference is expressed in favor of solutions which "look like" the results of proper up-sampling processes, which can be, in our case, adapted to the down-sampling counterparts that generated the observations. Thus, the choice of H for this constraint is critical as it assumes that H is the good solution to reverse L. For example in the case of bicubic down-sampling, H can be easily defined as the bicubic up-sampling process as described in Section II-A3. In fact, this constraint can be interpreted as a correction of the solution w.r.t. the artifacts introduced by a subsequent application of matched up-sampling and down-sampling operators.

The proposed framework can also be used to refine the solution of another super resolution algorithm by changing the initialization. When the initialization is obtained using a combination of the linearly up-sampled observations ($H_{m,i}(obs_{m,i})$), the constraint will help in balancing the solution. However, when a more complex method is used to provide a better initialization, this constraint might not take advantage of the additional information. For instance, example based super resolution methods introduce new information not contained in the observation due to their learning process. In this case, the above constraint will not take advantage of this, as it assumes that H is the "proper" way to reverse L and the new information is regarded as a distortion and corrected. Therefore, when using an initialization based on a complex super resolution algorithm rather than a filter based up-sampling this constraint should be disabled.

4) Wrapping up the SR model: Based on the convex constraints and the objective functions detailed previously, let us now formally define the considered optimization problem. To this aim, we denote by ι_C the characteristic function of a closed convex set C, defined by:

$$\iota_C(y) = \begin{cases} 0 & \text{if } y \in C \\ +\infty & \text{otherwise.} \end{cases} \tag{20}$$

We propose then to minimize the following criterion, with a parameter $\beta \in [0; +\infty[$ allowing to balance the cost functions:

$$\begin{aligned} \text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^{K\times N}}{\text{argmin}} \Bigg(& J_{\text{DF}}(x) + \beta J_{\text{SR}}(x) + \\ & \sum_{i=1}^{K}\sum_{m=1}^{M} \left(\iota_{C_{m,i}}(T_{m,i}(L_{m,i}x_i - \widetilde{x_{m,i}})) \right) + \\ & \sum_{i=1}^{K}\sum_{m=1}^{M} \left(\sum_{s=1}^{S} \iota_{D_s(m,i)}(F_s(x_i)) \right) \Bigg). \end{aligned} \tag{21}$$

where ι_{D_s} is the characteristic function of the convex set D_s for frame i of the m-th description. Note that F_s is used to introduce the range and smoothness constraints from Eq. (17) and Eq. (18) into the problem formulation. The range constraint is directly applied to the image, thus the characteristic function of D_1, $\iota_{D_1(m,i)}$ is applied to $F_1(x_i)$, where F_1 is the identity function and $D_1(m,i) = \{x \in \mathbb{R}^N : x^{(k)} \in [x_{\min}^{m,i}, x_{\max}^{m,i}] \, \forall k \in [1,N]\}$. For the isotropic TV-based smoothness constraint, the image gradient needs to be computed (with $\nabla_{hor}, \nabla_{ver}$ being the horizontal and vertical gradient operators respectively): $F_2 = (\nabla_{hor}, \nabla_{ver})$ and $D_2(m,i) = \{x \in \mathbb{R}^N : \sum_{k=1}^{N} \sqrt{\nabla_{hor}^2 x^{(k)} + \nabla_{ver}^2 x^{(k)}} \leq \eta_i\}$. Essentially, this constraints assure that pixel values remain within a predefined range (*e.g.* [0..255]) and image TV is maintained below a certain threshold η during the variational process.

Furthermore, if frames are compressed without the use of predictive coding, as is the case of intra frames in older coders that do not employ intra-prediction, the data fidelity criterion and the quantization interval based constraint can be easily adapted by replacing $L_{m,i}x_i - \widetilde{x_{m,i}}$ with $L_{m,i}x_i$. Also, note that, $\widetilde{x_{m,i}}$ is given as a constant for each x_i. We could allow the prediction to vary with respect to its reference: $\widetilde{x_{m,i}} = M_{m,i}L_{m,i-1}\widehat{x}_{i-1}$ where $M_{m,i}$ denotes a motion compensation operation. In the case of intra frames that use intra-prediction, we would need to define a new operator that models the intra-prediction process in the specific video coder. However, using such a formulation will introduce non linear operators which complicate the optimization problem. Furthermore, the coefficients of the residual are computed with respect to a certain prediction at the decoder side. Using a different prediction, albeit a better one, might lead to overall worse results when summing with the residual. Therefore, we recommend using a fixed prediction. This is achieved by computing it from the compressed observations before solving the problem. As such, the optimization process can be applied for each frame (x_i) independently and the summation over i can be removed from the model.

III. A CONVEX OPTIMIZATION SOLVER

In this section, we tackle the problem of solving Eq. (21). As discussed in Sec. II-B4, each frame can be optimized independently. As such, for the sake of simplicity we remove the frame index coefficient i in the following description. Considering our problem is based on linear operators, our choice of solver falls on the primal-dual algorithm proposed by Combettes *et*

al. in [34], known as Monotone Lipschitz Forward-Backward-Forward (M-LFBF) algorithm. This algorithm, unlike other similar methods, assures a lower computational complexity for problems involving linear operators as it does not require any matrix inversion [34]. Furthermore, the block iterative structure of the algorithm allows for efficient parallel implementations on multi-core architectures.

In the following section, we will further detail some properties of the proximity operators which are used in this work, followed by a description and discussion of the proposed algorithm.

A. Proximity operators

We begin by defining the proximity operator [42] in a real Hilbert space $\mathcal{H}$ with norm $\|\cdot\|$ for a function $\varphi \in \Gamma_0(\mathcal{H})$. Here, $\Gamma_0(\mathcal{H})$ denotes the class of proper lower semi-continuous convex functions from $\mathcal{H}$ to $]-\infty,+\infty]$. This gives the following definition:

$$\operatorname{prox}_{\varphi}\colon \mathcal{H} \to \mathcal{H}\colon u \mapsto \underset{v\in\mathcal{H}}{\operatorname{argmin}} \frac{1}{2}\|v-u\|^2 + \varphi(v). \quad (22)$$

A useful property which allows us to deal with the reconstructed coefficients in the transform domain (z_m) states the following: If $\psi = \varphi(\cdot - v)$, where $v \in \mathcal{H}$, then

$$(\forall u \in \mathcal{H}) \qquad \operatorname{prox}_{\psi} u = v + \operatorname{prox}_{\varphi}(u - v). \quad (23)$$

Based on this, we can compute the proximity for the data fidelity term $J_{\mathrm{DF}}(\mathbf{x})$. Let us consider $\Phi_m \triangleq \phi_m(\cdot - z_m)$. As such, the data fidelity term can be expressed as $\Phi((T_m(L_m\widehat{x} - \widetilde{x_m})))$ and by applying Eq. (23), we obtain the following expression for the proximity operator:

$$\operatorname{prox}_{\Phi} u = z_m + \operatorname{prox}_{\phi}(u - z_m) \quad with \quad u \mapsto T_m(L_m\widehat{x} - \widetilde{x_m}) \quad (24)$$

Another property of interest is the relation between the projection ($\mathcal{P}$) and proximity operators for characteristics functions of closed convex sets. If $\psi = \iota$ and C is a closed convex set on $\mathcal{H}$, then:

$$(\forall u \in \mathcal{H}) \qquad \operatorname{prox}_{\psi} u = \operatorname{prox}_{\iota_C} u = \mathcal{P}_C(u). \quad (25)$$

B. Algorithm

Using the properties above, the algorithm in [34] can be adapted for solving the problem of Eq. (21). As discussed in Sec. II-A, we need to account for frames which use predictive coding (intra or inter prediction) and also frames for which only transform coding is employed. As the prediction is a constant during the iterative process, we only need to compute it once. Furthermore, if the initialization differs from $H_m(obs)$ (for example a state-of-the-art SR method is used) the super-resolution prior given by Eq. (19) will be disabled.

The algorithm relies on successive computation of the criterion, constraints and their adjoints, and the projection or proximity operators associated to each. For an in-depth understanding, the reader is encouraged to refer to [34].

C. Discussion

As the transform and resampling operations are linear, their adjoint operators are easily computed as discussed in II-A3.

The explicit expression for computing the projection onto C_m set is given in Eq. (7) and (8). In a similar fashion, the projection on the range constraint set D_1 is achieved by setting all out-of-range pixels to the closest bound of interval $[x_{min}, x_{max}]$. For the smoothness constraint, the projection is not available in closed form, but several approaches exist in the literature to compute it [43], [44]. The iterative technique described in [44] and [45] is employed in our algorithm. The computation of proximity operators is based on the explicit expressions available for a large number of convex functions [46], [47].

Furthermore, in order to assure the convergence of the algorithm to an optimal solution according to [34], the iteration step size of the optimization algorithm denoted by γ in [34], *Theorem* 4.2, belongs to the interval $[\epsilon, (1-\epsilon)/\xi]$ and in our case we have:

$$\epsilon \in]0, 1/(\xi+1)[\quad \text{and}$$

$$\xi = \sqrt{\sum_{m=1}^{M}\left(2\|T_m(L_m - \widetilde{M_m})\|^2 + \|\operatorname{Id} - H_m L_m\|^2\right) + \overline{\sum_{s=1}^{S}\|F_s\|^2}}. \quad (26)$$

Note that if predictive coding is not used, $\|T_m(L_m - \widetilde{M_m})\|^2$ becomes $\|T_m(L_m)\|^2$. The norm of the operators can be computed using the iterative algorithm in [48, Algorithm 4].

IV. HEVC Integration

We propose in this paper to apply the proposed SR model to videos compressed with the High Efficiency Video Coding (HEVC) [35]. It is to be highlighted that HEVC always computes a prediction for a coding unit (CU) (more specifically, for each prediction unit PU in a CU), either by Intra or Inter prediction, before encoding the CU residual (Eq. 3). In particular, the predicted frame $\widetilde{x_{m,i}}$ can be built by concatenation of all the predicted units, without explicitly knowing the prediction mode used for each unit.

HEVC computes residual signals at the CU level, but these residuals are transformed at the Transform Unit (TU) level. TUs are square pixel units that can be recursively subdivided, so different transform sizes are specified in HEVC (4x4, 8x8, 16x16, 32x32). Due to complexity considerations, HEVC relies on finite approximations of well-known transforms: the Discrete Cosine Transform (DCT) and its inverse (IDCT). Moreover, a Discrete Sine Transform (DST) is specifically used for 4x4 Intra units. The transform matrices are fully standardized and can be found in [49].

Given an input residual frame (obtained by subtracting the predicted frame from the source signal), the HEVC transform $T_{m,i}$ requires the extraction of the following two elements from HEVC bitstreams : the frame type (to distinguish between DST and DCT for 4x4 units) and the TU partitioning.

HEVC quantization is performed at a TU level on the transformed residual. HEVC implements a scalar quantizer similar to the one presented in Section II-A2. The applicable quantizer is indicated by a Quantization Parameter (QP) ranging from 0 to 51 which serves as an integer index to derive the applicable step size Δ_q. HEVC follows a logarithmic structure : the step size doubles when the QP increases by 6. The first six step sizes (for QP ranging from 0 to 5) are presented below, alongside with the formula allowing to infer the step-size at higher QPs

$$\Delta_{q,0..5} = \{2^{-4/6}, 2^{-3/6}, 2^{-2/6}, 2^{-1/6}, 1, 2^{1/6}\} \quad (27)$$

$$\Delta_q(QP) = \Delta_{q,QP\,mod\,6} \cdot 2^{\lfloor QP/6 \rfloor}. \quad (28)$$

Given an input frame of transformed coefficients, in addition to the information extracted in previous section (TU Partitioning, frame type), the HEVC quantizer $Q_{m,i}$ only requires the extraction of the QP map (containing the QP of each TU) to compute the step-size for each unit.

A. Implementation issues

1) Extracting the required HEVC information: Applying the SR model to HEVC encoded video streams relies on information we can extract during the decoding process (TU partitioning, QP map, etc.). An OpenHEVC decoder [50] has been patched to output the required elements for the SR approach. In particular, the modified decoder generates the following informative streams: the reconstructed frames, the encoded coefficients (denoted as $z_{m,i}$ in Section II) the predicted frames, the TU partitioning and TU types and the QP map.

2) Encoding configurations: HEVC compliant video streams are generated using the reference software HM 15.0 [51]. The encoding uses the default Random Access configuration, with some slight modifications. First, CU-based multi-QP optimization is enabled by setting the parameter $MaxDeltaQP$ at 2. Second, since our SR model has not considered HEVC in-loop filters yet, both the deblocking and sample adaptive offset filters are turned off.

3) A closer look on the HEVC residual skip coding tools: In a generic compression framework as the one denoted by VC in this work (Sec. V-A1), residual information is systematically transformed and quantized. However, HEVC may entirely skip the residual for a block, i.e. when the prediction is good enough given the target quality. This choice is made during Rate-Distortion Optimization (RDO) at the encoder side, and is indicated explicitly in the bitstream. Indeed, HEVC standard defines in the transform tree syntax, for each TU, a flag *cbf luma* which indicates if residual luminance information is present for the current TU (similar codewords *cbf cr* and *cbf cb* are used for chrominance residuals). Besides, the absence of residual is automatically inferred for 64x64 TUs in Inter frames. These 2 scenarios where TUs have no residual are not naturally modeled by our framework. First, the data-fidelity cost function (Eq. (15)) relies on quantized coefficients observed in the bitstream, which are missing in this case. Considering the absence of reliable information, skipped TUs are not taken into account in the data-fidelity computation. Besides, the solution validity (Eq. (16)) relies on quantization intervals which cannot be extracted from the bitstream when the QP of a TU is not defined. This constraint may help to model the uncertainty of skipped TU residuals. Therefore, we relied on empirical testing (see Section V-A2) to define the best selected QP for a skipped TU, by evaluating our SR model on one Inter frame of two HEVC encoded low-resolution observations generated with two different degradation operators: bicubic with anti-aliasing (BicAA) and without (BicNAA).

V. Experimental results

In this section, we report the main results of this work. We begin by discussing the experimental setup and defining the test bench architecture and algorithm parameters which are used throughout the main experiments. A series of small tests are performed in order to show the proposed method's behavior and motivate the framework set-up for the two proposed applications. A discussion of the results concludes this section.

A. Experimental setup and preliminary tests

1) Quick presentation of the experimental setup: In Fig. 4, we depict our test bench architecture. Two observations are generated by applying two different degradation operators denoted by L_1 and L_2 followed by a compression step with a video coder.

In this work, we mainly use the HEVC video coder, with the configuration described in Sec. IV-A2. Additionally, some preliminary tests are also performed using a Matlab implementation of a generic hybrid video coder that matches the scheme in Fig. 1.

The choice of initialization may influence the end result as the algorithm may converge towards a different local minimum. Furthermore, as discussed in Sec. III, if the up-sampled observation introduces new information, the SR prior $I_d - H_m L_m$ will be disabled. Each up-sampling method generates $m + 1$ (m=2 in our tests) natural initialization candidates: $\uparrow_{\mathcal{U}}(Obs_1)$, $\uparrow_{\mathcal{U}}(Obs_2)$, $W.Avg._{\alpha_m}(\uparrow_{\mathcal{U}})$,

$$\uparrow_{\mathcal{U}}(Obs_1), ..., \uparrow_{\mathcal{U}}(Obs_M),$$

$$W.Avg._{\alpha_m}(\uparrow_{\mathcal{U}}) = \sum_{m=1}^{M} \alpha_m \uparrow_{\mathcal{U}}(Obs_m) \quad (29)$$

where $\mathcal{U}$ denotes the up-sampling method (H_m, SOA). Of course, any number of observations can be used if required by a certain scenario and other SOA methods can be combined. However, different test architectures are left as a future study direction. A comparison of different initializations is discussed in Sec. V-A5.

2) Skipped blocks QP selection: As discussed in SectionIV-A3 we performed empirical tests to select a QP for skipped blocks. We tested three different QPs to apply to skipped units: the QP range boundaries: 1 and 51, and the maximum available QP during the frame encoding, which depends on the $MaxDeltaQP$ parameter in HEVC. The results are gathered in Tab. I. When using QP 1, only very small modifications of the quantized coefficients are tolerated

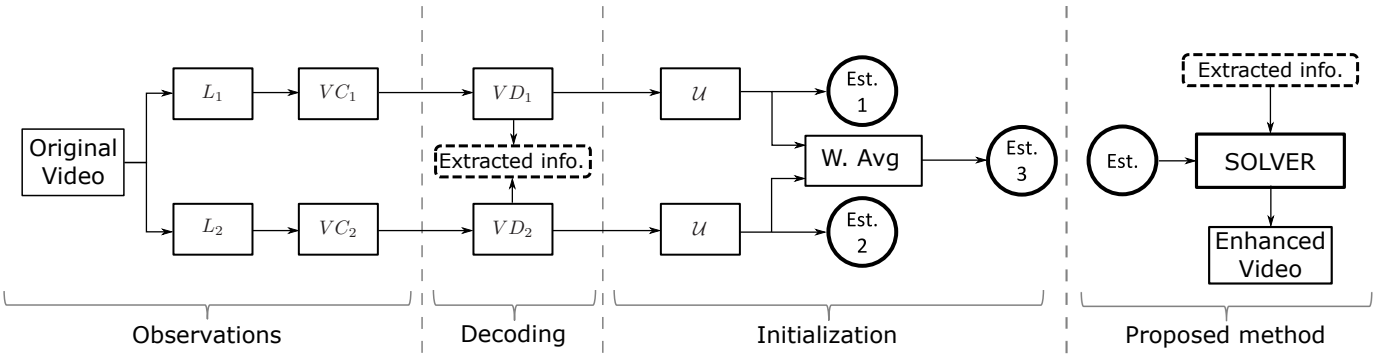

Fig. 4. A schematic view of the experimental setup. Two re-sampling operators are applied on an input sequence. Each observation is compressed and decompressed, and useful information is extracted. Decoded observations are up-sampled to their original resolution using either the reverse of the degradation operator or a State Of the Art (SOA) SR method. Then, the proposed framework is initialized with one high-resolution estimate and the information extracted during the decoding.

		QP of skipped units					
		1		Max. *		51	
Sequence	Enc. QP	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Akiyo	20	**36.61**	0.9598	35.60	**0.9612**	35.29	0.9600
	30	32.45	0.9176	**34.17**	**0.9416**	34.16	0.9415
	40	27.95	0.8449	**30.35**	**0.8802**	**30.35**	**0.8802**
Foreman	20	33.69	0.9179	**34.24**	**0.9287**	32.90	0.9281
	30	29.75	0.8530	**31.25**	0.8873	31.05	**0.8875**
	40	26.11	0.7721	**27.96**	**0.8190**	**27.96**	**0.8190**
Bus	20	**28.24**	**0.8608**	**28.24**	0.8510	27.82	0.8221
	30	25.25	0.7403	26.12	0.7637	**26.24**	**0.7703**
	40	22.02	0.5498	23.36	0.5818	**23.40**	**0.5857**

TABLE I
USING DIFFERENT QP SETTINGS FOR SKIPPED UNITS. (*: MAX. MODE USES THE FRAME MAXIMUM AVAILABLE QP)

(smallest possible quantization interval). Interestingly, this choice is not always the worst at low QPs for static sequences (i.e. Akiyo), but huge quality drops are observed at higher QPs. On the opposite, using QP 51 is almost equivalent to removing the constraint for skipped units: it implies a low confidence on pixel values, and the very wide quantization intervals enable unit variations while preserving their validity. This option has been found slightly inferior (in terms of average quality and stability over sequences/QPs) than the solution based on the maximum encoding QP; we thus applied this latter solution in the remaining of this paper. Such a result is eventually quite intuitive: if the encoder RDO decides to skip a TU, it is most probably because no information is to be coded at the encoding QP, which is thus a natural candidate for modeling the degree of confidence we can have in the unit pixel values.

3) Parameter selection: The application of the proposed framework relies on the definition of some parameters and metrics. First, as presented in Sec.s II-B1 and II-B3, the data-fidelity cost function J_{DF} and SR prior J_{SR} depend on the suitable metrics ϕ_m and ψ_m. As we use the Peak Signal to Noise Ratio (PSNR) to evaluate the results quality, consequently, we rely on the l^2 norm for ϕ_m and ψ_m.

The parameter α_m accounts for the unequal quality of the observations. This parameter may not be easily estimated, since the quality of the observations is not measurable w.r.t. the unavailable original sequence at the decoder side. In the remaining of this work (unless when explicitly indicated), we simply set all α_m to $1/M$ which implies equal importance of each observation.

The constraint imposed on the TV norm of the result (Sec. II-B3, Eq. (18)) has to be defined. In order to obtain an adequate smoothness level, we impose a content dependent boundary on the TV. Namely, we measure for a video frame i the TV of the high-resolution initialization denoted by x_i^0. The TV boundary used for the final result is derived according to:

$$\text{find } \widehat{x}: \quad \text{TV}(\hat{x_i}) \leq \eta \quad \text{where} \quad \eta = \eta_0 \cdot \text{TV}(x_i^0), \qquad (30)$$

where η_0 is used to weight the smoothness of the result. As such, a value close to 1 will lead to a similar smoothness level as the observation whereas smaller η_0 values increase the result smoothness. The η_0 parameter (unless when explicitly indicated) is empirically determined and set to 0.95 in the remaining of this work.

Finally, the β parameter is used to weight the super resolution prior J_{SR} (Eq. 19). Different values could be assigned for individual observations which should reflect the performance of subsequent application of down-sampling and up-sampling ($H_m L_m$) w.r.t. the level of compression. However, in order to preserve the generality of the method we set this parameter to a value of 0.15 in all tests, which provided overall good results on all tested scenarios.

4) Choice of L_m: In order to select the down-sampling operators used in our main experiments, we perform a preliminary test with different choices for the L and H operators (see Sec. II-A3). Our goal is to study the impact of the down-sampling operator L on the performance of our algorithm, the H based up-sampling, A+ [23] and Vnet [32]. We thus limit the test to a certain set of conditions that emphasize the impact of L. Mainly, by using a near lossless compression and relaxing the TV constraint. We select the Foreman sequence in a full intra mode coding configuration as it provides a mid-level TV w.r.t. the test dataset. The choice of video coder for this test falls on the generic VC compression model (see Sec. V-A1 and Fig. 1). To minimize the impact of compression on the performance of the filters we use a QP of 1. In this scenario, we use only 1 description generated with various L models. We test 2 popular interpolation functions: Bicubic (Bic) and Lanczos3 ($Lanc_3$). The Bicubic and Lanczos3

$\downarrow\uparrow$ method	$\uparrow_H$	$\uparrow_{A+}$	$\uparrow_{Vnet}$	$\uparrow_{Prop.}(\uparrow_H)$	$\uparrow_{Prop.}(\uparrow_{A+})$
$\downarrow_L (Bic4)$	34.80	35.36	34.75	34.88	36.95
$\downarrow_L (Bic8)$	34.22	37.92	37.73	37.18	37.96
$\downarrow_L (Bic12)$	32.32	33.84	33.72	36.86	36.88
$\downarrow_L (Lanc_3 6)$	34.51	33.42	32.86	34.53	35.47
$\downarrow_L (Lanc_3 10)$	35.13	37.27	37.06	36.01	38.01
$\downarrow_L (Lanc_3 14)$	32.91	33.84	33.76	36.23	35.61

TABLE II
COMPARING THE PSNR (DB) OF UP-SAMPLING METHODS $\uparrow_H$, $\uparrow_{A+}$, $\uparrow_{Vnet}$, $\uparrow_{Prop}(\uparrow_H)$, $\uparrow_{Prop}(\uparrow_{A+})$ W.R.T DIFFERENT DOWN-SAMPLING FILTERS. (GENERIC VC)

$\uparrow Met.$	$\uparrow_f (Obs_1)$	$\uparrow_f (Obs_2)$	$Avg(\uparrow_f)$
$\uparrow_H$	31.11	31.55	31.56
$\uparrow_{Prop.}(\uparrow_H)$	33.27	33.21	33.36
$\uparrow_{A+}$	32.01	31.72	32.51
$\uparrow_{Prop.}(\uparrow_{A+})$	33.64	33.58	33.97

TABLE III
PSNR (DB) COMPARISONS WHEN USING DIFFERENT INITIALIZATIONS FOR THE PROPOSED METHOD (HEVC). $\uparrow_f$ DENOTES THE UP-SAMPLING WITH METHOD f. THE COLUMNS CORRESPOND TO THE UP-SAMPLED OBSERVATIONS AND THEIR AVERAGE.

functions are defined on the intervals $[-2, 2]$ and $[-3, 3]$. Thus, the phase used in a down-sampling of scale $1/2$, with each filter, has 4 and 6 taps, respectively. Furthermore, we also combine the filter with an anti-aliasing effect by stretching the functions, resulting in a larger number of taps for each phase. The results are reported in Tab. II, where $\uparrow_H$, $\uparrow_{A+}$, $\uparrow_{Vnet}$, $\uparrow_{Prop}(\uparrow_H)$, $\uparrow_{Prop}(\uparrow_{A+})$ denote the up-sampling using the matching filter from L, the dictionary based SOA method [23], the CNN based SOA method [31] and our result when the method is initiated with either $\uparrow_H$ or $\uparrow_{A+}$, respectively. Bic interpolation filters use 4, 8 and 12 taps while $Lanc_3$ has 6, 10 and 14 taps. From the start we can notice that the A+ method provides a significant improvement over filter based up-sampling using the same interpolation function as L. An interesting observation is that the A+ and Vnet methods exhibit a non-uniform performance behavior w.r.t. the number of taps. Best A+ and Vnet performance is achieved for $\downarrow Bic8$ and $\downarrow Lanc_3 10$. Overall, the proposed method obtain comparable quality with SOA methods for $Bic4\&8$, lower quality for $Lanc_3 10$, and outperforms them on the other filters. Using $A+$ as initialization further improves the result. Thus, for fairness of comparison, in our main tests we decide to select the best performing case for A+ and Vnet, $Bic8$ and $Bic4$ as L_1 and L_2 operators. We will refer to these choices as BicAA and BicNAA.

5) Initialization of the proposed method: As discussed in our previous work [33], the initialization can have significant impact on the final result quality. We conduct a preliminary test on Foreman sequence to discuss this phenomenon (CIF, SRx2, 10 HEVC Intra frames, QP 25). Two observation are generated using BicAA and BicNAA (see Sec. V-A4) degradation operators. After compression and decompression, 6 HR estimates are generated : three from the reverse polyphase filters -and their average- and three other generated using the A+ SR method. In Tab. III, we detail the quality of the high-resolution estimates we can derive from the observations, and gather the results obtained by our framework using each estimate as the initialization.

Tab. III highlights typical behavior of the proposed framework: amongst all available high-resolution estimates, the best option is to select the one with highest quality. This namely justifies the use of single-image SR to generate the high-resolution estimate. In this particular example where both observations are encoded with similar compression settings, decoded frames are of comparable quality and to use their average exhibits the best results (with either polyphase up-sampling or A+ SR). In general, observations may be of very different quality, in which case a weighted average may be a better option. Yet, in real-world scenarios at a decoder side, one cannot compute the quality w.r.t. the unavailable original sequence. Thus, choosing the optimal initialization is in general a more difficult problem.

B. Main experiments.

1) SR from two low-resolution observations: In a first scenario, we consider the issue of SR from two low-resolution observations. Here, we assume two compressed video streams and attempt to combine them in order to obtain the best possible quality. The experimental setup follows Fig. 4, and the two observations are generated by down-sampling (by a factor of 2 in each dimension) a given input sequence using BicAA and BicNAA. HEVC is used to compress each LR observation. Two coding configurations are specifically analyzed, denoted by II and IP. II mode corresponds to a full Intra configuration: each frame of the sequence is treated as an independent Intra frame, without motion estimation and compensation tools. On the opposite, IP configuration exploits P frames to improve the coding efficiency, and in this case, a GOP size of 8 is picked. Evaluations are carried out on 6 CIF sequences, mainly: Akyio, Foreman, Bus, Mobile, Football and Flower, denoted by Ak, Fm, Bu, Mo, Fb, and Fl respectively. In this scenario, low-resolution observations are of comparable quality, thus using the average between the observations is a coherent choice. In Tab. IV, we detail results obtained for each sequence at different QP values expressed in terms of PSNR and SSIM [52]. QP1 corresponds to a near lossless compression which is quite relevant as most SR methods are tested on uncompressed frames. Before computing the results, we cropped the borders as Vnet method cannot properly reconstruct these areas. For each QP, three values are presented: the one denoted by *Ref* represents the average between up-sampled decoded observations using polyphase filters. Similarly, the column denoted by A+ measures the quality obtained when averaging the up-sampled decoded observations using the single image SR work from Timofte *et al.* [23]. The column denoted by Vnet represents the quality of the BicAA description up-sampled with Vnet. Note, that in this case we do not use an average as Vnet provides poor results for BicNAA description and the average has a lower quality. Finally, *Prop.* column stands for the proposed algorithm initialized with the A+ result (i.e. the average between decoded observations up-sampled with A+ SR).

Tab. IV highlights the efficiency of the proposed framework in the tested scenario. First, we show a significant improvement over the reference obtained by averaging the bi-cubic up-samplings. Namely, the PSNR gain can be superior to 5dBs for

		QP1				QP15				QP25				QP35				Average			
Seq	Mode	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.
Ak	II	34.8	37.6	38.4	**41.0**	34.6	37.4	38.0	**39.3**	34.1	36.4	36.2	**36.9**	31.8	**32.7**	31.7	32.3	33.8	36.1	36.1	**37.4**
	IP	34.8	37.6	38.4	**40.6**	34.6	37.3	37.7	**37.8**	34.2	36.3	35.9	**36.5**	32.2	33.0	31.8	**33.0**	33.9	36.0	36.0	**37.0**
Fm	II	32.2	37.4	37.8	**39.2**	32.1	37.2	37.2	**38.0**	31.6	35.7	34.9	**35.9**	29.5	**31.6**	30.6	31.3	31.3	35.4	35.1	**36.1**
	IP	32.1	37.4	37.7	**39.3**	32.0	37.0	36.6	**37.6**	31.3	34.7	33.6	**34.8**	28.8	30.3	29.4	**30.3**	31.0	34.8	34.3	**35.5**
Bu	II	26.8	27.5	29.1	**31.4**	26.8	27.5	29.0	**30.1**	26.6	27.3	**28.4**	28.4	25.2	25.7	25.3	**25.7**	26.4	27.0	28.0	**28.9**
	IP	26.8	27.5	29.1	**30.7**	26.8	27.5	28.9	**29.5**	26.4	27.1	27.4	**27.6**	24.1	24.3	23.6	**24.3**	26.0	26.6	27.3	**28.0**
Mo	II	22.7	23.7	25.5	**27.6**	22.7	23.7	25.5	**26.6**	22.6	23.6	**25.1**	25.1	22.0	22.9	22.8	**23.1**	22.5	23.5	24.7	**25.6**
	IP	22.7	23.7	25.5	**26.9**	22.7	23.7	25.4	**26.0**	22.5	23.6	24.4	**24.4**	21.1	21.7	21.3	**21.8**	22.2	23.2	24.1	**24.8**
Fb	II	28.0	30.8	31.4	**33.9**	28.0	30.8	31.2	**32.6**	27.7	30.3	30.2	**31.1**	26.0	**27.4**	26.4	27.3	27.4	29.8	29.8	**31.2**
	IP	28.0	30.8	31.4	**33.3**	28.0	30.7	31.0	**32.0**	27.5	29.7	28.9	**30.0**	24.5	**24.7**	23.9	24.7	27.0	29.0	28.8	**30.0**
Fl	II	23.0	23.4	24.3	**26.5**	23.0	23.4	24.3	**26.0**	22.9	23.3	24.1	**24.8**	22.5	22.9	22.9	**23.3**	22.8	23.2	23.9	**25.1**
	IP	23.0	23.4	24.3	**26.3**	23.0	23.4	24.3	**25.7**	22.8	23.3	23.7	**24.0**	21.9	22.2	22.1	**22.3**	22.7	23.1	23.6	**24.6**
Avg		27.9	30.1	31.1	**33.1**	27.8	29.9	30.8	**31.7**	27.5	29.3	29.4	**29.9**	25.8	26.6	26.0	**26.6**	27.3	29.0	29.3	**30.3**
Ak	II	.964	.977	.977	**.983**	.959	.972	.968	**.975**	.947	**.958**	.949	.955	.907	**.910**	.899	.895	.944	.954	.948	**.952**
	IP	.964	.977	.977	**.979**	.959	.971	.970	**.973**	.948	.958	.951	**.960**	.914	.916	.895	**.918**	.946	.956	.948	**.957**
Fm	II	.940	.958	.960	**.969**	.936	.953	.940	**.955**	.909	**.923**	.894	.922	.851	**.860**	.819	.852	.909	.923	.903	**.924**
	IP	.940	.957	.961	**.968**	.932	.948	**.951**	.950	.900	.909	**.911**	.911	.833	.838	**.843**	.838	.901	.913	.916	**.917**
Bu	II	.852	.893	.914	**.939**	.850	.890	.901	**.915**	.827	.869	.835	**.875**	.700	**.743**	.658	.731	.807	.849	.827	**.865**
	IP	.852	.893	.915	**.931**	.847	.888	**.910**	.907	.809	.851	**.872**	.853	.665	.697	**.708**	.695	.793	.832	.851	**.846**
Mo	II	.787	.858	.883	**.924**	.786	.856	.875	**.897**	.776	.844	.822	**.859**	.714	**.774**	.663	.761	.766	.833	.811	**.860**
	IP	.787	.858	.883	**.908**	.784	.854	.880	**.883**	.765	.831	**.861**	.838	.660	.697	**.734**	.704	.749	.810	.840	**.833**
Fb	II	.877	.922	.918	**.948**	.875	.920	.907	**.930**	.849	**.896**	.829	.893	.703	**.737**	.604	.721	.826	.869	.815	**.873**
	IP	.877	.922	.919	**.944**	.872	.917	.914	**.922**	.836	**.870**	.866	.868	.651	.648	**.693**	.643	.809	.840	.848	**.844**
Fl	II	.829	.868	.882	**.925**	.828	.866	.876	**.908**	.819	.857	.846	**.879**	.778	**.815**	.775	.812	.814	.852	.845	**.881**
	IP	.829	.868	.882	**.917**	.826	.865	.879	**.900**	.813	.851	**.865**	.859	.756	.789	**.797**	.790	.806	.843	.856	**.867**
Avg		.875	.913	.923	**.945**	.871	.908	.914	**.926**	.850	.885	.875	**.889**	.761	**.785**	.757	.780	.839	.873	.867	**.885**

TABLE IV
PSNR (DB) COMPARISON OF THE REFERENCE METHOD, A+ [23], VSRNET (VNET) AND PROPOSED FRAMEWORK, WHEN TWO LOW RESOLUTION OBSERVATIONS ENCODED WITH HEVC ARE AVAILABLE.

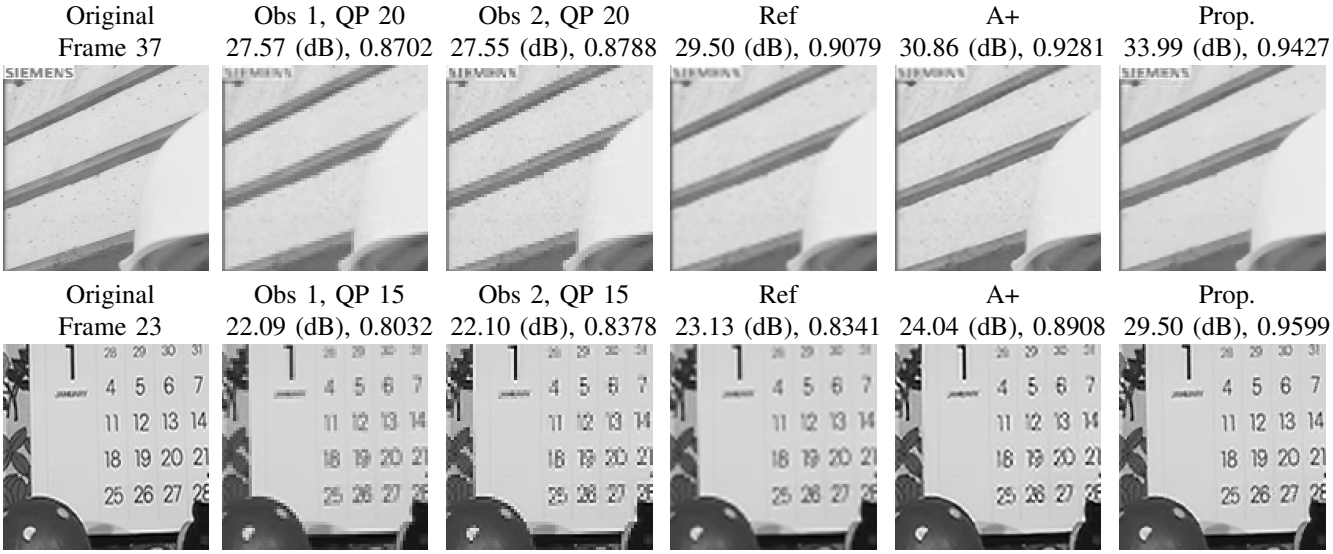

Fig. 5. Details of the up-sampled images and corresponding results of the super-resolution tested methods. PSNR and SSIM values are computed on the compared image patch.

low QP encoding, and up to 0.8dB gain is achieved at QP35 in average. PSNR gains superior to 2 and 3dBs can be observed at low QPs, when compared to Vnet and A+, respectively. However, note that at high QPs the methods tend to exhibit similar performance, as the high compression level combined with the down-sampling operation lead to a highly unreliable description for inferring additional information.

Significant gains are obtained over the SOA SR methods, and best results are obtained at low QPs. The impact of the QP on the method efficiency is related to the extensive use of transformed coefficients -observed in the bitstream- and their respective quantization intervals. The reliability of these anchors highly depends on the encoding QP, which justifies the higher performance of the approach for high quality encodings.

Visual results are available in Fig. 5. Image details from Foreman and Mobile sequences are depicted for each observation and tested method. PSNR and SSIM results are also reported for each image. It is easily noticeable that the proposed method provides the best visual results. Considering the multiple sources of artifacts (down-sampling and compression) and that PSNR and SSIM results are not always consistent, the evaluation could be further extend in the future. It could

be interesting to perform a subjective study and also use non-reference quality evaluation techniques such as the one proposed by Zhang *et al.* in [53].

The learning based methods have no prior knowledge of the compression and degradation models. As such, for fairness of comparison we perform an additional experiment. The best performing reference method is Vnet. We consider a best case scenario in which we have access to the degraded and original video sequences and the network can be fine tunned on this particular dataset. It should be noted, that for a general-purpose training, large video databases should be used without overlapping between training and test data. However, in this experiment we aim to maximize the performance of the network on our test sequences.

In order to account for the different compression models and filters we fine tune the network over 3 QP intervals, mainly $1-15-20$, $20-25-30$ and $30-35-40$. Following a similar procedure to [31], we extract 30000 $36 \times 36 \times 5$ data cubes (co-located patches in 5 consecutive frames after motion compensating the first 2 and last 2 frames) obtained at random positions in frames of 5 of the tested video sequences, compressed and down-sampled with the two degradation models. Note, that Flower sequence was not used for training. The fine tuning was performed starting from the original filter weights of the network and using a reduced learning rate. The rest of the parameters are the same as in [31]. In this scenario the network will gain knowledge of both the down-sampling models and the video compression, while using multiple networks to assure the best results for each compression interval. Furthermore, using the same training and test data, provides a higher performance for the network. We refer to this approach as V+, obviously the results for each QP are obtained using the appropriate weights.

The results are shown in Table V. The first two columns for each QP show the PSNR and SSIM results for the up-sampled BicAA observation and the average of the two observation. As the BicNAA observation still has a lower performance we do not show these results. The sequences highlighted in green were also used in the training, while the red highlighted sequence was not used in the training database. We can notice that there is an improvement with respect to Table IV where the results for Vnet BicAA are reported. However, unlike the first experiment, the average of the two observations (V+ avg) now performs significantly better for higher QPs, managing to outperform our model on the sequences that were used in training. Unfortunately, some losses can be observed at lower QPs w.r.t. V+ O1.

Another interesting thing to note in this case are the results obtained on Flower sequence which was not used in the training data. Losses can be observed for low QPs w.r.t. the results in Table IV, while higher QPs are slightly improved. In average the proposed approach still manages to outperform this best case scenario even when using the same training and test data. At the cost of loosing generality given a known dataset, degradation model and compression level a network can learn to reconstruct high frequency components for that specific content, which cannot be reproduced by model based approaches. However, this is not always applicable for data that was not used in the training set, as can be observed by the performance gains of the network on Flower sequence.

2) SR from one low-resolution observation and one high-resolution observation: In a second scenario, we consider the case where one observation is available at LR and the other one is available at the original HR. Our framework is capable of combining these observations naturally, since each observation is modeled with its own degradation model. In general, HR coded streams exhibit higher quality than up-sampled low-resolution streams, encoded with similar parameters. This behavior leads to a large ΔPSNR between HR and LR descriptions. Intuitively, if the ΔPSNR is very large there is not a lot of information that can be extracted from a LR observation which is not already contained in the HR description. Therefore, we begin this scenario with a small test performed on a few frames of Bus sequence, with generic HEVC in full Intra mode. Our goal is to analyze the algorithms behavior w.r.t. the ΔPSNR of the two observations, denoted by Δ_{Obs} in Tab. VI. $\uparrow_H$ Obs 1, $\uparrow_{A+}$ Obs 1 and Prop denote the up-sampled observation with H and A+ methods and the result obtained by our proposed method. Δ is the improvement obtained with Prop over Obs 2. First column shows the QPs used in coding Obs 1 and Obs 2, respectively. In this test the initialization of M-LFBF solver was Obs 2. We can easily notice that higher gains are achieved when the descriptions are more similar in terms of quality. An interesting observation can be made for QPs 1&20 and 15&20. Even though the quality of Obs 1 only increases by 0.02dB and Obs 2 remains unchanged we can see a large difference in Δ (from 0.69dB to 1.47dB). This behavior can be explained by the algorithm dependency on the information variety between descriptions, rather than their individual quality. Tests performed on other sequences reveals a similar behavior, however, for the sake of brevity we do not repeat this test for each encoder, configuration and sequence. As such, we decide to perform a complete set of tests using a QP combination that provides similar quality observations: QPs 1 and 40, 15 and 40 and 15 and 35 for the LR and HR observations.

As we did in our previous experiment (Sec. V-B1), we report both PSNR and SSIM scores. The LR observation is obtained with BicAA down-sampling. As the quality of the observations is closer than in our preliminary tests, we initialize the algorithm using the average. Ref, A+ and Vnet in this case denote the average between Obs 2 and Obs 1 up-sampled with H, A+ and Vnet, respectively. The results are reported in Tab. VII. Our algorithm outperforms the reference and A+ methods on all sequences, and Vnet in most cases. On average over all sequences and all QPs combinations, gains of 2.1dB, 0.9dB and 0.3dB are obtained over Ref, A+ and Vnet, respectively.

C. Convergence speed

In Fig. 6, we show the PSNR of the estimated HR image and the distance to the convex set C at each iteration of the solver. The distance to the convex set C in this case is computed as the absolute error between the transform of the down-sampled estimate and its projection on the convex set

		QP1			QP15			QP25			QP35			Average		
Seq	Mode	V+ O1	V+ avg	Prop.	V+ O1	V+ avg	Prop.	V+ O1	V+ avg	Prop.	V+ O1	V+ avg	Prop.	Vnet+	Vnet+	Prop.
Ak	II	38.6	38.3	**41.0**	38.2	38.0	**39.3**	36.5	**37.1**	36.9	31.9	**33.0**	32.3	36.3	36.6	**37.4**
	IP	38.5	38.3	**40.6**	**38.0**	37.9	37.8	36.3	**36.9**	36.5	32.1	**33.3**	33.0	36.2	36.6	**37.0**
Fm	II	37.6	37.7	**39.2**	37.2	37.5	**38.0**	35.2	**36.1**	35.9	30.8	**31.8**	31.3	35.2	35.8	**36.1**
	IP	37.6	37.7	**39.3**	36.7	37.3	**37.6**	34.0	**35.0**	34.8	29.5	**30.4**	30.3	34.4	35.1	**35.5**
Bu	II	28.8	29.2	**31.4**	28.8	29.1	**30.1**	28.2	**28.8**	28.4	25.4	**26.4**	25.7	27.8	28.4	**28.9**
	IP	28.8	29.2	**30.7**	28.7	29.1	**29.5**	27.3	**28.3**	27.6	23.8	**24.7**	24.3	27.2	27.8	**28.0**
Mo	II	25.6	26.0	**27.6**	25.6	26.0	**26.6**	25.3	**25.8**	25.1	23.0	**24.2**	23.1	24.9	25.5	**25.6**
	IP	25.6	26.0	**26.9**	25.5	25.9	**26.0**	24.5	**25.5**	24.4	21.5	**22.3**	21.8	24.3	**24.9**	24.8
Fb	II	30.9	31.3	**33.9**	30.8	31.3	**32.6**	29.9	30.8	**31.1**	26.5	**27.7**	27.3	29.5	30.3	**31.2**
	IP	30.9	31.3	**33.3**	30.7	31.2	**32.0**	28.9	**30.1**	30.0	24.0	**24.9**	24.7	28.6	29.4	**30.0**
Fl	II	24.0	23.9	**26.5**	24.0	23.9	**26.0**	23.9	23.8	**24.8**	22.8	23.3	**23.3**	23.7	23.7	**25.1**
	IP	24.0	23.9	**26.3**	24.0	23.9	**25.7**	23.5	23.7	**24.0**	22.1	**22.5**	22.3	23.4	23.5	**24.6**
Avg		30.9	31.1	**33.1**	30.7	30.9	**31.7**	29.5	**30.1**	29.9	26.1	**27.0**	26.6	29.3	29.8	**30.3**
Ak	II	.978	.978	**.983**	.971	.973	**.975**	.954	**.961**	.955	.902	**.916**	.895	.951	**.957**	.952
	IP	.977	.977	**.979**	.970	.972	**.973**	.953	**.960**	.960	.906	**.920**	.918	.952	.957	**.957**
Fm	II	.954	.957	**.969**	.947	.953	**.955**	.912	**.924**	.922	.844	**.861**	.852	.914	.924	**.924**
	IP	.954	.957	**.968**	.938	.948	**.950**	.897	**.911**	.911	.820	**.839**	.838	.902	.914	**.917**
Bu	II	.901	.913	**.939**	.896	.911	**.915**	.862	**.889**	.875	.705	**.749**	.731	.841	.865	**.865**
	IP	.900	.913	**.931**	.889	**.908**	.907	.829	**.865**	.853	.659	**.704**	.695	.819	**.847**	.846
Mo	II	.886	.901	**.924**	.883	**.899**	.897	.866	**.888**	.859	.743	**.808**	.761	.844	**.874**	.860
	IP	.886	.901	**.908**	.880	**.898**	.883	.832	**.873**	.838	.678	**.730**	.704	.819	**.850**	.833
Fb	II	.909	.919	**.948**	.905	.917	**.930**	.861	**.894**	.893	.694	**.734**	.721	.842	**.866**	.873
	IP	.909	.919	**.944**	.899	.914	**.922**	.830	**.870**	.868	.610	**.652**	.643	.812	.839	**.844**
Fl	II	.864	.870	**.925**	.862	.868	**.908**	.849	.860	**.879**	.785	**.814**	.812	.840	.853	**.881**
	IP	.864	.870	**.917**	.859	.867	**.900**	.833	.853	**.859**	.765	.788	**.790**	.830	.844	**.867**
Average		.915	.923	**.945**	.908	.919	**.926**	.873	**.896**	.889	.759	**.793**	.780	.864	.883	**.885**

TABLE V
PSNR (DB) COMPARISON OF THE REFERENCE METHOD VSRNET FINE TUNNED ON THE TEST SEQUENCES FOR MULTIPLE QP INTERVALS FOR THE BICAA OBSERVATION (V+ O1), THE AVERAGE (V+AVG) AND PROPOSED FRAMEWORK, WHEN TWO LOW RESOLUTION OBSERVATIONS ENCODED WITH HEVC ARE AVAILABLE.

QPs	$\uparrow_H Obs_1$	$\uparrow_{A+} Obs_1$	Obs_2	Δ_{Obs}	Prop.	Δ
15 20	26.41	28.56	43.76	**17.35**	44.45	**0.69**
1 20	26.43	28.63	43.76	**17.32**	45.23	**1.47**
1 25	26.43	28.64	39.33	**12.9**	41.45	**2.12**
1 30	26.43	28.63	35.15	**8.72**	37.63	**2.48**

TABLE VI
PSNR (DB) COMPARISON OF DIFFERENT QP COMBINATIONS FOR A LOW-RESOLUTION AND A HIGH-RESOLUTION DESCRIPTION ON BUS SEQUENCE WITH GENERIC VC USING II CONFIGURATION.

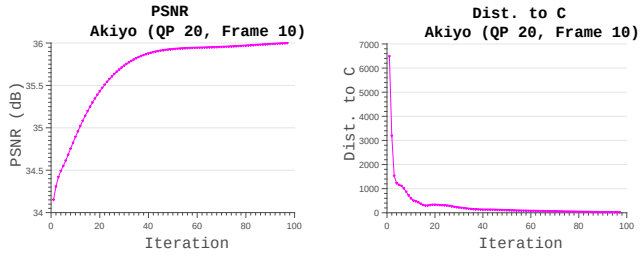

Fig. 6. PSNR and distance to the convex set C for each iteration. The distance represents the absolute error between the estimated image and its projection onto C.

C. The test is performed on Akiyo sequence, frame 10, from two descriptions using BicAA and BicNAA down-samplings encoded with QP 20. For the sake of brevity, we do not show other examples, as the behavior is similar across different sequences and compression levels. The stop criterion used in our experiments is:

$$\text{Stop if:} \quad \text{mean}\left(|x_i - x_{i-1}|\right) \leq 10^{-4} \tag{31}$$

where $||$ denotes the absolute value and mean is the average value of the image pixels. In this case, 97 iterations were performed before the algorithm was stopped, of course, the number of iterations can be increased by modifying the threshold. However, we found that in most tests, 80% of the maximum gain was obtained in the first 30 to 40 iterations. Note that the PSNR is still increasing when the distance to set C is 0, as the cost function uses multiple constraints.

The average time per iteration with a Matlab sequential implementation, for two LR descriptions, on a workstation with core I7-6700 processor is 0.4 seconds. Thus, one frame can be super-resolved in 20 to 40 seconds with 50 to 100 iterations. In our experiments we used a limit of 50 iterations. A+ method has a runtime of approximately 1 second. In the case of Vnet, super resolving 1 frame required about 6-7 seconds with the optical flow computation accounting for most of the runtime and the forward pass of the network taking less than 0.3 seconds. However, it should be noted that both A+ and Vnet require an additional training step. Furthermore, Vnet uses an optimized GPU implementation as the network is modeled in the Caffe framework [54]. Looking at algorithm complexity, the proposed approach uses linear operators which are applied through convolutions and matrix multiplication operations, thus, an optimized implementation can bring a significant reduction in computational time. Depending on the usage scenario, the algorithm can be limited to a relatively small number of iterations (10-20) with a reduced gain and a lower computational time. Furthermore, this type of convex optimization solvers can be easily parallelized for a multi-core implementation[55].

VI. CONCLUSIONS AND FUTURE WORK

This work presents a model-based SR approach specifically designed for compressed video streams, and focuses on scenarios where multiple observations are available. The proposed model makes an explicit use of the available compressed

		QPs 1 & 40				QPs 15 & 40				QPs 15 & 35				Average			
Sequence	Mode	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.	Ref	A+	Vnet	Prop.
Akiyo	II	34.9	36.7	37.0	**37.8**	34.8	36.5	36.8	**37.2**	34.6	38.6	38.5	**39.3**	34.8	37.2	37.4	**38.1**
	IP	35.3	37.1	37.3	**38.3**	35.2	36.8	37.0	**37.4**	34.5	**38.6**	38.4	38.5	35.0	37.5	37.5	**38.1**
Foreman	II	32.6	33.8	35.3	**36.0**	32.6	33.6	35.1	**35.4**	32.1	35.0	36.3	**36.6**	32.4	34.1	35.6	**36.0**
	IP	32.3	33.2	34.6	**35.2**	32.2	33.0	34.2	**34.2**	32.0	34.3	**35.1**	35.0	32.1	33.5	34.6	**34.8**
Bus	II	28.1	29.4	29.8	**30.3**	28.1	29.3	29.7	**30.2**	26.5	31.1	31.6	**32.1**	27.6	29.9	30.3	**30.9**
	IP	27.4	28.6	29.0	**29.4**	27.4	28.5	28.9	**29.1**	26.4	30.1	**30.3**	30.3	27.1	29.1	29.4	**29.6**
Mobile	II	25.2	26.5	27.3	**27.3**	25.2	26.5	27.2	**27.3**	22.6	28.3	29.1	**29.5**	24.4	27.1	27.9	**28.1**
	IP	25.1	26.4	**27.1**	27.0	25.1	26.3	**27.0**	27.0	22.6	27.6	**28.3**	28.3	24.3	26.8	27.5	**27.4**
Football	II	28.9	30.6	31.1	**31.7**	28.9	30.5	31.1	**31.5**	28.7	32.7	32.8	**33.5**	28.8	31.3	31.7	**32.2**
	IP	27.8	29.2	29.3	**29.5**	27.8	29.1	29.2	**29.2**	28.6	**31.0**	30.7	30.8	28.1	29.8	29.7	**29.8**
Flower	II	25.9	26.5	27.0	**27.2**	25.9	26.5	27.0	**27.2**	23.0	28.2	28.7	**29.4**	24.9	27.1	27.6	**27.9**
	IP	25.3	25.9	26.4	**26.5**	25.3	25.9	26.4	**26.5**	23.0	27.1	27.4	**27.7**	24.5	26.3	26.7	**26.9**
Average		29.1	30.3	30.9	**31.4**	29.0	30.2	30.8	**31.0**	27.9	31.9	32.3	**32.6**	28.7	30.8	31.3	**31.7**
Akiyo	II	.952	.965	.964	**.973**	.948	.960	.959	**.964**	.956	.969	.965	**.971**	.952	.965	.963	**.969**
	IP	.954	.966	.966	**.974**	.951	.961	.961	**.964**	.955	**.970**	.966	.965	.954	.966	.964	**.968**
Foreman	II	.913	.931	.934	**.951**	.909	.925	.928	**.934**	.930	.936	.929	**.939**	.917	.931	.930	**.941**
	IP	.907	.925	.928	**.938**	.901	.914	**.917**	.912	.923	**.926**	.918	.920	.910	.922	.921	**.923**
Bus	II	.828	.868	.887	**.903**	.826	.865	.884	**.888**	.844	.912	.916	**.917**	.832	.882	.896	**.903**
	IP	.820	.861	.879	**.891**	.816	.854	**.872**	.870	.838	**.901**	.899	.893	.825	.872	.883	**.885**
Mobile	II	.849	.890	.901	**.907**	.848	.888	.899	**.901**	.772	.928	**.933**	.932	.823	.902	.911	**.913**
	IP	.857	.895	.903	**.905**	.855	.891	**.900**	.897	.770	.921	**.924**	.918	.827	.902	.909	**.907**
Football	II	.830	.874	.883	**.912**	.828	.871	.880	**.895**	.880	.912	.907	**.916**	.846	.886	.890	**.908**
	IP	.808	.855	.857	**.879**	.804	.849	.850	**.854**	.876	**.890**	.877	.879	.829	.864	.862	**.871**
Flower	II	.885	.906	.915	**.917**	.884	.903	**.913**	.910	.823	.934	**.939**	.936	.864	.914	.923	**.921**
	IP	.878	.899	**.910**	.909	.876	.895	**.907**	.901	.821	.922	**.925**	.918	.858	.906	.914	**.909**
Average		.873	.903	.911	**.921**	.870	.898	.906	**.908**	.866	.927	.925	**.925**	.870	.909	.914	**.918**

TABLE VII
PSNR (DB) COMPARISON OF THE REFERENCE METHOD, A+, VNET AND PROPOSED FRAMEWORK, WHEN ONE LOW RESOLUTION AND ONE HIGH RESOLUTION OBSERVATIONS ENCODED WITH HEVC ARE AVAILABLE.

syntax (encoded coefficients, unit sizes, etc.) and builds a heterogeneous cost function combining data-fidelity objectives and a priori constraints. The resulting minimization problem, efficiently solved via convex optimization, embeds the SR result into a domain that closely fits the given compressed observations. Experimental results demonstrate that in most cases combining the complementary information available in the different observations allows very efficient SR, significantly outperforming the capabilities of image [23] or single video stream [31]. Indeed, quality improvements superior to 2dB w.r.t. one of the best performing learning-based single image SR method can be observed for high-quality encodings, which has a noticeable impact on the visual quality of the reconstructed video sequence. The flexibility of the proposed framework is also to be highlighted. First, an arbitrary number of observations can be considered. Second, each observation is modeled with its own degradation model, allowing to combine observations at different resolutions and smoothness characteristics. Such an explicit modeling enables to avoid a typical pitfall of learning-based approaches whose performance may dramatically vary depending on the re-sampling used to generate the observation. Third, the approach can be applied to different video coders, which is illustrated in the present work using both a generic coding model VC and the HEVC standard. Extending the framework application to other compression schemes (AVC, JPEG, JPEG2000, VC9, etc.) is straightforward. Another interesting future research direction is to combine AVC and HEVC video streams. Yet, short-term research focuses on investigating more thoroughly the complexity and real-time capabilities of the proposed framework, requiring the implementation and optimization of the convex solver on parallel processing platforms.

REFERENCES

[1] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 20–34, 2015.

[2] A. Purica, E. G. Mora, M. Cagnazzo, B. Pesquet-Popescu, and B. Ionescu, "Multiview plus depth video coding with temporal prediction view synthesis," *IEEE Transactions on circuits and systems for video technology*, vol. 26, no. 2, pp. 360–374, 2016.

[3] P. Milanfar, *Super-Resolution Imaging, Digital Imaging and Computer Vision*. Taylor&Francis/CRC Press, 2010.

[4] S. K. Nelson and A. Bhatti, "Performance evaluation of multi-frame super-resolution algorithms," in *International Conference on Digital Image Comput-ing: Techniques and Applications (DICTA)*, Centre for Intelligent Systems Research, Deakin University, Geelong,Victoria,Australia, 2012.

[5] L. Tian, A. Suzuki, and H. Koike, "Task-oriented evaluation of super-resolution techniques," in *nt. Conference on Pattern Recognition (ICPR)*, 2010, pp. 493–498.

[6] Y. J. Kim, J. H. Park, G. S. Shin, H.-S. Lee, D.-H. Kim, S. H. Park, and J. Kim, "Evaluating super resolution algorithms," in *SPIE-IS&T Electronic Imaging, Image Quality and System Performance VIII*, vol. 7867, 2011.

[7] Z. Jin, T. Tillo, C. Yao, J. Xiao, and Y. Zhao, "Virtual-view-assisted video super-resolution and enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 467–478, March 2016.

[8] S. Babacan, R. Molina, and A. Katsaggelos, "Variational bayesian super resolution," *IEEE Transactions in Image Process*, pp. 984–999, 2011.

[9] R. Molina, J. Nez, F. Cortijo, and J. Mateos, "Image restoration in astronomy bayesian perspective," *IEEE Signal Process. Mag.*, vol. 18, pp. 11–29, 2001.

[10] S. Villena, M. Vega, D. Babacan, R. Molina, and A. Katsaggelos, "Bayesian combination of sparse and non sparse priors in image super resolution," *Digital Signal Processing*, vol. 23 (2), pp. 530–541, 2013.

[11] S.Villenaa, M.Vegaa, R.Molinab, and A.K.Katsaggelosc, "A non-stationary image prior combination in super-resolution," *Digital Signal Processing*, vol. 32, 2014.

[12] C. Liu and D. Sun, "On bayesian adaptive video super resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 346–360, 2014.

[13] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 16, pp. 185–203, August 1981.

[14] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Processing Magazine*, vol. 13, no. 3, pp. 43–64, May 1996.

[15] C. Liu, R. Szeliski, S. Kang, C. Zitnick, and W. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 299–314, February 2008.

[16] Z. Ma, J. Jia, and E. Wu, "Handling motion blur in multi-frame super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[17] E. Faramarzi, D. Rajan, F. C. A. Fernandes, and M. P. Christensen, "Blind super resolution of real-life video sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1544–1555, April 2016.

[18] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos, *Super-Resolution from Compressed Video*. Boston, MA: Springer US, 2002, ch. 11, pp. 211–242.

[19] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Super-resolution reconstruction of compressed video using transform-domain statistics," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 1, pp. 33–43, January 2004.

[20] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos, "Bayesian resolution enhancement of compressed video," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 898–911, July 2004.

[21] S. P. Belekos, N. P. Galatsanos, and A. K. Katsaggelos, "Maximum a posteriori super-resolution of compressed video with a novel multi-channel image prior and a new observation model," *European Signal Processing Conference*, 2011.

[22] C. Wang, G. Yang, and Y.-P. Tan, "Reconstructing videos from multiple compressed copies," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 19, no. 9, pp. 1342–1352, September 2009.

[23] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast superresolution," in *Computer Vision ACCV 2014*, ser. Lecture Notes in Computer Science, D. Cremers, I. Reid, H. Saito, and M. Yang, Eds., vol. 9006. Springer International Publishing, 2015, pp. 111–126.

[24] Y. Zhu, Y. Zhang, and A. L. Yuille, "Single image super-resolution using deformable patches," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, September 2014.

[25] W. Shi, J. Caballero, F. Huszar, and L. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," *Proc. IEEE Eur. Conf. Comput. Vis.*, pp. 184–199, 2014.

[27] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1382–1394, April 2013.

[28] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, November 2010.

[29] Q. Dai, S. Yoo, A. Kappeler, and A. K. Katsaggelos, "Dictionary-based multiple frame video super-resolution," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 83–87.

[30] Q. Dai, S. Yoo, A. Kappeler, and A. K. Katsaggelos, "Sparse representation-based multiple frame video super-resolution," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 765–781, February 2017.

[31] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging*, vol. 2, pp. 109–122, June 2016.

[32] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Super-resolution of compressed videos using convolutional neural networks," in *IEEE International Conference on Image Processing (ICIP)*, September 2016.

[33] B. Boyadjis, B. Pesquet-Popescu, F. Dufaux, and C. Bergeron, "Super-resolution of hevc videos via convex optimization," in *IEEE International Conference on Image Processing (ICIP)*, September 2016.

[34] P. L. Combettes and J.-C. Pesquet, "Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators," *Set-Valued Anal*, vol. 20, no. 2, pp. 307–330, 2012.

[35] B. Bross, W.-J. Han, J.-R. Ohm, G. Sullivan, and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 8," in *JCTVC-J1003*, Stockholm, Sweden, July 2012.

[36] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Joint draft itu-t rec. h. 264 iso/iec 14496-10/amd. 3 scalable video coding," *Joint Video Team (JVT) JVT-X201*, vol. 108, p. 1990, 2007.

[37] ISO/IEC 15444-1, "Jpeg 2000 image coding system," JPEG, Tech. Rep., 2000.

[38] K. Turkowski and S. Gabriel, "Filters for common resampling tasks," *Andrew S. Glassner Graphics Gems I, Academic Press*, pp. 147–165, 1990.

[39] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 6, December 1981.

[40] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, "Scalable HEVC (SHVC) Test Model 10 (SHM 10)," JCT-VC of ITU-T SG16 WP 3and ISO/IEX JTC 1/SC 29/WG 11, June 2015.

[41] P. L. Combettes and J.-C. Pesquet, "Image restoration subject to a total variation constraint," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1213–1222, 2004.

[42] J.-J. Moreau, "Proximite et dualite dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[43] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[44] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "Epigraphical projection and proximal tools for solving constrained convex optimization problems: Part i," *Tech. Rep. Telecom ParisTech*, 2012.

[45] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "Epigraphical projection and proximal tools for solving constrained convex optimization problems," *Signal, Image and Video Processing*, vol. 9, no. 8, pp. 1737–1749, 2015.

[46] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," *Fixed-point algorithms for inverse problems in science and engineering*, p. 185212, 2010.

[47] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs, "A variational formulation for frame based inverse problems," *Inverse Probl.*, vol. 23, no. 4, pp. 1495–1518, 2007.

[48] L. Chaari, N. Pustelnik, C. Chaux, and J.-C. Pesquet, "Solving inverse problems with overcomplete transforms and convex optimization techniques," in *Proc. SPIE Wavelets, San Diego, CA, USA*, 2009.

[49] M. Budagavi, A. Fuldseth, and G. Bjontegaard, "HEVC Transform and Quantization," in *High Efficiency Video Coding (HEVC)*, ser. Integrated Circuits and Systems, V. Sze, M. Budagavi, and G. Sullivan, Eds. Springer International Publishing, 2014, pp. 141–169.

[50] "OpenHEVC Open source HEVC decoder," https:/github.com/openhevc/openhevc/.

[51] J. video team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "H.265/HEVC HM reference software," http://hevc.fraunhofer.de//, May 2013.

[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.

[53] L. Zhang, L. Zhang, and A. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2579–2591, August 2015.

[54] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[55] R. Gaetano, G. Chierchia, and B. Pesquet-Popescu, "Parallel implementations of a disparity estimation algorithm based on a proximal splitting method," in *IEEE Visual Communications and Image Processing (VCIP)*, 2012, pp. 1–6.