Unified Single-Image and Video Super-Resolution via Denoising Algorithms

Alon Brifman, Yaniv Romano, and Michael Elad, *Fellow, IEEE*

***Abstract*—Single Image Super-Resolution (SISR) aims to recover a high-resolution image from a given low-resolution version of it. Video Super Resolution (VSR) targets series of given images, aiming to fuse them to create a higher resolution outcome. Although SISR and VSR seem to have a lot in common, most SISR algorithms do not have a simple and direct extension to VSR. VSR is considered a more challenging inverse problem, mainly due to its reliance on a sub-pixel accurate motion-estimation, which has no parallel in SISR. Another complication is the dynamics of the video, often addressed by simply generating a single frame instead of a complete output sequence.**

In this work we suggest a simple and robust super-resolution framework that can be applied to single images and easily extended to video. Our work relies on the observation that *denoising* of images and videos is well-managed and very effectively treated by a variety of methods. We exploit the Plug-and-Play-Prior framework and the Regularization-by-Denoising (RED) approach that extends it, and show how to use such denoisers in order to handle the SISR and the VSR problems using a unified formulation and framework. This way, we benefit from the effectiveness and efficiency of existing image/video denoising algorithms, while solving much more challenging problems. More specifically, harnessing the VBM3D video denoiser, we obtain a strongly competitive motion-estimation free VSR algorithm, showing tendency to a high-quality output and fast processing.

***Index Terms*—Single Image Super-Resolution, Video Super-Resolution, Plug-and-Play-Prior, RED, Denoising, ADMM**

I. Introduction

The single-image super-resolution (SISR) problem assumes that a given measured image $\mathbf{y}$ is a blurred, spatially decimated, and noisy version of a high quality image $\mathbf{x}$. Our goal in SISR is the recovery of $\mathbf{x}$ from $\mathbf{y}$. This is a highly ill-posed inverse problem, typically handled by the Maximum a'posteriori Probability (MAP) estimator. Such a MAP strategy relies on the introduction of an image prior, representing the minus log of the probability density function of images. Indeed, most of the existing algorithms for SISR differ in the prior they use – see for example [1]–[9]. We should mention that convolutional neural networks (CNN) have been brought recently as well to serve the SISR [10]–[12], often times leading to state-of-the-art results.

The Video Super Resolution (VSR) task is very similar to SISR but adds another important complication – the temporal domain. Each frame in the video sequence is assumed to be a blurred, decimated, and noisy version of a higher-resolution original frame. Our goal remains the same: recovery of the higher resolution video sequence from its measured degraded version. However, while this sounds quite close in spirit to the SISR problem, the two are very much different due to the involvement of the temporal domain. Indeed, one might be tempted to handle the VSR problem as a simple sequence of SISR tasks, scaling-up each frame independently. However, this is highly sub-optimal, due to the lack of use of cross relations between adjacent frames in the reconstruction process.

More specifically, the VSR task can be formulated using the MAP estimator in a way that is similar to the formulation of the SISR problem. Such an energy function should include a log-likelihood term that describes the connection between the desired video and the measured one, and a video prior. While the first expression is expected to look the same for SISR and VSR, the video prior is likely to be markedly different, as it should take into account both the spatial considerations, as in the single image case, and add a proper reference to the temporal relations between frames. Thus, although SISR and VSR have a lot in common, the suggested priors for each task differ substantially, and hence SISR algorithms do not tend to have an easy adaptation to VSR.

The gap between the two super-resolution problems explains why VSR algorithms tend to use entirely different methods to tackle their recovery problem, rather than just extending SISR methods. Classic VSR methods commonly turn to explicit subpixel motion-estimation[1] [13]–[25], which has no parallel in SISR. For years it was believed that this ingredient is unavoidable, as fusing the video frames amounts to merging their grids, a fact that implies a need for an accurate sub-pixel registration between these images. Exceptions to the above are the two algorithms reported in [26], [27], which use implicit motion-estimation, and thus are capable of handling more complex video content.

This work's objective is to create a robust joint framework for super resolution that can be applied to SISR and easily be adapted to solve the VSR problem just as well. Our goal is to formulate the two problems in a unified way, and derive a single algorithm that can serve them both. The key to the formation of such a bridge is the use of the Plug-and-Play-Prior (PPP) method [28], and the more recent framework of Regularization-by-Denoising (RED) [29]. Both PPP and RED offer a path for turning any inverse problem into a chain of denoising steps. As such, our proposed solution for SISR and VSR would rely on the vast progress made in the past two decades in handling the image and video denoising problems.

This work presents a novel and highly effective SISR and VSR recovery algorithm that uses top-performing image and video denoisers. More importantly, however, this algorithm has the very same structure and format when handling either a single image or a video sequence, differing only in the

[1]Even CNN based solutions have been relying on such motion estimation and compensation.

deployed denoiser. In our previous work [30] we have successfully harnessed the PPP scheme to solve the SISR problem. In this paper our focus is on extending this formulation to VSR while keeping its architecture. With the recent introduction of the Regularization-by-Denoising (RED) framework [29], we use this as well in the migration from the single image case to video. As demonstrated in the results section, the proposed paradigm is not only simple and easy to implement, but also leads to state-of-the-art results in video super-resolution, favorably competing with the best available alternatives.

We should note that in parallel to the release of our work several VSR methods based on deep-learning were published [31]–[36]. These methods achieve very impressive results, yet CNN based algorithms often need to be re-tuned due to small changes in the restoration task. [36], for example, has different hyper-parameters for different upscaling ratios. A change in the input's size might call for a change in the network's architecture, and a different blur might require separate training. Our unified framework has only a few parameters and is stable for a variety of scale factors, blur kernels and input sizes.

The rest of this paper is organized as follows: Section II introduces the PPP and RED schemes, which play a critical role in this work. Section III presents our suggested framework and its properties, and section IV provides extensive experimental results. Section V concludes this work.

II. Background on PPP and RED

In this section we present the Plug-and-Play-Prior [28] and Regularization-by-Denoising [29] schemes, which are central to our framework. This section mostly follows [28], [29], [37].

A. Plug-and-Play-Prior

Many inverse problems (including the super-resolution ones) are formulated as a MAP estimation, a factored sum of two expressions, or penalties: a data fidelity term (usually the log-likelihood) and a prior function. Such a general inverse problem may appear as follows:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{G}\mathbf{x}-\mathbf{y}\|_2^2 + \beta R\left(\mathbf{x}\right), \tag{1}$$

where $\mathbf{x}$ is the unknown image to be recovered, $\mathbf{y}$ is the measured image, assumed to be a noisy contaminated version of $\mathbf{G}\mathbf{x}$, $\mathbf{G}$ being the degradation operator. The functional $R(\cdot)$ stands for the image prior, and the parameter β multiplying it sets the relative weights between the two penalties.

The Plug-and-Play-Prior (PPP) scheme [28] offers a method to separate the two in a manner that allows us to use prior functions that are already integrated into Gaussian denoising algorithms. Thus, we use the denoiser as a black-box tool, while solving for another, more challenging, inverse problem. Let us illustrate the PPP on the problem posed in Equation (1). Using variable splitting, we can separate the degradation model (the ℓ_2 data fidelity term) from the prior:

$$\begin{aligned}\mathbf{x}^* = &\arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{G}\mathbf{x}-\mathbf{y}\|_2^2 + \beta R(\mathbf{v}),\\ &\text{s.t} \quad \mathbf{x}=\mathbf{v}.\end{aligned} \tag{2}$$

Using the Augmented Lagrangian strategy, the constraint can be turned into an additive penalty,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{G}\mathbf{x}-\mathbf{y}\|_2^2 + \beta R(\mathbf{v}) + \frac{\rho}{2}\|\mathbf{x}-\mathbf{v}+\mathbf{u}\|_2^2, \tag{3}$$

where $\mathbf{u}$ is the scaled Lagrange multipliers vector, and ρ is a parameter to be set[2]. Applying the ADMM [37], we obtain the following iterative scheme to minimize Equation (2):

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{G}\mathbf{x}-\mathbf{y}\|_2^2 + \frac{\rho}{2}\|\mathbf{x}-\mathbf{v}^k+\mathbf{u}^k\|_2^2 \tag{4a}$$

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \beta R(\mathbf{v}) + \frac{\rho}{2}\|\mathbf{x}^{k+1}-\mathbf{v}+\mathbf{u}^k\|_2^2 \tag{4b}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{v}^{k+1} \tag{4c}$$

Notice that Equation (4a) is quadratic in $\mathbf{x}$ and therefore can be solved analytically. Moving to Equation (4b), this can be re-written as

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \beta R(\mathbf{v}) + \frac{1}{2(\frac{1}{\sqrt{\rho}})^2}\|\mathbf{v}-\widetilde{\mathbf{v}}\|_2^2, \tag{5}$$

where $\widetilde{\mathbf{v}} = \mathbf{x}^{k+1}+\mathbf{u}^k$. The above is nothing but a denoising problem, aiming at cleaning the noisy image $\widetilde{\mathbf{v}}$, with $\sigma = \frac{1}{\sqrt{\rho}}$. Hence we can use the denoiser as a black-box tool for solving step (4b), even without having access to the explicit prior $R(\mathbf{v})$ we are relying on.

B. Regularization by Denoising

Like PPP, Regularization by Denoising (RED) is another scheme that integrates denoisers into a reconstruction algorithm in order to solve other inverse problems. Yet, as opposed to the PPP scheme, RED defines an explicit prior, constructed by a chosen denoising algorithm. Then RED solves the MAP estimator with the defined prior in order to achieve a global optimizer (under mild conditions).

More specifically, given a denoising function $f(\mathbf{x})$, RED sets the prior to be $R(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\left(\mathbf{x}-f\left(\mathbf{x}\right)\right)$. This is an image-adaptive Laplacian-based regularization functional, penalizing over the inner product between a signal $\mathbf{x}$ and its denoising residual $\mathbf{x}-f(\mathbf{x})$. Under the assumptions that (i) $\nabla f(\mathbf{x})$ exists (i.e. f is differentiable) and is symmetric, (ii) $f\left(c\mathbf{x}\right) = cf\left(\mathbf{x}\right)$, for $c \to 1$ (local homogeneity), and (iii) the spectral radius of $\nabla f(\mathbf{x})$ is smaller or equal to 1, the authors of [29] prove two key properties:

(i) The gradient of the prior expression is nothing but the denoising residual,

$$R(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T(\mathbf{x}-f(\mathbf{x})) \to \nabla R(\mathbf{x}) = \mathbf{x}-f(\mathbf{x}).$$

(ii) The convexity of the suggested prior (and thus the overall objective) is guaranteed, meaning that the MAP optimization process will yield a global minimizer.

It turns out that various state-of-the-art denoising algorithms satisfy these conditions [29], or nearly so, posing RED as an appealing alternative to the PPP approach.

[2]Often times, it is found beneficial to increase this parameter throughput the iterative algorithm given below.

Equation (1) with the newly introduced prior may be solved using several different strategies, as indeed was done in [29]. One of the proposed strategies uses ADMM, derived in a similar manner to the one in Section II-A. However, this time we have a specific regularizer at hand. Concentrating on step (4b), by plugging the RED regularizer we get

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \frac{\beta}{2}\mathbf{v}^T\left(\mathbf{v} - f\left(\mathbf{v}\right)\right) + \frac{\rho^k}{2}\|\mathbf{x}^{k+1} - \mathbf{v} + \mathbf{u}\|_2^2. \tag{6}$$

Exploiting the relation $\nabla R(\mathbf{x}) = \mathbf{x} - f(\mathbf{x})$, and setting the gradient of this cost function to zero, we obtain

$$\beta\left(\mathbf{v} - f\left(\mathbf{v}\right)\right) + \rho\left(\mathbf{v} - \mathbf{x}^{k+1} - \mathbf{u}\right) = 0, \tag{7}$$

which can be solved by the fixed point strategy, leading to the following update rule for $\mathbf{v}$ (please refer to [29] for more details):

$$\mathbf{v}^{j+1} = \frac{1}{\beta + \rho}\left(\beta f\left(\mathbf{v}^j\right) + \rho\left(\mathbf{x}^{k+1} + \mathbf{u}\right)\right). \tag{8}$$

The signal $\mathbf{v}^j$ is the estimate of the j-th step of the fixed point method that minimizes Equation (6). Notice that this strategy leads to an inner iteration (denoted with index j) in the iteration described in (4) (denoted with index k). As a result, once again, the solution of $\mathbf{v}$ amounts to the application of a denoiser as a black-box tool (possibly for several iterations).[3]

In summary, we have in our hands powerful tools to take denoisers and use them in order to solve other, more involved, inverse problems. Our next step is to harness these to the super resolution problem, both for single images and video.

III. The Proposed Super-Resolution Framework

In this section we present our novel unified formulation of the SISR and the VSR problems, and the common algorithm that served them both. We start by describing the SISR problem, and then move to the video counterpart. Our next step is to describe our PPP/RED based approach to super-resolution reconstruction, and as this relies on denoising algorithms, we precede this by mentioning the existing state-of-the-art denoising methods for stills and video. We conclude this section by discussing computational complexity and convergence issues.

A. The Singe-Image Super-Resolution Problem

The single-image super-resolution (SISR) problem starts with an unknown High-Resolution (HR) image $\mathbf{x} \in \mathbf{R}^{sM\times sN}$ ($s > 1$), of which we are only given a blurred, spatially decimated (in a factor s in each axis), and noisy Low-Resolution (LR) measurement $\mathbf{y} \in \mathbf{R}^{M\times N}$. Our aim is to recover $\mathbf{x}$ from $\mathbf{y}$. This inverse problem can be formulated by the following expression that ties the measurements to the unknown:

$$\mathbf{y} = \mathbf{SHx} + \eta. \tag{9}$$

The matrix $\mathbf{H} \in \mathbf{R}^{s^2MN\times s^2MN}$ blurs the original image, $\mathbf{S} \in \mathbf{R}^{MN\times s^2MN}$ is the down-sampling operator and $\eta \sim N(0;\sigma^2\mathbf{I}) \in \mathbf{R}^{M\times N}$ is an additive zero-mean white Gaussian noise. Note that $\mathbf{x}$, $\mathbf{y}$ and η are all held as column vectors, after lexicographic ordering, that is, they are vectors of length s^2MN and MN respectively, where the columns of the image are concatenated one after another to form a long one-dimensional vector.

The Maximum Likelihood (ML) estimator for this problem is defined by

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{SHx} - \mathbf{y}\|_2^2, \tag{10}$$

yet the ill-posed nature of this problem ($\mathbf{SH}$ is non-invertible) renders this approach useless. Using the MAP estimator instead leads us to minimize the ML, augmented with a predefined prior. That is, Equation (10) should be regularized,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{SHx} - \mathbf{y}\|_2^2 + R\left(\mathbf{x}\right), \tag{11}$$

$R(\mathbf{x})$ being the prior, discriminating "good looking" images from "bad" ones by giving the "good" images a lower numerical value. And so, the quest of the holy grail, so to speak, begins: What is the appropriate prior to use for images? Vast amount of work has been invested in addressing this question. Indeed, most of the existing algorithms for SISR differ in the prior they use, or the numerical method they deploy for solving the presented optimization problem (11). Commonly chosen priors are based on sparse representations [1]–[5], spatial smoothness such as Total Variation [6], self-similarity [7]–[9], and more.

In recent years convolutional neural networks (CNN) have been used for SISR quite successfully [10]–[12]. Observe that this approach bypasses the explicit use of the MAP formulation, replacing it by a direct learning of the recovery process from the input low-resolution image to the desired high-resolution output. One may argue that this supervised approach incorporates the MAP strategy and the prior in it implicitly, by shaping the solver as a minimizer of the MAP energy task.

B. Moving to Video Super-Resolution

The Video Super Resolution (VSR) task may seem to be similar to the SISR problem, but it adds another complicating factor – the temporal domain. Here, the HR video, $\mathbf{x} \in \mathbf{R}^{sM\times sN\times T}$, is composed of T frames that are assumed to have some relation between them, manifested as motion or flow. Often, as introduced in [13] and later used in [14], [16], [22], [23], a warp or motion operator is used to express this relation between the frames,

$$\mathbf{x}^i = \mathbf{F}^i\mathbf{x}^0 + \mathbf{m}^i, \tag{12}$$

where $\mathbf{x}^i$ is the i-th frame, $\mathbf{F}^i$ is the motion operator that takes us from the 0-th frame to the i-th one, and $\mathbf{m}^i$ is new information appearing in $\mathbf{x}^i$, such as new objects entering the frame or changes in lighting. The given low-resolution video $\mathbf{y} \in \mathbf{R}^{M\times N\times T}$ is obtained from $\mathbf{x}$ via the same relation as in Equation (9), where the operators $\mathbf{H}$ and $\mathbf{S}$ apply their

[3] We note that in [29], an alternative scheme to the ADMM was proposed based on a direct Fixed-Point strategy.

degradations on the entire video sequence $\mathbf{x}$, operating on each frame independently[4].

Using the MAP estimator for the VSR task can be formulated exactly as in Equation (11), where $\mathbf{x}$ and $\mathbf{y}$ are now the complete high-resolution and low-resolution video sequences. Here again we face the need to find an appropriate prior that could grade video quality, and as already mentioned in the Introduction, such a prior is expected to be very different from a single image one, due to the need to refer to the temporal relations in Equation (12).

And so, although some of the SISR algorithms mentioned above are considered state-of-the-art, and although SISR and VSR have very similar formulations, none of these known algorithms was adapted to VSR (apart from the trivial adaptation of performing SISR for each frame independently). Put very simply, SISR algorithms do not generally have an easy adaptation to VSR. Indeed, VSR algorithms tend to use entirely different methods to tackle their recovery problem, built around an explicit optical-flow or (subpixel-)motion-estimation process, which has no parallel in SISR. Hence, most of the VSR algorithms are heavily dependent on such highly performing motion-estimation algorithms, a fact that leads to more costly overall recovery processes [13]–[25]. An additional unfortunate by-product of this strategy is an extreme sensitivity of these methods to motion-estimation errors, causing severe artifacts in the resulting frames. As a consequence, classic VSR algorithms are known to be limited in their ability to process videos with only simple and global motion trajectories.

As a side note, we mention the following: The work reported in [26] is the first VSR algorithm to abandon *explicit* motion-estimation by generalizing NLM [38] for super-resolution. The 3DSKR algorithm [27] followed it, replacing the accurate motion-estimation by a multidimensional kernel regression. Both these algorithms and their follow-up work [39] are capable of processing videos with far more complex motion. Still, these methods rely on the computation of weights based on every pixel's neighbourhood or patch, making them computationally heavy. As in the SISR, CNNs have made their appearance to the VSR problem as well [22]–[25], yet these methods too often integrate motion-estimation into their algorithmic process, hence being computationally heavy.

C. Advancements in Denoising Algorithms

We move now to discuss a simpler inverse problem – denoising – the removal of additive noise from images and video. This task is a special case of the SISR and VSR problems, obtained by setting $\mathbf{S} = \mathbf{H} = \mathbf{I}$. While this may appear as a diversion from this paper's main theme, the opposite is true. As we rely on the PPP or the RED schemes to construct an alternative super-resolution reconstruction algorithm, denoising algorithms are central to our work.

Image and video denoising have made great advancements over the past two decades, resulting in highly effective and efficient algorithms. As denoising is the simplest inverse problem, such algorithms clearly encompass in them some sort of a prior knowledge on the image or video, even if used implicitly. In the context of single image denoisers, leading algorithms rely on sparse representations [2], [40]–[42], self-similarity [43], [44], and more [45]–[48]. In addition, highly effective deep learning solutions of this problem are also available [12], [49], [50]. The performance of all these methods (deep-learning-based and others) is so good that recent work investigated the possibility that we are nearing a performance limit [51]–[53].

As expected, the priors used for image denoising are very similar to the ones used in SISR. Yet migrating these denoising algorithms to serve a different problem, such as SISR, is difficult. Consider the NCSR algorithm [2], which is a state-of-the-art denoiser. It was adapted to solve other inverse problems, one of which is the SISR task. However, the code for NCSR-SISR and NCSR-Denoising ships in two different code packages, indicating that the migration from denoising to SISR is not achieved only by a small modification to the degradation model.

The gap between VSR and video denoising is even wider. Video denoisers such as [54]–[60] have already abandoned the explicit motion-estimation algorithms still so commonly used by VSR. This results in very efficient, and highly effective denoisers for video noise removal. In contrast, with the exception of [26], [27], [39], VSR algorithms are left behind, still relying on explicit optical flow estimation.

D. Our Unified Super-Resolution Framework

This work's objective is to create a robust and unified framework for super resolution that can be applied to both SISR and VSR problems. Our goal is to propose a single formulation that covers both cases, leading to a single algorithm that operates on both these problems in the same manner. The path towards achieving this goal passes through the use of the PPP/RED schemes, and this implies that we shall also rely on image/video denoisers. Our starting point is the MAP formulation in Equation (11),

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{SHx} - \mathbf{y}\|_2^2 + R(\mathbf{x}) .$$

We choose to interpret this expression in two different ways. For the single image case, $\mathbf{x}$ and $\mathbf{y}$ are single images, and $R(\mathbf{x})$ is a single image prior. When moving to image sequences, the same equation remains relevant, where this time $\mathbf{x}$ and $\mathbf{y}$ are assumed to be complete image sequences (i.e., two volumetric datasets), and now the blur and the subsampling operations are assumed to apply to each of the frames in the sequence independently. As for $R(\mathbf{x})$, it represents a video prior, able to grade complete video solutions, by taking into account both inter-frame relations (as given in Equation (12)), along with intra-frame dependencies.

Now that the two problems are posed as a common energy minimization objective, the PPP and RED schemes as described in Section II are applicable and relevant to our needs. All that should be done is to replace the general degradation operator $\mathbf{G}$ in Equations (4a)-(4c) by $\mathbf{SH}$. More specifically, Equation (4b) is solved differently in the PPP and the RED schemes, yet both methods use a denoiser to solve this step,

[4]In fact, a spatio-temporal blur can be easily accommodated as well by the algorithms discussed in this paper.

be it a single image denoiser or a video one. Equation (4a) is quadratic in $\mathbf{x}$ and can be solved using simple Linear Algebra algorithms, such as conjugate gradient and similar tools. Therefore, both schemes lead to a sequence of denoising computations, surrounded by simple algebraic operations. To summarize, Equations (4a)-(4c) are relevant to both the single image and the video cases. As we move from SISR to VSR, the only difference is this: *The denoiser to be applied in Equation (4b) should a video denoiser, operating on the complete video volume* $\mathbf{x}^{k+1}+\mathbf{u}^k$ *at once, thus exploiting both inter- and intra redundancies.*

We note that we could have proposed a more classical frame-by-frame MAP estimator for VSR in the spirit of the work reported in [13]. This would have been done by embarking from Equation (11) and inserting the geometrical warp relations shown in Equation (12) into the log-likelihood term. However, the warp operators $\mathbf{F}^i$ are not known in advance, and therefore a preceding step of sub-pixel motion estimation would have been needed in order to estimate them. As a result, such a frame-by-frame VSR framework becomes a sequence of motion estimations followed by single-frame denoising steps. Thus, the classical frame-by-frame approach cannot simply apply the existing SISR framework, but rather solves first another challenging task of motion estimation. The approach we have proposed above overcome all these difficulties by deferring the inter-frame relations into the video prior.

Algorithm 1 formulates the unified super-resolution scheme for images and videos using PPP and RED. The input to the algorithm is a LR image in the SISR case, or the whole LR video in the VSR case (i.e. VSR operates on the whole video at once and not frame-by-frame). The algorithm follows closely Equation (4) and its adaptations discussed in sections II-A and II-B. The last two steps in the for-loop are a small modification to Equation (4), and their goal is to improve the convergence of the scheme. We shall elaborate more on this modification in the following subsection.

Regarding computations and implementation, the $\mathbf{x}$-update step, although given in a closed form, is not computed analytically, since $\mathbf{S}$ and $\mathbf{H}$ are huge matrices. Indeed, observe that applying L is the same as blurring, down-sampling, up-sampling and blurring again, all operations are linear in the input size. Hence instead of solving the $\mathbf{x}$-update analytically, we use the conjugate-gradient method to solve

$$\left(L+\rho^k\mathbf{I}\right)\mathbf{x}^{k+1}=\frac{1}{\sigma^2}\left(\mathbf{SH}\right)^T\mathbf{y}+\rho^k\left(\mathbf{v}^k-\mathbf{u}^k\right).$$

Hence, no vectorization is needed, and this update consists of several computations, linear in the input size. All other steps, excluding the denoising step, are simple algebraic operations, linear in the input size as well. Therefore, denoising is the bottleneck in the iteration. RED (in red) and PPP (in blue) differ only in the $\mathbf{v}$-update stage. Both apply a denoiser, yet PPP applies it only once, whereas RED applies it as part of a fixed-point iteration – possibly several times. The above implies that RED is expected to be slower as $iter_{inner}$ increases.

We should note that the work reported in [61] suggests an alternative to the conjugate gradient for the $\mathbf{x}$-update stage, replacing it with a closed-form formula in the frequency domain. This alternative was used there for the SISR problem as-well. Since we only aim to extend our SISR framework from [30] to handle VSR, we remain with the conjugate gradient approach. [62] on the other hand, suggests implementing PPP using primal-dual splitting [63] instead of using ADMM. Since both PPP and RED have an ADMM implementation, we chose to use this one in order to easily compare between the two. Neither [61] nor [62] were extended to VSR yet.

Input: $\mathbf{y}$ – a LR image/video;
$D(\mathbf{x},\sigma)$ – image/video denoiser, cleaning an image/video $\mathbf{x}$ contaminated by noise with std σ;
σ – The noise level in $\mathbf{y}$
$\mathbf{S}$ – The scaling operator;
$\mathbf{H}$ – The blur operator;
β – parameter of confidence in the prior;
ρ – ADMM penalty parameter;
α – ADMM penalty parameter update factor;
$iter$ – Number of iterations.
$iter_{inner}$ – Number of iterations in fixed-point method when using RED.

Output: a SR image/video

Initialization: $\mathbf{u}^0=\mathbf{0}$;
$\rho^0=\rho$;
$\mathbf{x}^0=\mathbf{v}^0=bicubic\_interpolation(\mathbf{y})$;
$L=\frac{1}{\sigma^2}\left(\mathbf{SH}\right)^T\mathbf{SH}$.

for $k=1:1:iter$ **do**
- $\mathbf{x}^{k+1}=\left(L+\rho^k\mathbf{I}\right)^{-1}\left(\frac{1}{\sigma^2}\left(\mathbf{SH}\right)^T\mathbf{y}+\rho^k\left(\mathbf{v}^k-\mathbf{u}^k\right)\right)$

if *PPP* **then**
- $\mathbf{v}^{k+1}=D\left(\mathbf{x}^{k+1}+\mathbf{u}^k,\sqrt{\frac{\beta}{\rho^k}}\right)$

else //RED
- $z^0=\mathbf{v}^k$
- **for** $j=0:1:iter_{inner}-1$ **do**
  - $z^{j+1}=\frac{1}{\beta+\rho^k}\left(\beta D\left(z^j,\sqrt{\frac{\beta}{\rho^k}}\right)+\rho^k\left(\mathbf{x}^{k+1}+\mathbf{u}^k\right)\right)$
- **end**
- $\mathbf{v}^{k+1}=z^{iter_{inner}}$

end
- Estimate the dual gap by computing $\|\rho^k\left(\mathbf{v}^{k+1}-\mathbf{v}^k\right)\|_2^2$, and decrease ρ^{k+1} if this measure constantly increases. Otherwise $\rho^{k+1}=\alpha\rho^k$
- $\mathbf{u}^{k+1}=\frac{\rho^k}{\rho^{k+1}}\left(\mathbf{u}^k+\mathbf{x}^{k+1}-\mathbf{v}^{k+1}\right)$

end

return $\mathbf{v}^{k+1}$

Algorithm 1: Our proposed scheme for turning an image/video denoiser into a super-resolution solver using PPP/RED.

E. Convergence

It is shown in [37] that ADMM is guaranteed to converge under two conditions:

- The two terms being minimized in Equation (2), that is, the log-likelihood term and the prior, are closed, proper and convex, and
- The unaugmented Lagrangian has a saddle point.

Optimality conditions are primal and dual feasibility. Primal feasibility is reached when $\mathbf{x} = \mathbf{v}$. Increasing ρ will increase the penalty for $\|\mathbf{x} - \mathbf{v}\|_2^2$ and hence will guarantee primal feasibility. On the other hand, [37] stresses that for dual feasibility, $\rho\left(\mathbf{v}^{k+1} - \mathbf{v}^k\right) \to 0$ must hold. Hence, to achieve primal feasibility ρ should increase, but in a manner that prevents the increase of $\rho\left(\mathbf{v}^{k+1} - \mathbf{v}^k\right)$, so dual feasibility may be achieved as-well. ρ's update in Algorithm 1 is meant to increase ρ as long as $\rho\left(\mathbf{v}^{k+1} - \mathbf{v}^k\right)$ keeps decreasing. Since $\mathbf{u}$ is the *scaled* Lagrange multipliers vector, a change in ρ demands a rescale of $\mathbf{u}$. Hence, Equation (4c) is rescaled by the factor $\frac{\rho^k}{\rho^{k+1}}$ in the algorithm.

PPP's convergence was discussed in [28], [64], yet, since the prior is implicit in this scheme, convergence is not guaranteed in general. RED, on the other hand, is known to converge under mild conditions (see [29] for more details). Indeed our experimental results in section IV show this very well.

IV. Experimental Results

In this section we present various experimental results[5] that demonstrate the effectiveness of our scheme. In our previous work [30], we tested the PPP scheme on the SISR problem, aiming to increase the resolution of a single LR image. The SISR problem is considered simpler, and less time consuming than VSR, and hence tuning our framework is made easier in this case. We used the NCSR [2] algorithm, both as a denoiser in Algorithm 1 and as a super-resolution algorithm to compare with. The proposed approach proved successful and the experiments are detailed in [30]. The transition to VSR leads to a counter-intuitive paradigm that shows how the VSR problem can be handled *without relying on an accurate and explicit motion-estimation algorithm*. A careful test of this core idea is the focus of this work, and is therefore detailed in the remainder of this section.

We use the VBM3D algorithm [54] as a video denoiser both in PPP and RED, as it provides state-of-the-art denoising results; it is motion-estimation free, and hence very fast and efficient. By doing so we achieve a VSR algorithm which is motion-estimation free, and benefits from the efficiency of the chosen denoiser. In this section, we show that the resulting algorithm is indeed more efficient than existing VSR algorithms, without compromising quality of the final result. We use the same tuned parameters as in the SISR case [30] but with a small increase in the number of iterations:

$$\rho^0 = 0.0001, \quad \beta = 0.2048, \quad \alpha = 1.2, \quad iter = 40,$$

where ρ^0 is the initial penalty parameter, β is the confidence in the prior, α is the penalty parameter step, meaning each iteration ρ will be multiplied by α (unless the dual gap increases) and $iter$ is the number of iterations. For the conjugate gradient method, which is used for the $\mathbf{x}$-update, we set the tolerance to $1e-6$ with a maximum of 30 iterations. We use the last value of $\mathbf{x}$ as the initial guess. For RED, the number of fixed-point iterations, $iter_{inner}$, should be set as well. RED-1 represents a setting where $iter_{inner} = 1$ and similarly, for RED-2 $iter_{inner} = 2$. We compare our framework on several scenarios:

1) Single frame super resolution from multiple frames of global translations.
2) Single frame super resolution from real videos.
3) Super-resolved video from real videos.

The scenarios and their corresponding experiments are depicted in the following subsections. Table I details the resolution of all the data sets used in the following sections.

Video / Image	Resolution
TIP04-lines	66×72
TIP04-smiley	72×66
TIP04-text	360×168
TIP04-hemingway	524×344
Calendar	$720 \times 576 \times 31$
City	$720 \times 480 \times 31$
Penguin	$1200 \times 800 \times 31$
Temple	$1200 \times 800 \times 31$
Walk	$720 \times 480 \times 31$
Coastguard	$168 \times 144 \times 30$
Bicycle	$720 \times 576 \times 30$
Foreman	$348 \times 288 \times 30$
MissAmerica	$360 \times 288 \times 30$
Salesman	$348 \times 288 \times 30$
Tennis	$348 \times 240 \times 30$

TABLE I: Resolution of images and videos used in the presented experiments

A. Single frame super resolution from multiple frames of global translations

In this test our input is a group of multiple frames, which are all a global translation of the first frame. The goal of this synthetic experiment is to validate that the proposed scheme leads to a truly super-resolved outcome in a controlled case, and verify that it extracts most of the aliasing for producing this result. We should note that multiple frames super resolution (not necessarily of global translations) is also studied in [14], [65]–[69].

The translation is chosen randomly for each frame, up to 5 pixels in each axis. We generate 30 frames, blurred with a Gaussian kernel of s.t.d. 1 and size 3×3, then down-sampled by factor 2 and contaminated with a white Gaussian noise of s.t.d. $\sqrt{2}$. On these LR frames we run the shift-and-add algorithm reported in [14] (referred to hereafter as TIP04), which suggests a fast and robust algorithm for recovering a high-resolution image from the group of LR global translationed, blurred and noisy versions of it. TIP04 minimizes an L_1 energy term and uses a Biliteral-TV regularization. A fast implementation is suggested for pure translations, and this is the one we use.

We compare the results to RED-2, by treating the whole set of frames jointly, reconstructing a whole SR video, and

[5]All tests were conducted on a computer running Windows 8.1, with an Intel Core i7-4500U CPU 1.80GHz and 8GB RAM installed.

then taking only the first from the outcome. For both competing methods we compute the Peak Signal to Noise Ratio (PSNR[6]) of the first frame (without its borders). Notice that our algorithm is unaware of the fact that the input video is just a translation of the first frame, whereas TIP04 relies on this knowledge explicitly. TIP04 shows exceptionally good results for images with large and smooth edges, yet when the details became smaller and sharper, TIP04 encounters difficulties, and RED-2 outperforms it, as can be seen in Figure 1. Figure 2 presents the results of a second and similar experiment on a real world image, down-sampled with factor 3. Observe the aliasing in the word "Adult" in the bicubic restoration (mainly in 'l' and 't'), which has no trace in super-resolved results of the two competing methods.

These two experiments we have just described are characterized by exhibiting a simple and global motion, for which classic super-resolution methods, such as TIP04, are very effective. In such scenarios, a near perfect super-resolved outcome can be expected, recovering small details immersed in strong aliasing effects. The goal in these tests was to verify that the proposed algorithms maintain this super-resolution capability. The results indicate that, indeed, our methods successfully resolve higher resolution images, being competitive with state-of-the-art methods that are explicitly designed for this regime. We now turn to more challenging experiments with more complex video content, for which classic methods are expected to fail.

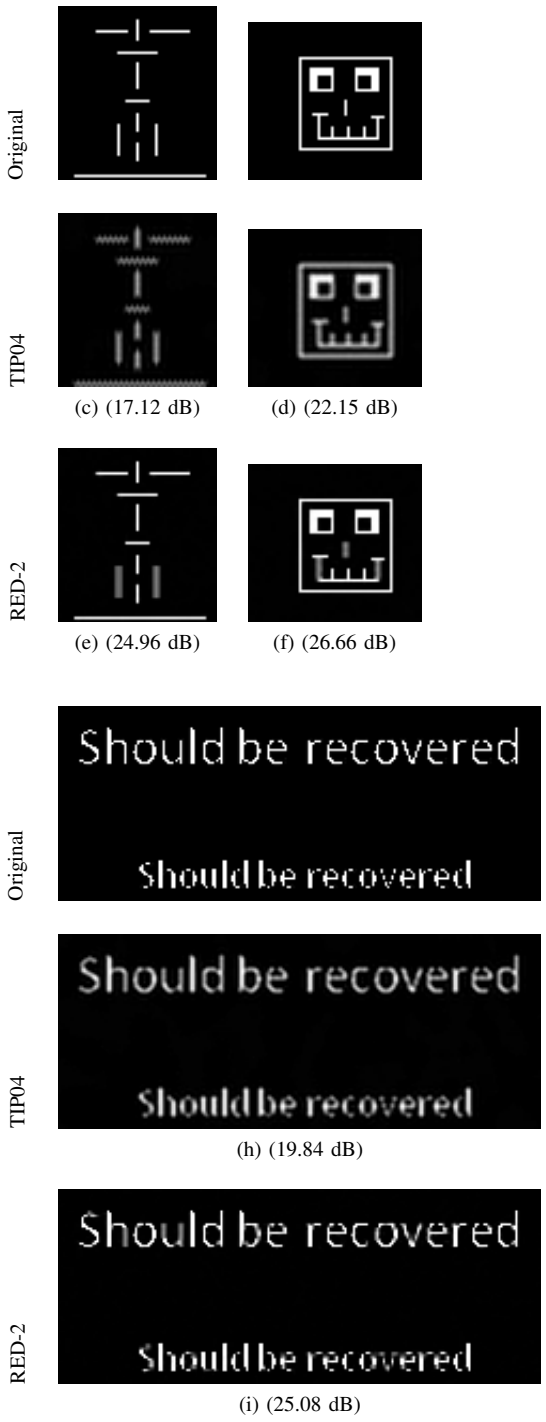
B. Single frame super resolution from real videos

The recently published DeepSR [23] is a state-of-the-art algorithm that aims to solve a slightly different problem than the classic VSR: Given the whole LR video, instead of restoring the entire sequence, DeepSR estimates only the mid-frame. It does so in two steps: First, several SR estimates for the mid-frame are generated from the LR video using different motion-estimations. The second stage is to merge HR details into a single frame by feeding all the above estimates to a trained CNN. The software package for DeepSR is available online [70], along with the dataset it was tested on. The code includes the pre-configured hyper parameters and a pre-trained CNN model. The provided package assumes a Gaussian blur of s.t.d. within the range of 1.2 to 2.4. The LR videos are created by (i) blurring each HR frame with a 7×7 Gaussian kernel of s.t.d. 1.5, followed by (ii) down scaling by a factor 4 in each axis, and (iii) adding a Gaussian noise with $\sigma = 1$ to the outcome.

We also find it interesting to compare our algorithm also to a trivial extension of SISR to VSR, hence we tested IRCNN [12], which is a state-of-the-art SISR method, on the same data set (applying it frame by frame). IRCNN uses a trained CNN that learned denoising priors to form a new denoiser that aids in solving inverse problems in a manner similar to PPP or RED.

Table II compares our PPP and RED schemes to the bicubic interpolation and IRCNN, where the PSNR is averaged over the entire sequence of frames in each video. As can be seen, RED outperforms PPP, and both lead to better reconstructions than the bicubic and IRCNN. Table III compares the proposed algorithms to DeepSR, where we measure the PSNR on the luminance channel of the mid-frame of each video. On average, RED is leading this table, the second best approach being the PPP, and both results lead to better reconstruction than DeepSR and IRCNN. Figure 3 compares visually cropped

[6]PSNR$(X, Y) = 10 \log_{10}\left(255^2 / \frac{1}{P}\|X - Y\|_2^2\right)$, where P is the size of the image.



(c) (17.12 dB) (d) (22.15 dB)

(e) (24.96 dB) (f) (26.66 dB)

(h) (19.84 dB)

(i) (25.08 dB)

Fig. 1: Three image reconstructions from a group of images with pure translations between them. Scale=2, Noise s.t.d=$\sqrt{2}$.

(a) Original

(b) LR

(c) Bicubic (21.07 dB)

(d) TIP04 (25.03 dB)

(e) RED-2 (26.48 dB)

Fig. 2: TIP04 and RED results on a real image and a zoom-in on the word *Adult*. Scale=3, Noise s.t.d=$\sqrt{2}$.

regions that are extracted from the recovered mid-frames of the Penguin video. As can be seen, DeepSR suffers from artifacts around fast moving objects such as the Penguin's wings. Figure 4 shows the squared error for the same images.

Table IV displays the time consumption of each algorithm per video. It is important to stress that we restore the entire sequence of frames in the reported time, whereas DeepSR reconstructs only the mid-frame. One can see that the PPP scheme restores the whole video in about half of the time that it takes for DeepSR to restore a single frame. As for RED-2, it is slower, but still faster then DeepSR. On average IRCNN is slower than applying RED-2.

(a) DeepSR (31.133 dB)

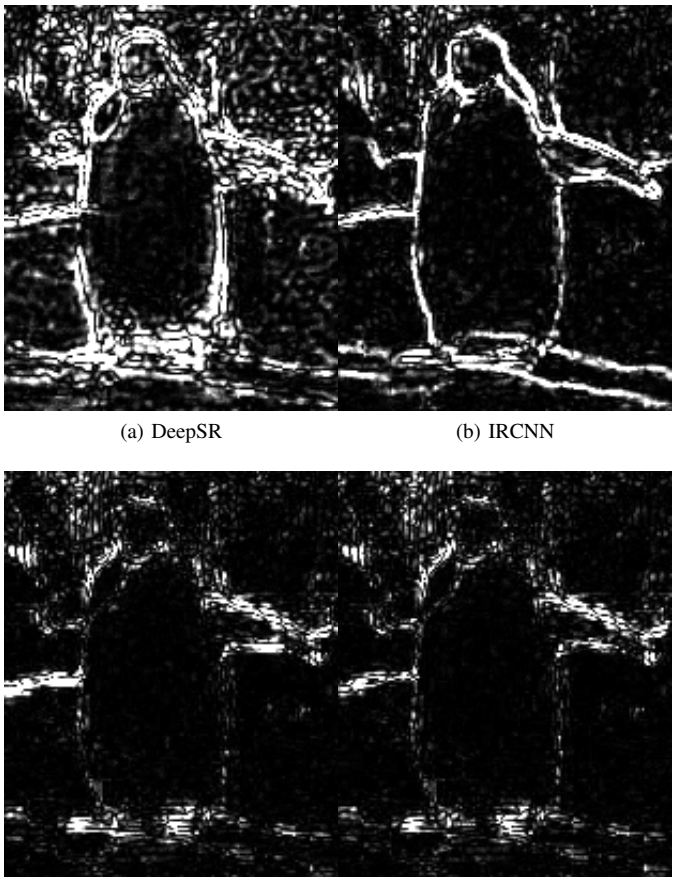
(b) IRCNN (33.0867 dB)

(c) PPP (37.137 dB)

(d) RED-2 (38.0377 dB)

(e) Original

Fig. 3: A zoomed-in area of the mid-frame of the Penguin sequence, along with the corresponding PSNR. Scale=4, Noise s.t.d=1.



(a) DeepSR (b) IRCNN

(c) PPP (d) RED-2

Fig. 4: Squared error of the areas presented in Figure 3

Video / Alg.	Bicubic	IRCNN	PPP	RED-2
PSNR				
Calendar	18.74	20.77	21.45	**21.53**
City	23.81	25.12	**26.34**	26.30
Foliage	21.54	23.58	**25.01**	24.99
Penguin	28.80	33.78	34.55	**35.54**
Temple	24.30	27.25	29.81	**29.98**
Walk	23.01	26.36	28.49	**28.57**
Average	23.37	26.14	27.60	**27.82**
SSIM				
Calendar	0.492	0.634	0.698	**0.700**
City	0.526	0.632	**0.702**	0.699
Foliage	0.451	0.614	**0.683**	0.681
Penguin	0.915	0.950	0.958	**0.963**
Temple	0.744	0.843	0.895	**0.897**
Walk	0.708	0.819	0.855	**0.857**
Average	0.639	0.749	0.798	**0.799**

TABLE II: PSNR [dB] and SSIM comparison (averaged over the entire sequence) between the bicubic interpolation, IRCNN and our algorithms. Scale=4, Noise s.t.d=1. The best results are highlighted.

Figure 5 shows a comparison between DeepSR and RED-2 on the Barcode sequence, a real low-resolution video with no ground truth. One may notice that DeepSR suffers from halos, which do not appear in RED-2's output.

Video / Alg.	IRCNN	DeepSR	PPP	RED-2
Calendar	20.76	21.53	21.53	**21.62**
City	24.54	**25.83**	25.52	25.50
Foliage	23.48	**24.95**	24.88	24.88
Penguin	33.88	32.10	34.56	**35.57**
Temple	27.49	30.60	30.77	**30.81**
Walk	26.42	26.46	28.54	**28.59**
Average	26.01	26.91	27.63	**27.83**

TABLE III: PSNR [dB] comparison between our algorithms, IRCNN and DeepSR (PSNR computed only on the midframe of our restoration). Scale=4, Noise s.t.d=1. The best results are highlighted.

Video / Alg.	DeepSR	IRCNN	PPP	RED-2
Calendar	3983	4462	**2421**	4563
City	3929	4419	**2367**	3816
Foliage	3372	3695	**1965**	3094
Penguin	11574	11113	**5321**	8360
Temple	12031	10465	**5874**	9116
Walk	3359	3687	**1951**	3049
Average	6375	6307	**3317**	5333

TABLE IV: Duration of each algorithm measured [sec]. DeepSR only reconstructs the midframe while all the others reconstruct 31 frames. The best results are highlighted.

(a) Input

(b) DeepSR

(c) RED-2

Fig. 5: Midframe of a real low resolution video constructed by DeepSR and RED-2. Scale=4.

C. Super-resolved video from real videos

3DSKR [27] is a VSR algorithm free of explicit sub-pixel motion-estimation, and thereby capable of processing videos

Video / Alg.	Bicubic	3DSKR +Deblur	PPP	RED-1	RED-2
PSNR					
Coastguard	23.77	24.75	25.16	25.16	**25.27**
Bicycle	21.62	24.32	28.79	28.80	**29.15**
Foreman	27.97	31.15	33.43	33.43	**33.53**
Salesman	24.18	25.96	26.64	26.64	**26.77**
MissAmerica	32.10	35.55	36.85	36.85	**37.20**
Tennis	21.91	22.78	23.14	23.14	**23.16**
Average	25.26	27.42	29.00	29.00	**29.18**
SSIM					
Coastguard	0.531	0.577	0.613	0.614	**0.616**
Bicycle	0.777	0.853	0.926	0.926	**0.934**
Foreman	0.823	0.874	0.900	0.900	**0.905**
Salesman	0.654	0.727	0.770	0.770	**0.774**
MissAmerica	0.877	0.912	0.910	0.910	**0.920**
Tennis	0.361	0.395	0.440	**0.441**	0.440
Average	0.671	0.723	0.760	0.760	**0.765**

TABLE V: PSNR [dB] and SSIM comparison between the bicubic, 3DSKR+deblurring, and our algorithms (PPP and RED) (RED-X is RED with X inner iterations). The PSNR is computed only on the areas restored by 3DSKR. Scale=3, Noise s.t.d=2. The best results are highlighted.

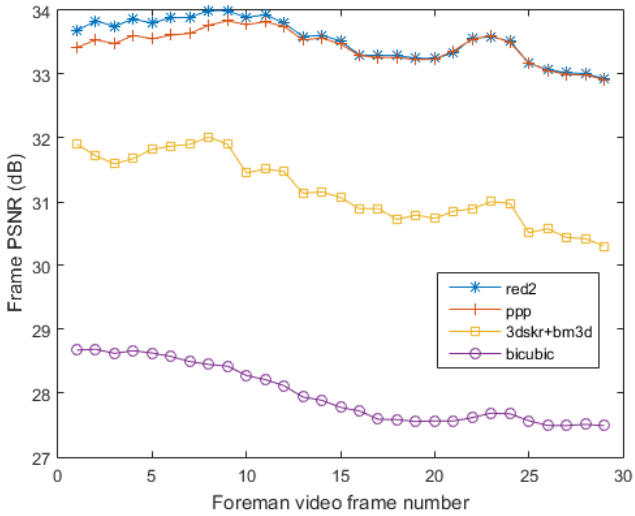


Fig. 6: PSNR [dB] per frame for the restoration of the Foreman video. Scale=3, Noise s.t.d=2.

Video / Alg.	3DSKR	3DSKR +Deblur	PPP	RED-1	RED-2
Coastgaurd	3268	3282	92	**91**	156
Bicycle	73485	73748	1835	**1780**	3170
Foreman	16590	16649	**373**	381	686
Salesman	15020	15081	**343**	351	662
MissAmerica	16463	16524	385	**377**	659
Tennis	13108	13159	**290**	300	560
Average	22989	23074	553	**547**	982

TABLE VI: Runtime [sec] of 3DSKR, 3DSKR+Deblurring, and our algorithms (RED-X is RED with X inner iterations). The best results are highlighted.

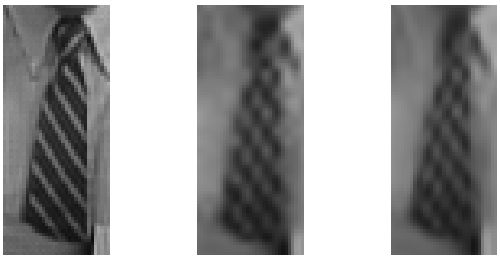
with complex motion. Specifically, the 3DSKR is composed of two-stages; The first is a super-resolution process (this step is formulated as a weighted Least-Squares problem that captures the local motion trajectories) that ignores the blurring kernel, while the second is a deblurring step (can be thought of as a post-processing operation) that takes into account the blur kernel. Using the code supplied by the authors [71], we perform a comparison of our methods and 3DSKR on a standard dataset that contains several grayscale videos: Coastguard, Bicycle, MissAmerica, Tennis and Salesman. The 3DSKR package does not include the de-blurring phase. Following previous work [22] and as suggested in the supplied code [71], we further improve the performance of this method by adding the state-of-the-art BM3D deblurring [72] algorithm as a post processing step. The parameters of the deblurring are tuned to achieve the highest PSNR. For each video, we apply the same blur kernel as in 3DSKR's demo (average blur of size 3), same decimation (factor 3 in each axis) and add the same Gaussian noise with $\sigma = 2$. Table V presents the PSNR score of each recovered video (averaged over the frames), estimated by the bicubic interpolation, 3DSKR, and the proposed algorithms. Since 3DSKR does not restore the borders of the video nor the first frame, we did not take into consideration these parts in the PSNR computation. The results in Table V suggest that RED-2 is the best performing algorithm, followed by the PPP, and these two outperform the 3DSKR and bicubic methods. Figure 6 provides a frame-by-frame PSNR analysis for the Foreman video.

Table VI depicts the runtime for the different videos and algorithms, indicating that a massive boost in runtime is also achieved. Specifically, the PPP is 30 to 50 times faster than 3DSKR (average factor of 42); RED is slightly behind, with a gain in speed-up that is approximately 23. Note that PPP is roughly 2 times faster than RED-2, as the later applies the denoiser twice in each iteration, while PPP calls the denoiser only once. This aligns with the observation that the denoising operation is the most time consuming step in our algorithms. At this point we should stress that the Fixed-Point algorithm, suggested in RED [29], might be the key to obtain a much faster process. We defer this for a future work.

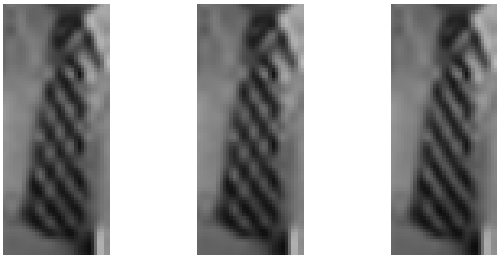
Returning to the outcome visual quality, one might falsely deduce that the small increase in PSNR between PPP and RED-2 does not justify the increase in time consumption. Yet, a closer look at the results shows that RED-2 performs much better. Figure 7 shows how RED-2 restores the pattern of the tie, while all the other algorithms confuse it with a tile-like pattern due to aliasing. Figure 8 shows that RED-2 suffers less from "pixelized" edges across the stripes.

Figure 9 displays the PSNR during the iterations of RED-2 and PPP on the Salesman sequence. The sharp drop in PSNR occurs when ρ is decreased to ensure dual feasibility. One can see that PPP converges more slowly, and is highly dependent on the ρ-update (the PSNR is flattened until ρ is decreased). RED, on the other hand, converges faster, and could have stopped the iteration earlier with almost the same PSNR score.

Our last reported experiment offers a comparison with a method called SPMC [36]. SPMC reconstructs the video frame-by-frame using a batch of frames around the

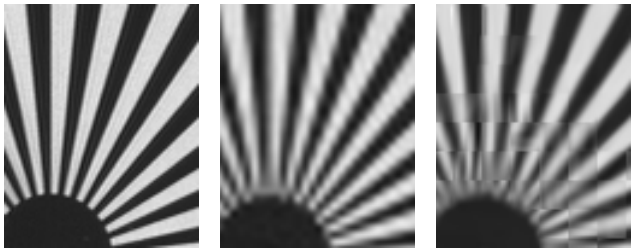


(a) Original (b) Bicubic (22.72 dB) (c) 3DSKR+BM3D (23.98 dB)

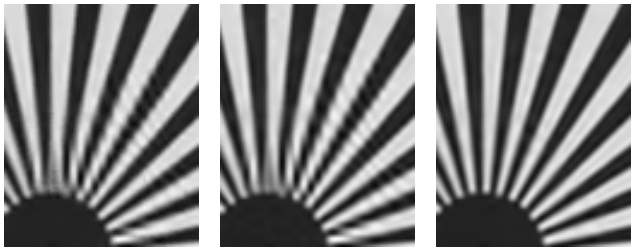


(d) PPP (25.65 dB) (e) RED-1 (25.62 dB) (f) RED-2 (26.00 dB)

Fig. 7: Zoomed in versions of the tie region (and the corresponding PSNR), extracted from Salesman. Scale=3, Noise s.t.d=2.



(a) Original (b) Bicubic (14.40 dB) (c) 3DSKR+BM3D (17.88 dB)



(d) PPP (22.74 dB) (e) RED-1 (22.69 dB) (f) RED-2 (26.22 dB)

Fig. 8: Zoomed in versions of the striped region (and the corresponding PSNR), extracted from Bicycle. Scale=3, Noise s.t.d=2.

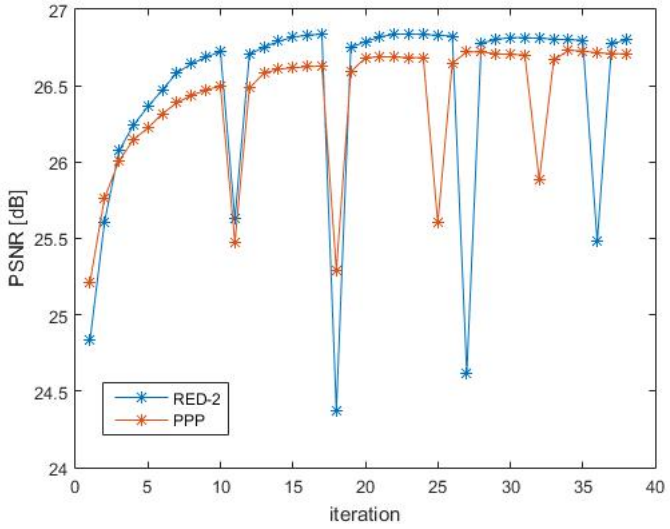
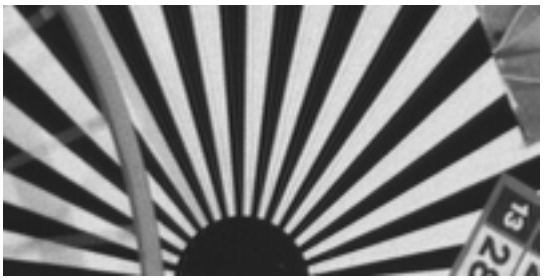


Fig. 9: PSNR during iteration of PPP and RED-2 on the Salesman sequence.

reconstructed one. It is composed of several stages: motion-estimation, followed by sub-pixel motion compensation, and finally the aligned frames are given as an input to a detail-fusion CNN. SPMC is very effective and achieves impressive and stable results. Their code is available online [73], yet supports only scale factors 2 and 4. Therefore we conducted the same test as with 3DSKR, but with a scale factor 4. Since [36] does not specify any assumption on the blur kernel, we used two degradation models: an average- and a Gaussian-blur kernel of size 7×7 and s.t.d. 1.5. The additive noise level was set to $\sigma = 1$. Table VII presents the resulting PSNR and SSIM when using average blur and Table VIII for the Gaussian one. Figure 12 provides a frame-by-frame SSIM analysis for the Foreman video with Gaussian blur. Following Tables VII and VIII, SPMC yields disappointing PSNR results, but it seems that this is mainly due to a change in grayscale level in SPMC's output as can be seen in Figures 10 and 11. SPMC's SSIM results remain high. A visual comparison shows that SPMC performs very well for the Gaussian blur, while RED-2 is roughly of the same quality. For the average blur, RED-2 seems to outperform SPMC.

V. Conclusion

In this work we have presented a simple unified scheme for integrating denoisers into both single-frame and video super-resolution, relying on the PPP and RED frameworks. The integration is done by using the denoiser as a black-box tool, applied within the iterative recovery process. The algorithm's parameters were first tuned for SISR, the easier problem, and then used for VSR without a change. We compared our proposed schemes to super-resolution algorithms for SISR, Multi-Frame Super-Resolution and VSR, achieving in all cases state-of-the-art results. More specifically, using an existing and efficient video denoiser (VBM3D [54]) we have created a robust and powerful VSR algorithm that does not depend on good locality or explicit motion-estimation.



(a) Original

(b) LR

(c) Bicubic (14.13 dB)

(d) SPMC (16.66 dB)

(e) RED-2 (21.89 dB)

Fig. 10: Zoomed in patch of the Bicycle video, frame 10 (and the corresponding PSNR) for Bicubic, SPMC and RED-2. These results refer to the average blur case with scale=4.

(a) Original

(b) LR

(c) Bicubic (28.31 dB)

(d) SPMC (28.94 dB)

(e) RED-2 (34.72 dB)

Fig. 11: Zoomed in patch of the Foreman video, frame 5, (and the corresponding PSNR) for Bicubic, SPMC and RED-2. These results refer to the Gaussian blur case with scale=4.

Video / Alg.	Bicubic	SPMC	RED-2
PSNR			
Coastguard	22.63	22.97	**24.34**
Bicycle	18.88	20.38	**24.60**
Foreman	24.42	24.11	**30.56**
Salesman	23.06	23.37	**25.79**
MissAmerica	29.53	27.16	**34.57**
Tennis	21.38	21.88	**22.76**
Average	23.31	23.31	**27.10**
SSIM			
Coastguard	0.483	0.494	**0.542**
Bicycle	0.635	0.685	**0.842**
Foreman	0.780	0.798	**0.867**
Salesman	0.572	0.675	**0.700**
MissAmerica	0.867	0.862	**0.911**
Tennis	0.319	0.335	**0.377**
Average	0.609	0.642	**0.706**

TABLE VII: PSNR [dB] and SSIM comparison between the bicubic, SPMC, and RED-2 on **average** blur and scale=4. The best results are highlighted.

Video / Alg.	Bicubic	SPMC	RED-2
PSNR			
Coastguard	22.85	23.72	**24.48**
Bicycle	19.00	23.46	**25.18**
Foreman	24.70	25.85	**30.48**
Salesman	23.19	24.25	**25.63**
MissAmerica	29.68	27.80	**34.90**
Tennis	21.38	**23.02**	22.76
Average	23.47	24.68	**27.24**
SSIM			
Coastguard	0.483	0.551	**0.560**
Bicycle	0.635	0.841	**0.868**
Foreman	0.780	0.873	**0.880**
Salesman	0.572	**0.675**	0.6700
MissAmerica	0.867	0.885	**0.914**
Tennis	0.319	**0.397**	0.388
Average	0.610	0.704	**0.713**

TABLE VIII: PSNR [dB] and SSIM comparison between the bicubic, SPMC, and RED-2 on **Gaussian** blur and scale=4. The best results are highlighted.

Future work can adopt other optimization methods, suggested in [29], such as the fixed point strategy. Another option is to use the alternative to the conjugate gradient presented in [61]. This may lead to a further improvement in time consumption. Another promising direction is modifying the proposed scheme to work on batches or even sequentially and causally rather than the entire sequence, reducing the memory requirements. Another possibility would be to further improve the results by plugging better denoising algorithms such as the VBM4D [55] or using a CNN based denoiser in a similar manner to [12], [74]. More work should be done regarding the choice of the parameters in the ADMM. Currently, the conditions to increase or decrease ρ were set empirically according to the term $\rho\left(\mathbf{v}^k - \mathbf{v}^{k+1}\right)$. Other approaches are suggested in [37], but thresholds with better mathematical reasoning are yet to be found.

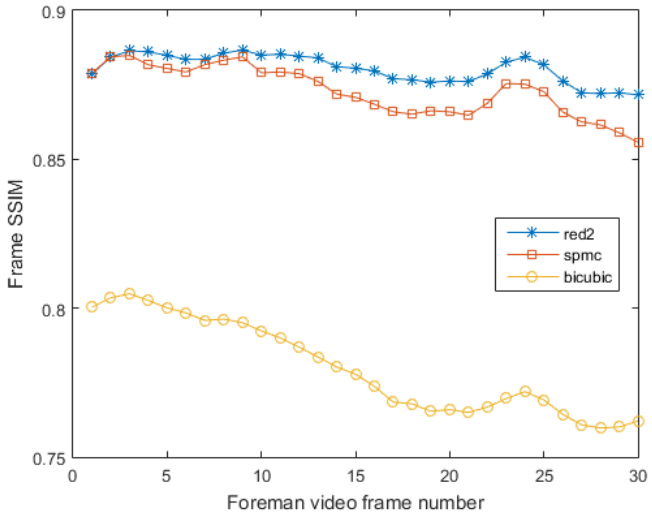


Fig. 12: SSIM per frame for the restoration of the Foreman video. Scale=3, Noise s.t.d=2, **Gaussian** blur.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council under European Unions Seventh Framework Program, ERC Grant agreement no. 320649, and from the Israel Science Foundation (ISF) grant number 1770/14. Y. Romano would like to thank the Zuckerman Institute, ISEF foundation and Viterbi fellowship from the Technion for supporting this research.

REFERENCES

[1] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, July 2011.

[2] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, April 2013.

[3] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[4] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2569–2582, June 2014.

[5] Y. Romano, M. Protter, and M. Elad, "Single image interpolation via adaptive nonlocal sparsity-based modeling," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3085–3098, 2014.

[6] A. Marquina and S. J. Osher, "Image super-resolution by tv-regularization and bregman iteration," *J. Sci. Comput.*, vol. 37, no. 3, pp. 367–382, Dec. 2008.

[7] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 349–356.

[8] H. Chen, X. He, O. Teng, and C. Ren, "Single image super resolution using local smoothness and nonlocal self-similarity priors," *Signal Processing: Image Communication*, vol. 43, pp. 68 – 81, 2016.

[9] C. Xie, W. Zeng, S. Jiang, and X. Lu, "Multiscale self-similarity and sparse representation based single image super-resolution," *Neurocomputing*, vol. 260, pp. 92 – 103, 2017.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[11] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 49–64.

[12] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2808–2817.

[13] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, Dec 1997.

[14] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct 2004.

[15] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Multiframe resolution-enhancement methods for compressed video," *IEEE Signal Processing Letters*, vol. 9, no. 6, pp. 170–174, June 2002.

[16] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.

[17] M. Ben-Ezra, A. Zomet, and S. K. Nayar, "Video super-resolution using controlled subpixel detector shifts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 977–987, 2005.

[18] M. Elad and Y. Hel-Or, "A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur," in *Electrical and electronic engineers in israel, 2000. the 21st ieee convention of the*. IEEE, 2000, pp. 402–405.

[19] M. Irani and S. Peleg, "Super resolution from image sequences," in *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 2. IEEE, 1990, pp. 115–120.

[20] A. Zomet, A. Rav-Acha, and S. Peleg, "Robust super-resolution," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.

[21] D. Mitzel, T. Pock, T. Schoenemann, and D. Cremers, "Video super resolution using duality based TV-L1 optical flow," in *Joint Pattern Recognition Symposium*. Springer, 2009, pp. 432–441.

[22] 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, Feb 2014.

[23] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 531–539, Dec 2015.

[24] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, pp. 109–122, June 2016.

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[26] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 36–51, Jan 2009.

[27] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1958–1975, Sept 2009.

[28] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, Dec 2013.

[29] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imaging Science*, vol. 10, no. 4, pp. 1804–1844, 2017.

[30] A. Brifman, Y. Romano, and M. Elad, "Turning a denoiser into a super-resolver using plug and play priors," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1404–1408.

[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, no. 2, pp. 109–122, June 2016.

[32] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 1015–1028, April 2018.

[33] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 3224–3232.

[34] J. Caballero, C. Ledig, A. Aitken, A. A., J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2848–2857, 2017.

[35] M. S. M. Sajjadi, R. V., and M. Brown, "Frame-recurrent video super-resolution," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6626–6634, 2018.

[36] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[37] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[38] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[39] M. Protter and M. Elad, "Super resolution with probabilistic motion estimation," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1899–1904, Aug 2009.

[40] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

[41] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, Dec 2006.

[42] X. Lu, H. Yuan, P. Yan, L. Li, and X. Li, "Image denoising via improved sparse coding," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 74.1–74.0.

[43] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 244–252.

[44] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D image denoising with shape-adaptive principal component analysis," in *SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations*, R. Gribonval, Ed., Apr. 2009. [Online]. Available: https://hal.inria.fr/inria-00369582

[45] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Transactions on Image Processing*, vol. 21, no. 4, p. 1635, 2012.

[46] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 755–768, Feb 2014.

[47] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: a low-rank approach," *IEEE transactions on image processing*, vol. 22, no. 2, pp. 700–711, 2013.

[48] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 2862–2869.

[49] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[50] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, Sept 2018.

[51] P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, April 2010.

[52] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2833–2840.

[53] P. Chatterjee and P. Milanfar, "Practical bounds on image denoising: From estimation to information," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1221–1233, 2011.

[54] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," *2007 15th European Signal Processing Conference*, pp. 145–149, Sept 2007.

[55] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, Sept 2012.

[56] R. Almahdi and R. C. Hardie, "Recursive non-local means filter for video denoising with poisson-gaussian noise," *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*, pp. 318–322, July 2016.

[57] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 839–842, Dec 2005.

[58] M. Protter and M. Elad, "Sparse and redundant representations and motion-estimation-free algorithm for video denoising," in *Wavelets XII*,

vol. 6701. International Society for Optics and Photonics, 2007, p. 67011D.

[59] A. Buades, B. Coll, and J. M. Morel, "Denoising image sequences does not require motion estimation," in *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, Sept 2005, pp. 70–74.

[60] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," in *IEEE Transactions on Image Processing*, vol. 18, no. 1, Jan 2009, pp. 27–35.

[61] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play admm for image restoration: Fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, March 2017.

[62] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1108–1112, Aug 2017.

[63] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011. [Online]. Available: http://dx.doi.org/10.1007/s10851-010-0251-1

[64] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, Dec 2016.

[65] P. Milanfar, S. Farsiu, M. Elad, and M. D. Robinson, "Robust reconstruction of high resolution grayscale images from a sequence of low resolution frames," Jan. 13 2009, US Patent 7,477,802.

[66] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning, "A multi-frame image super-resolution method," *Signal Processing*, vol. 90, no. 2, pp. 405–414, 2010.

[67] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, Dec 1997.

[68] S. Lertrattanapanich and N. K. Bose, "High resolution image formation from low resolution frames using delaunay triangulation," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1427–1441, Dec 2002.

[69] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust super-resolution," in *2003 International Conference on Image Processing (ICIP)*, Sep. 2003, pp. II–291.

[70] DeepSR website. [Online]. Available: http://www.cse.cuhk.edu.hk/leojia/projects/DeepSR/

[71] 3DSKR website. [Online]. Available: http://alumni.soe.ucsc.edu/~htakeda/SpaceTimeSKR.htm

[72] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3d transform-domain collaborative filtering," *Proc.SPIE*, vol. 6812, pp. 6812 – 6812 – 12, 2008.

[73] SPMC github. [Online]. Available: https://github.com/jiangsutx/SPMC_VideoSR

[74] W. Dong, P. Wang, W. Yin, and G. Shi, "Denoising prior driven deep neural network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.