



**HAL**  
open science

## A study of norms in convex optimization super-resolution from compressed sources

Andrei Purica, Benoit Boyadjis, Béatrice Pesquet-Popescu, Frederic Dufaux

► **To cite this version:**

Andrei Purica, Benoit Boyadjis, Béatrice Pesquet-Popescu, Frederic Dufaux. A study of norms in convex optimization super-resolution from compressed sources. 19th International Workshop on Multimedia Signal Processing (MMSP), Oct 2017, London-Luton, United Kingdom. 10.1109/mmsp.2017.8122216 . hal-01567378

**HAL Id: hal-01567378**

**<https://hal.science/hal-01567378>**

Submitted on 10 Jan 2020

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

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

# A study of norms in convex optimization super-resolution from compressed sources

Andrei Purica\*, Benoit Boyadjis\*<sup>‡</sup>, Béatrice Pesquet-Popescu\*, Frédéric Dufaux<sup>†</sup>

\*LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France

<sup>†</sup>L2S, CNRS, CentraleSupélec, Univ. Paris Sud – 3 rue Joliot Curie, Gif-sur-Yvette, France

<sup>‡</sup>Thales Communications and Security, OPS/HTE/STR/MMP – Gennevilliers, France

*first.last@telecom-paristech.fr – first.last@thalesgroup.com – first.last@l2s.centralesupelec.fr*

**Abstract**—Advancements over the last decade in video acquisition and display technologies lead to a continuous increase of video content resolution. These aspects combined with the shift towards cloud multimedia services and the underway adoption of High Efficiency Video Coding standards (HEVC) create a lot of interest for Super-Resolution (SR) and video enhancing techniques. Recent works showed that proximal based convex optimization approaches provide a promising direction in video restoration. An important aspect in the definition of a SR model is the metric used in defining the objective function. Most techniques are based on the classical  $l_2$  norm. In this paper we further investigate the use of other norms and their behavior w.r.t. multiple quality evaluation metrics. We show that significant gains of up to 0.5 dB can be obtained when using different norms.

## I. INTRODUCTION

Technological advancements over the past decade led to a significant increase of display resolutions. Ultra High Definition (UHD) videos are now available to the public and can be viewed on modern TVs or even mobile phones with HR displays. This situation creates a lot of interest for Super-Resolution (SR) techniques which enhance the quality of LR legacy video content or provide a valuable tool in scalable transmission systems.

Video Super-Resolution (SR) tackles the problem of reconstructing a Higher Resolution (HR) or quality video from one or multiple Low Resolution (LR), degraded representations of the video. In general, this problem is approached either by super resolving each frame individually or by using multiple frames from one or more video sequences in order to further enhance the quality of a frame by exploiting the additional information contained in videos. The simplest SR methods use filter based interpolation. Some of the most popular interpolation methods are bicubic [1] or Lanczos [2]. Other filters were also developed as a part of the scalable video compression standards such as the AVC/H.264 and HEVC extensions SVC [3] and SHVC [4], respectively.

More complex SR methods can be generally divided in model based and learning based methods. The most popular learning/example based methods usually rely on finding and storing correlations between LR and HR representations [5]–[9]. This is achieved by creating a dictionary of LR and HR image patches and learning a mapping between them from a

large number of samples. If trained on a sufficient amount of data, this type of methods can provide very good results and an efficient computational time. However, taking into account multiple degradation models (e.g. different smoothing levels or compression artifacts) is relatively tricky, as using too many degradation models in the training may decrease the mapping precision. Other learning based SR methods use convolutional neural networks [10]. Kappeler *et al.* [11] train a convolutional neural network to take 3 consecutive frames of a video sequence as input in order to perform the up-sampling of the middle frame.

Model based SR, unlike example based methods attempt to model the degradation process of the image and impose a series of constraints based on prior knowledge or assumptions about the image. The so obtained model is then used to retrieve a HR image that best suits this objective function given the known LR image. One of the most popular approaches for Multi Frame SR (MF-SR) is based on a Bayesian formulation with SR priors based on smoothness with Total Variation (TV) [12],  $l_1$  norm based priors [13] or non-stationary image prior combinations [14]. However these methods are best suited for image registration where a high number of descriptions are available with a rather simple motion between them. In [15] Liu and Sun extend the Bayesian formulation to videos.

A different approach for image SR and reconstruction which shows great promise is based on proximal dual splitting convex optimization [16]. In our recent work [17] we extended this method to intra coded HEVC videos and in [18] we defined a framework that can combine multiple descriptions of a video using individual compression and degradation models.

Most model based SR methods rely on iterative approaches to find a solution that best fits the model. A critical aspect is the metric used to measure the closeness of the solution to the model, as this determines the changes performed during each iteration. More precisely, in proximal approaches, the function/metric determines the computation of the proximity operator of the objective function. While the  $l_2$  norm is expected to maximize the PSNR of the result, it is a well known fact that PSNR is not always the best measure of quality. As such, using other norms may provide better results on other quality evaluation metrics in certain cases.

This paper investigates the use of different norms in proximal approaches for SR, in order to determine the impact

on different quality evaluation metrics and establish which norm is best suited for such an approach. We perform a large number of tests using 5 different norms and up to 11 metrics for various descriptions of multiple videos. The rest of this paper is organized as follows. In Section II we provide a brief description of the key aspects of the framework we use in our tests. Section III describes the computation of proximity operators for different norms. Experimental results are provided in Section IV while Section V concludes the paper.

## II. PROXIMAL BASED VIDEO ENHANCEMENT FRAMEWORK

In this section we provide a summary of the SR resolution framework we use. Note that any proximal approach method can be used, however, we relate here to our recent work [18] due to the encouraging results obtained and the flexibility of the framework. For a more detailed description of the approach we refer the reader to [16] for a description of image SR and our recent work [18] for a detailed description of the complete framework used in these experiments.

We begin by denoting with  $\bar{x} = [\bar{x}_1, \dots, \bar{x}_k]$  with  $\forall i \in [1, k], \bar{x}_i \in \mathbb{R}^N$  the original HR video sequence which we want to reconstruct. This sequence is subjected to a down-sampling and compression process. The down-sampling operator is denoted by  $L$  and involves a typical decimation and filtering step. For example, the most common filters in literature, used for image down-sampling, are known as bicubic [1] or Lanczos [2] filters. Furthermore, to account for the compression step we need to relate to the typical hybrid video coders architecture depicted here in Fig. 1. The transform operator  $T$  denotes a

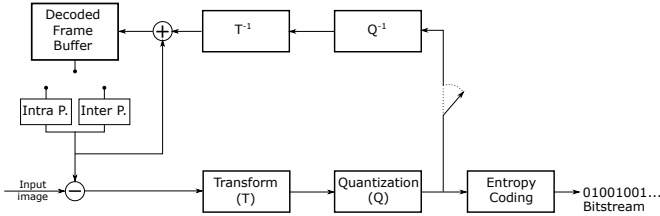


Fig. 1. Hybrid video coding generic scheme.

linear transformation as performed by all hybrid video coders, while  $Q$  denotes a quantization operator. As in this work we use High Efficiency Video Coding standards (HEVC) we will not go into more details and refer the reader to [19] for a more detailed overview of the HEVC including the prediction tools, transforms and quantization process which are of significance to our framework. As such, we denote by  $z$  the quantized transform coefficients obtained for the residual of a frame and with  $\tilde{x}$  the prediction of a frame. Note that  $\tilde{x}$  can be intra or inter predicted. Using the above notations and taking into account the possibility of having  $M$  input videos, we can now formulate the data fidelity (DF) function as:

$$J_{\text{DF}}(\mathbf{x}) = \sum_{m=1}^M \frac{1}{M} \|T_m(L_m \hat{x} - \tilde{x}_m) - z_m\|_p, \quad (1)$$

where  $\hat{x}$  denotes the reconstructed frame and  $\|\cdot\|_p$  is the  $l_p$ -norm. This criterion is further combined with additional

constraints based on the available information. First we use the reconstruction intervals of the quantized coefficients denoted by the convex set  $C$  to impose an additional constraint as:

$$\sum_{m=1}^M \left( \iota_{C_m}(T_m(L_m \tilde{x} - \tilde{x}_m)) \right), \quad (2)$$

where  $\iota_{C_m}$  is the characteristic function of the convex set  $C_m$ , defined as:

$$\iota_C(x) = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{otherwise.} \end{cases} \quad (3)$$

Other constraints used in this model are based on Total Variation (TV) and a limitation of the solution to  $[0, 255]$ . They are denoted by  $F_1$  and  $F_2$  respectively, and defined as:

$$F_1 = (\nabla_h, \nabla_v), \quad (4)$$

$$D_1(m, i) = \{x \in \mathbb{R}^N : \sqrt{\nabla_h^2 x + \nabla_v^2}, x \leq \eta\}, \quad (5)$$

and

$$F_2 = \text{Id}, \quad (6)$$

$$D_2(m) = \{x \in \mathbb{R}^N : x^{(k)} \in [0, 255] \forall k \in [1, N]\}, \quad (7)$$

where  $\nabla_h$  and  $\nabla_v$  are the horizontal and vertical gradients and  $\eta$  is a TV threshold.

Finally the objective function can be expressed as:

$$\text{Find } \hat{x} \in \underset{x \in \mathbb{R}^{K \times N}}{\text{argmin}} \left( J_{\text{DF}}(x) + \sum_{m=1}^M \left( \iota_{C_m}(T_m(L_m x - \tilde{x}_m)) \right) + \sum_{m=1}^M \left( \sum_{s=1}^2 \iota_{D_s(m)}(F_s x) \right) \right), \quad (8)$$

where  $\iota_{D_s(m)}$  is the characteristic function of the convex sets  $D_s$  for each video.

The above problem is solved using an adaptation of the proximal primal-dual algorithm in [20]. A detailed presentation of the algorithm is available in [18]. For the sake of brevity we will not go into additional details regarding the implementation. We simply note that the algorithm is based on successive computations of the proximity operator [21]. The following section will detail the proximity operator and its computation on various norms used in this work.

## III. PROXIMITY OPERATORS

The proximity operator in a real Hilbert space  $\mathcal{H}$  with norm  $\|\cdot\|$  for a function  $\varphi \in \Gamma_0(\mathcal{H})$  where,  $\Gamma_0(\mathcal{H})$  denotes the class of proper lower semi-continuous convex functions from  $\mathcal{H}$  to  $]-\infty, +\infty]$  is defined as:

$$\text{prox}_{\varphi}: \mathcal{H} \rightarrow \mathcal{H}: u \mapsto \underset{v \in \mathcal{H}}{\text{argmin}} \frac{1}{2} \|v - u\|^2 + \varphi(v). \quad (9)$$

In this work we want to compute the proximity operator for different norms used in the DF term of the objective function. This problem was extensively studied and explicit expressions

and computation methods are available for a large number of functions [22], [23].

The objective function described in Section II using the algorithm presented in [18] requires successive computations of the proximity operator of a function  $\varphi_p(y) = \gamma \cdot \|y\|_p$  where  $\gamma \in \mathbb{R}$ ,  $\|\cdot\|_p$  is the  $l_p$ -norm,  $y = T_m(L_m\hat{x} - \tilde{x}_m) - z_m$  and  $\hat{x}$  and  $\tilde{x}$  are single-band 8-bit images. In this work we extend the original framework and in addition to the commonly used  $l_2$  norm we use  $l_1$ ,  $l_{4/3}$ ,  $l_{3/2}$  and  $l_4$  norms. The proximity operators of  $\varphi_p(y)$  in explicit form for  $p = 1, 4/3, 3/2, 2, 4$  are:

$$\text{prox}_{\varphi_1}(y) = \max\{(|y| - \gamma), 0\} \cdot \text{sign}(y), \quad (10)$$

where  $|\cdot|$  denotes the absolute value.

$$\tau = \sqrt{256 \cdot \gamma^3 + 729 \cdot y^2}, \quad (11)$$

$$\text{prox}_{\varphi_{4/3}}(y) = \frac{((108 \cdot |y| + 4 \cdot \tau)^{\frac{2}{3}} - 16 \cdot \gamma)^3}{864} \cdot \text{sign}(y), \quad (12)$$

$$\text{prox}_{\varphi_{3/2}}(y) = y + \frac{9}{8} \cdot \gamma \cdot \text{sign } y \cdot \left( \gamma - \sqrt{\gamma^2 + \frac{16}{9} \cdot |y|} \right) \quad (13)$$

$$\text{prox}_{\varphi_2}(y) = \frac{y}{1 + 2 \cdot \gamma} \quad (14)$$

$$\beta = \sqrt{y^2 + \frac{1}{27 * \gamma}}, \quad (15)$$

$$\text{prox}_{\varphi_{4/3}}(y) = \frac{((y + \beta)^{\frac{1}{3}} - (\beta - y)^{\frac{1}{3}})}{(8 \cdot \gamma)^{\frac{1}{3}}}, \quad (16)$$

Based on the above explicit expressions we can efficiently compute the proximity operator for different norms. Note, that this only applies to the DF term. The additional constraints use the same computation method detailed in [18].

## IV. EXPERIMENTS

### A. Experimental setup

1) *Input data*: We test the video enhancement framework using 6 CIF video sequences (Akiyo, Foreman, Bus, Mobile, Football, Flower) containing both smooth and complex textures. Two descriptions are generated for each sequence using a bicubic polyphase filter with 4-taps defined on  $[-2, 2]$  [1] and an 8-tap extension of the filter obtained by stretching the function to  $[-4, 4]$ . The encoding of each LR description (QCIF) is performed using HEVC [24]. The prediction ( $\tilde{x}$ ) is extracted directly from the stream at the time of decoding. Due to the algorithms constant performance across frames we limit the tests to 1 GOP of 8 frames in an IP configuration. Four compression levels are used by setting the QPs to 1, 15, 20 and 25. Note, that videos compressed using a QP of 1 are almost un-degraded by compression.

Metric	Description	Metric	Description
SAD	Sum of Absolute Difference	PSNR	Peak Signal to Noise Ratio
SSIM	Structural SIMilarity index	MSSIM	Multi-scale SSIM
VSNR	Visual Signal to Noise Ratio	VIF	Visual Information Fidelity
VIFP	pixel-based VIF	UQI	Universal Quality Index
IFC	Information Fidelity Criterion	NQM	Noise Quality Measure
WSNR	Weighted Signal to Noise Ratio		

TABLE I  
VIDEO QUALITY EVALUATION METHODS USED IN OUR EXPERIMENTS.

2) *Algorithm setup*: The proximal based Convex Optimization solver denoted by *CO* is then ran on pairs of descriptions using the same compression and different down-sampling operators. The TV threshold  $\eta$  is defined as:

$$\eta = \eta_0 \cdot TV(x_0) \quad (17)$$

where  $x_0$  is the initialization of the algorithm and  $\eta_0 \in \mathbb{R}$  is used to adjust the threshold. To better emphasize the behavior of the DF function we use two values for  $\eta_0$ , first an empirically determined value of 0.95 which was shown to provide overall good results and a value of 1.1 which relaxes the smoothness constraint. The initialization is computed as the average of the two up-sampled descriptions. For reference we also include the results obtained using bicubic (Bic) up-sampling and one of the best performing learning based SR methods [9] (A+SR). Furthermore, we limit the number of iterations to 200. We empirically found that the solution can be determined with a per-pixel precision of  $\sim 10^{-4}$  for 200 iterations. Albeit, most of the gains are obtained in the first 50-100 iterations as shown in [18], we are interested in studying the behavior of the algorithm when using different norms which requires a more precise computation.

3) *Video quality evaluation*: Using only a Mean Square Error based quality metric such as PSNR would most likely be biased towards the  $l_2$ -norm. Thus, we evaluate our results using multiple video quality evaluation methods including metrics that take into account the particularities of the human visual system. Table I shows the quality evaluation methods used in these experiments. For additional information on each quality evaluation method and implementation details we refer the reader to [25].

### B. Experimental results

As discussed in Section IV-A1 we used 4 QPs, 6 sequences, 5 norms each with 2 TV thresholds. Due to the limited space available we only include the complete tables for two representative sequences: Foreman and Flower. Foreman sequence contains smooth texture while Flower sequence has high frequency components. The other sequences can be viewed as in-between from a texture complexity point of view. For the full results please download the supplementary materials from [https://purica.wp.imt.fr/files/2017/06/MMSP2017\\_supplementary\\_materials.pdf](https://purica.wp.imt.fr/files/2017/06/MMSP2017_supplementary_materials.pdf). Tables IV and V show the numerical results obtained on Akiyo and Flower sequences.

QP	SAD	PSNR	SSIM	MSSIM	VSNR	VIF	VIFP	UQI	IFC	NQM	WSNR
1	$l_1$	$l_1$	$l_1$	$l_1$	$l_1$	$l_1$	$l_{4/3}$	$l_1$	$l_{3/2}$	$l_2$	$l_{3/2}$
15	$l_4$	$l_4$	$l_4$	$l_4$	$l_4$	$l_4$	$l_2$	$l_2$	$l_4$	$l_2$	$l_4$
20	$l_2$	$l_2$	$l_2$	$l_2$	$l_4$	$l_2$	$l_{3/2}$	$l_2$	$l_2$	$l_2$	$l_4$
25	$l_{3/2}$	$l_{3/2}$	$l_{3/2}$	$l_{3/2}$	$l_2$	$l_{3/2}$	$l_{4/3}$	$l_{3/2}$	$l_{3/2}$	$l_2$	$l_2$

TABLE II  
BEST PERFORMING NORMS FOR EACH METRIC AND QP.

Each table is divided in 3 parts. The first part contains the reference results obtained with bicubic up-sampling and the A+SR learning based SR method [9]. The second part depict the results obtained with the Convex Optimization (CO) video enhancement framework described in [18] when using different norms and a TV threshold defined as in Eq. 17 using  $\eta_0 = 0.95$  while the third part uses  $\eta_0 = 1.1$  (i.e. relaxes the smoothness constraint). The light colors green, cyan, red and blue correspond to QPs 1, 15, 20 and 25, respectively. Saturated colors indicate the best result obtained for a fixed metric and QP over the different norms in each block of the table. To interpret these results and identify behavioral trends with respect to different norms in the proximal based CO algorithm we have several possibilities. We begin by analyzing the results with respect to compression levels. In doing so, we can easily notice that the best results for QP 1 (almost no compression) are generally grouped for all tested metrics. This seems to indicate that the quality evaluation methods score the artifacts introduced by filter smoothing in a similar manner as opposed to compression artifacts where the best results generally vary between different metrics. Another interesting aspect to note is that the best results for QP 1 are generally obtained using  $l_1$  norm. However, the results for QP 1 are generally very close between norms. A more significant differentiation is obtained for higher compression levels.

A second way to interpret the results is related to the behavior of the CO method w.r.t. different norms and texture complexity. We can easily notice that on Flower sequence (high texture complexity) the best results are obtained for  $l_4$  norm while for the smooth texture in Foreman sequence the best performance seems to be obtained towards  $l_1$  norm. Furthermore, the different thresholds used in the TV constraint also plays an important role. For example on Foreman sequence we can see that relaxing this constraint will shift the best results towards  $l_1$ , a similar behavior can also be observed on other sequences. It is interesting to note that SAD, SSIM, VIFP, UQI metrics generally tend to obtain higher scores for lower power norms on higher QPs, while PSNR and WSNR tend to be closer to  $l_2$  norm.

Note, that if the DF function is used without any constraint MSE based metrics are always optimized for  $l_2$  norm. While DF with  $l_2$  norm is maximizing the PSNR of the result, the other constraints which are independent of DF are “pulling” the solution in a different direction. Furthermore, the skipped blocks do not allow the precise recovery of quantized values and the results may convergence towards a somewhat less precise value making it harder to isolate the gains or losses produced by the changing of the norm. Using  $l_2$  norm, even if it maximizes the PSNR it will do so w.r.t. the value reconstructed from the intra/inter prediction without the residual,

SAD	PSNR	SSIM	MSSIM	VSNR	VIF
+167279	+0.51	+0.0112	+0.0021	+1.15	+0.0363
VIFP	UQI	IFC	NQM	WSNR	
+0.0179	+0.0172	+0.571	+1.59	+2.10	

TABLE III  
HIGHEST GAINS THAT CAN BE OBTAINED OVER THE  $l_2$  NORM.

thus the behavior is somewhat unpredictable when evaluating the quality against the real values.

For all sequences, we notice that the best results for medium to high compression levels tend to be obtained for  $l_{3/2}$  norm. However, textures with high frequency components present the best results towards a high power norm (Flower,  $l_4$  norm). When compression is very low we notice that the best results are obtained for  $l_1$  norm and in the case of complex textures best results are achieved with a relaxed TV constraint which allows the reconstruction of high frequency components. Note, that the reconstruction precision is higher for low compression as the quantization intervals are very small, for high compression the reconstruction reliability decreases and the model may introduce less accurate high frequency components. Overall, properly adjusting the norm selection and TV thresholding w.r.t. the compression level and texture complexity of the super resolved sequence can provide additional gains over the widely used  $l_2$  norm. Table III shows the maximum gains that can be obtained over  $l_2$  norm and Table II shows the best performing norms for each metric and QP.

## V. CONCLUSIONS

In this paper we performed an extensive study on the use of different norms for a proximal based convex optimization algorithm for super resolution and video enhancement. In addition to the commonly used  $l_2$  norm we implemented and tested the  $l_1$ ,  $l_{4/3}$ ,  $l_{3/2}$  and  $l_4$  norms. Extensive test were performed on 6 video sequences compressed with HEVC at QPs 1, 15, 20 and 25 and super-resolved using a proximal based convex optimization algorithm with each norm and two thresholds of the total variation constraint. 11 metrics were used to evaluate the quality of each approach and 2 additional methods were included as reference: bicubic and A+SR [9]. Our results indicate a strong relation between norms, texture complexity and compression. Gains of up to 0.5 dB can be achieved over the  $l_2$  norm.

## REFERENCES

- [1] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 6, December 1981.
- [2] K. Turkowski and S. Gabriel, “Filters for common resampling tasks,” *Andrew S. Glassner Graphics Gems I*, Academic Press, pp. 147–165, 1990.
- [3] 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.

Method	QP	Metrics										
		SAD	PSNR	SSIM	MSSIM	VSNR	VIFP	UQI	IFC	NQM	WSNR	
<i>Bic</i>	1	301998.75	32.14	0.9397	0.9928	30.06	0.7309	0.6270	0.8204	6.034	32.90	44.70
	15	321895.63	32.01	0.9315	0.9907	29.64	0.6788	0.6031	0.7677	4.681	31.27	43.88
	20	345672.63	31.78	0.9204	0.9876	28.97	0.6114	0.5730	0.7233	3.887	29.52	42.48
	25	389785.50	31.25	0.8998	0.9805	27.43	0.5036	0.5193	0.6576	2.998	26.69	39.64
<i>A + SR</i>	1	257530.25	33.27	0.9546	0.9948	32.55	0.8250	0.6924	0.8551	6.688	33.28	45.14
	15	280684.50	33.11	0.9453	0.9925	31.98	0.7551	0.6610	0.7940	5.070	31.46	44.22
	20	308158.00	32.80	0.9323	0.9894	31.04	0.6683	0.6215	0.7433	4.138	29.59	42.69
	25	358227.88	32.07	0.9089	0.9819	28.77	0.5369	0.5541	0.6695	3.126	26.69	39.74
$CO_{l_1}^{\eta_0=0.95}$	1	183231.00	38.65	0.9683	0.9968	37.44	0.8663	0.7440	0.8756	7.563	39.52	56.88
	15	246176.63	36.06	0.9443	0.9926	33.36	0.7471	0.6667	0.7831	5.167	33.45	48.49
	20	284672.50	34.74	0.9302	0.9893	31.47	0.6570	0.6191	0.7290	4.167	30.67	44.86
	25	344488.13	33.01	0.9071	0.9817	28.96	0.5309	0.5497	0.6587	3.148	27.05	40.45
$CO_{l_{4/3}}^{\eta_0=0.95}$	1	183357.13	38.65	0.9682	0.9968	37.44	0.8660	0.7438	0.8754	7.555	39.52	56.85
	15	244354.50	36.13	0.9451	0.9928	33.48	0.7522	0.6686	0.7869	5.219	33.61	48.76
	20	283245.13	34.80	0.9308	0.9895	31.62	0.6615	0.6205	0.7331	4.198	30.80	45.07
	25	343674.50	33.05	0.9075	0.9819	29.03	0.5334	0.5502	0.6627	3.160	27.18	40.59
$CO_{l_{3/2}}^{\eta_0=0.95}$	1	183429.75	38.64	0.9682	0.9968	37.43	0.8659	0.7437	0.8752	7.550	39.52	56.84
	15	243866.25	36.16	0.9454	0.9929	33.54	0.7535	0.6689	0.7879	5.228	33.64	48.84
	20	283149.13	34.83	0.9309	0.9895	31.66	0.6624	0.6202	0.7341	4.199	30.82	45.13
	25	343878.88	33.07	0.9074	0.9819	29.06	0.5337	0.5495	0.6634	3.159	27.21	40.64
$CO_{l_2}^{\eta_0=0.95}$	1	183639.13	38.64	0.9681	0.9968	37.43	0.8654	0.7434	0.8746	7.530	39.47	56.78
	15	243543.25	36.22	0.9457	0.9929	33.65	0.7545	0.6682	0.7893	5.223	33.66	48.96
	20	284850.25	34.84	0.9302	0.9895	31.79	0.6607	0.6160	0.7339	4.177	30.83	45.21
	25	347969.63	33.02	0.9056	0.9818	29.06	0.5301	0.5427	0.6619	3.131	27.23	40.68
$CO_{l_4}^{\eta_0=0.95}$	1	184018.25	38.60	0.9679	0.9968	37.38	0.8646	0.7435	0.8737	7.498	39.34	56.61
	15	248833.50	36.17	0.9445	0.9928	33.83	0.7464	0.6564	0.7863	5.091	33.54	48.95
	20	303273.88	34.31	0.9239	0.9886	31.61	0.6369	0.5868	0.7241	3.964	30.68	45.14
	25	373245.88	32.30	0.8958	0.9804	28.39	0.5054	0.5114	0.6477	2.957	27.11	40.61
$CO_{l_1}^{\eta_0=1.1}$	1	207191.88	37.57	0.9598	0.9956	35.94	0.8659	0.7369	0.8653	7.656	40.24	55.19
	15	266382.88	35.63	0.9387	0.9922	32.97	0.7583	0.6598	0.7818	5.151	33.67	48.46
	20	303294.00	34.43	0.9248	0.9890	31.19	0.6661	0.6128	0.7308	4.139	30.79	44.93
	25	361586.25	32.80	0.9012	0.9814	28.75	0.5344	0.5436	0.6592	3.109	27.14	40.53
$CO_{l_{4/3}}^{\eta_0=1.1}$	1	207850.25	37.54	0.9595	0.9955	35.90	0.8653	0.7364	0.8651	7.672	40.30	55.20
	15	266281.63	35.66	0.9386	0.9923	33.04	0.7590	0.6593	0.7812	5.182	33.74	48.62
	20	303752.88	34.45	0.9242	0.9891	31.30	0.6665	0.6111	0.7294	4.162	30.86	45.06
	25	362290.50	32.82	0.9003	0.9814	28.85	0.5349	0.5415	0.6581	3.123	27.22	40.63
$CO_{l_{3/2}}^{\eta_0=1.1}$	1	208106.88	37.52	0.9594	0.9955	35.88	0.8648	0.7360	0.8649	7.672	40.31	55.19
	15	266612.88	35.67	0.9384	0.9923	33.05	0.7586	0.6585	0.7808	5.177	33.74	48.64
	20	304655.13	34.45	0.9236	0.9890	31.32	0.6654	0.6094	0.7281	4.157	30.86	45.08
	25	363647.63	32.80	0.8994	0.9813	28.88	0.5339	0.5393	0.6571	3.120	27.23	40.65
$CO_{l_2}^{\eta_0=1.1}$	1	208677.25	37.49	0.9591	0.9955	35.86	0.8637	0.7349	0.8648	7.654	40.33	55.19
	15	268096.00	35.68	0.9376	0.9922	33.07	0.7558	0.6556	0.7792	5.137	33.73	48.65
	20	309459.75	34.39	0.9211	0.9887	31.32	0.6587	0.6018	0.7239	4.100	30.82	45.07
	25	372074.13	32.67	0.8952	0.9809	28.77	0.5262	0.5285	0.6519	3.069	27.20	40.64
$CO_{l_4}^{\eta_0=1.1}$	1	209295.63	37.48	0.9593	0.9957	35.94	0.8602	0.7318	0.8644	7.561	40.07	55.33
	15	275642.63	35.61	0.9334	0.9918	33.03	0.7434	0.6421	0.7706	4.966	33.54	48.54
	20	333228.38	33.80	0.9088	0.9873	31.07	0.6263	0.5663	0.7062	3.819	30.61	44.90
	25	406125.38	31.66	0.8782	0.9788	28.04	0.4913	0.4865	0.6311	2.824	27.05	40.49

TABLE IV

RESULTS OBTAINED ON FOREMAN SEQUENCE USING 4 QPS, 5 NORMS EACH WITH 2 TV THRESHOLDS AND 11 OF THE MOST POPULAR VIDEO QUALITY EVALUATION TECHNIQUES.

- [4] 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 3 and ISO/IEC JTC 1/SC 29/WG 11, June 2015.
- [5] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," *2009 IEEE 12th International Conference on Computer Vision*, pp. 349–356, September 2009.
- [6] R. Zeyde, M. Elad, and M. Protter, *On single image scale-up using sparse-representations*, ser. Curves and Surfaces, J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. Schumaker, Eds. Springer Berlin Heidelberg, 2012, vol. 6920.
- [7] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *2013 IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 561–568.
- [8] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Single-image super-resolution via linear mapping of interpolated self-examples," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5334–5347, December 2014.
- [9] 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.
- [10] C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8692. Springer International Publishing, 2014, pp. 184–199.
- [11] 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.
- [12] S. Babacan, R. Molina, and A. Katsaggelos, "Variational bayesian super resolution," *IEEE Transactions in Image Process*, pp. 984–999, 2011.
- [13] 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.
- [14] S.Villena, M.Vegaa, R.Molinab, and A.K.Katsaggelosc, "A non-

Method	QP	Metrics										
		SAD	PSNR	SSIM	MSSIM	VSNR	VIFP	UQI	IFC	NQM	WSNR	
<i>Bic</i>	1	1011859.50	22.97	0.8288	0.9734	19.44	0.3859	0.3296	0.5905	3.600	26.76	39.32
	15	1019696.00	22.96	0.8262	0.9725	19.40	0.3769	0.3243	0.5288	3.135	26.16	39.21
	20	1027394.13	22.94	0.8225	0.9711	19.30	0.3653	0.3184	0.5076	2.926	25.69	39.01
	25	1049480.63	22.84	0.8125	0.9673	18.91	0.3379	0.3040	0.4765	2.605	24.36	38.29
<i>A + SR</i>	1	974043.63	23.19	0.8640	0.9797	19.93	0.4097	0.3511	0.6525	3.792	26.51	39.72
	15	982369.63	23.19	0.8611	0.9788	19.90	0.3989	0.3437	0.5701	3.253	25.97	39.60
	20	989709.13	23.18	0.8570	0.9775	19.83	0.3860	0.3364	0.5456	3.041	25.52	39.38
	25	1007933.00	23.13	0.8463	0.9740	19.49	0.3577	0.3201	0.5119	2.726	24.23	38.64
$CO_{l_1}^{\eta_0=0.95}$	1	680159.63	26.54	0.9248	0.9909	25.23	0.5815	0.4467	0.6890	5.632	36.00	49.88
	15	742828.63	25.94	0.9045	0.9870	23.71	0.5071	0.4103	0.6044	4.221	31.06	46.40
	20	817470.88	25.18	0.8832	0.9828	22.28	0.4464	0.3776	0.5626	3.545	28.83	43.70
	25	920409.50	24.16	0.8545	0.9760	20.70	0.3766	0.3357	0.5108	2.880	25.59	40.50
$CO_{l_{4/3}}^{\eta_0=0.95}$	1	680144.63	26.54	0.9248	0.9909	25.23	0.5814	0.4467	0.6892	5.631	36.01	49.88
	15	740399.00	25.96	0.9055	0.9872	23.73	0.5088	0.4117	0.6085	4.246	31.18	46.46
	20	810957.25	25.23	0.8859	0.9832	22.35	0.4501	0.3806	0.5699	3.582	28.97	43.83
	25	909720.63	24.24	0.8585	0.9767	20.80	0.3818	0.3395	0.5222	2.922	25.76	40.71
$CO_{l_{3/2}}^{\eta_0=0.95}$	1	680141.75	26.54	0.9248	0.9909	25.24	0.5815	0.4468	0.6892	5.632	36.00	49.88
	15	739244.00	25.97	0.9060	0.9872	23.74	0.5097	0.4123	0.6099	4.257	31.20	46.49
	20	807800.50	25.26	0.8870	0.9834	22.40	0.4520	0.3819	0.5718	3.599	29.04	43.90
	25	904595.38	24.29	0.8602	0.9770	20.86	0.3842	0.3411	0.5246	2.943	25.82	40.80
$CO_{l_2}^{\eta_0=0.95}$	1	680124.88	26.54	0.9248	0.9909	25.23	0.5815	0.4468	0.6891	5.632	35.99	49.88
	15	735804.50	26.00	0.9071	0.9874	23.80	0.5126	0.4141	0.6123	4.294	31.28	46.60
	20	797675.75	25.36	0.8902	0.9839	22.55	0.4587	0.3859	0.5760	3.659	29.18	44.16
	25	889103.25	24.45	0.8643	0.9776	21.06	0.3910	0.3455	0.5280	2.994	26.04	41.08
$CO_{l_4}^{\eta_0=0.95}$	1	679845.50	26.54	0.9248	0.9909	25.24	0.5817	0.4470	0.6893	5.636	36.01	49.89
	15	727505.00	26.09	0.9099	0.9878	24.02	0.5207	0.4181	0.6150	4.379	31.35	46.83
	20	781279.50	25.55	0.8944	0.9847	23.14	0.4734	0.3902	0.5793	3.772	29.24	44.76
	25	899429.38	24.34	0.8623	0.9773	21.19	0.3912	0.3397	0.5271	2.992	26.04	41.33
$CO_{l_1}^{\eta_0=1.1}$	1	650478.75	27.05	0.9303	0.9921	26.28	0.6178	0.4647	0.6835	6.198	37.50	51.98
	15	748579.88	25.99	0.9053	0.9874	23.85	0.5131	0.4117	0.5953	4.279	31.34	46.62
	20	833752.63	25.09	0.8840	0.9833	22.38	0.4492	0.3769	0.5637	3.551	28.87	43.74
	25	942759.88	24.03	0.8532	0.9760	20.65	0.3762	0.3328	0.5097	2.859	25.52	40.44
$CO_{l_{4/3}}^{\eta_0=1.1}$	1	650655.38	27.05	0.9302	0.9921	26.28	0.6178	0.4645	0.6836	6.203	37.59	51.98
	15	746551.50	26.01	0.9060	0.9875	23.88	0.5162	0.4130	0.5992	4.320	31.49	46.74
	20	828081.38	25.14	0.8855	0.9835	22.43	0.4532	0.3790	0.5678	3.593	29.07	43.91
	25	934682.25	24.11	0.8551	0.9765	20.74	0.3810	0.3355	0.5137	2.902	25.76	40.67
$CO_{l_{3/2}}^{\eta_0=1.1}$	1	650743.38	27.05	0.9302	0.9921	26.28	0.6177	0.4645	0.6832	6.202	37.59	51.98
	15	745619.13	26.02	0.9063	0.9876	23.90	0.5176	0.4135	0.6004	4.337	31.54	46.80
	20	825243.88	25.17	0.8862	0.9837	22.46	0.4554	0.3800	0.5693	3.614	29.12	44.00
	25	930164.75	24.15	0.8560	0.9767	20.79	0.3834	0.3366	0.5161	2.922	25.83	40.77
$CO_{l_2}^{\eta_0=1.1}$	1	651026.63	27.05	0.9301	0.9920	26.27	0.6173	0.4642	0.6812	6.188	37.52	51.96
	15	743633.63	26.04	0.9069	0.9877	23.97	0.5211	0.4145	0.6031	4.378	31.62	46.94
	20	816152.50	25.28	0.8878	0.9840	22.67	0.4623	0.3829	0.5715	3.674	29.28	44.29
	25	916375.13	24.31	0.8586	0.9772	21.00	0.3894	0.3395	0.5212	2.967	26.02	41.07
$CO_{l_4}^{\eta_0=1.1}$	1	651641.00	27.04	0.9301	0.9920	26.19	0.6145	0.4636	0.6785	6.126	37.28	51.79
	15	739231.25	26.10	0.9076	0.9878	24.07	0.5268	0.4162	0.6058	4.429	31.75	47.12
	20	796836.50	25.57	0.8877	0.9839	23.15	0.4744	0.3855	0.5695	3.748	29.34	44.86
	25	926938.13	24.19	0.8552	0.9762	21.06	0.3864	0.3321	0.5192	2.935	25.98	41.27

TABLE V

RESULTS OBTAINED ON FLOWER SEQUENCE USING 4 QPS, 5 NORMS EACH WITH 2 TV THRESHOLDS AND 11 OF THE MOST POPULAR VIDEO QUALITY EVALUATION TECHNIQUES.

- stationary image prior combination in super-resolution,” *Digital Signal Processing*, vol. 32, 2014.
- [15] 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.
- [16] R. Gaetano, B. Pesquet-Popescu, and C. Chau, “A convex optimization approach for image resolution enhancement from compressed representations,” in *18th International Conference on Digital Signal Processing (DSP)*, July 2013, pp. 1–8.
- [17] 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.
- [18] A. Purica, B. Boyadjis, B. Pesquet-Popescu, F. Dufaux, and C. Bergeron, “A convex optimization framework for video quality and resolution enhancement from multiple descriptions,” *IEEE Transactions on Image Processing (submitted to)*, 2017.
- [19] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.
- [20] 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.
- [21] J.-J. Moreau, “Proximité et dualité dans un espace hilbertien,” *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.
- [22] 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.
- [23] C. Chau, 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.
- [24] J. video team (JVT) of ISO/IEC MPEG & ITU-T VCEG, “H.265/HEVC HM reference software,” <http://hevc.fraunhofer.de/>, May 2013.
- [25] Metrix mux. [http://ollie-imac.cs.northwestern.edu/ollie/GMM/code/metrix\\_mux/](http://ollie-imac.cs.northwestern.edu/ollie/GMM/code/metrix_mux/).