SSIM Compliant Modeling Framework With Denoising and Deblurring Applications

Rajesh Bhatt, Naren Naik, and Venkatesh K. Subramanian

Abstract— **In image processing, it is well known that mean square error criteria is perceptually inadequate. Consequently, image quality assessment (IQA) has emerged as a new branch to overcome this issue, and this has led to the discovery of one of the most popular perceptual measures, namely, the structural similarity index (SSIM). This measure is mathematically simple, yet powerful enough to express the quality of an image. Therefore, it is natural to deploy SSIM in model based applications, such as denoising, restoration, classification, etc. However, the non-convex nature of this measure makes this task difficult. Our attempt in this work is to discuss problems associated with its convex program and take remedial action in the process of obtaining a generalized convex framework. The obtained framework has been seen as a component of an alternative learning scheme for the case of a regularized linear model. Subsequently, we develop a relevant dictionary learning module as a part of alternative learning. This alternative learning scheme with sparsity prior is finally used in denoising and deblurring applications. To further boost the performance, an iterative scheme is developed based on the statistical nature of added noise. Experiments on image denoising and deblurring validate the effectiveness of the proposed scheme. Furthermore, it has been shown that the proposed framework achieves highly competitive performance with respect to other schemes in literature and performs better in natural images in terms of SSIM and visual inspection.**

Index Terms— **SSIM, sparse representation, denoising, deblurring.**

I. Introduction

IN IMAGE science, a perceptually adequate measure is an old and fundamental quest. Such a measure would play a vital role in numerous image processing algorithms and applications such as error criterion based algorithmic design such as denoising, restoration, classification, super-resolution and much more. However, deriving such a measure is a non-trivial task [1]. A popular alternative to a perceptual measure is the mean square error (MSE). The reasons for its popularity is efficient computability for optimization tasks, the amenability to analysis and optimality in estimation of Gaussian distributed data. However, it does not consider the true structure of the underlying signal. Bernd [2] had shown its inefficiency to produce high quality images with an algorithm based on optimal MSE. Furthermore, Wang and Bovik [1], [3] elaborately discuss pitfalls of MSE in terms of image quality assessment (IQA). In particular, a good contrast image may not meet with the original image (poor fidelity) but may be perceptually adequate (of good quality). Therefore, it is natural to design algorithms for different image processing tasks based on the image quality metric (IQM).

We start our journey from IQA, which provides a convenient platform over which ideas of good IQM were built. Indeed, numerous contributions in the past two decades addressed issue of providing good IQM measure [1], [4]–[7]. However, in the context of using IQM in image processing algorithms, desirable features of this metrics were its visual agreement with human visual system (HVS), computational plausibility and simplified mathematical definition so that it could be used in various algorithms. In the literature, various IQA techniques have been proposed, including full-reference (where IQA is aimed to design assessment for test image in the presence of a reference) [8]–[11], no-reference (where quality assessment needs no reference image) [6], [12], [13] and reduced-reference (where reference would be known partially) [6], [14], [15] based schemes. For our purpose, we focus on full-reference IQA (FR-IQA). In FR-IQA, the structural similarity index (SSIM) proposed by Wang et. al. [8] is one of the most aligned IQM with the above mentioned desirable properties [3].

The popularity of SSIM is unparalleled, it is even incorporated into many practical hardware and software systems [4]. The SSIM index is an objective top-down approach. It is a manifestation of usage of structural information by human visual system (HVS) for comparison purposes and therefore, it takes structural correlation as its main ingredient for comparison between reference and test image. It locally compares luminance and contrast along with structural content of reference and test image. Finally, it pools up all this to give overall IQM. It is experimentally found that quality assessment based on SSIM index correlates better to mean opinion score (MOS) obtained by subjective evaluation for large image database. For details, interested readers may refer to Wang *et. al.* [8], Wang and Bovik [1]. Many researchers have further tried to explore other applications of the SSIM index. For instance, Mahmud *et. al.* [16] modified Wiener filtering process in BM3D denoising scheme [17] based on SSIM index. Their scheme performs better than BM3D, however applied modification is confined only to BM3D scheme. Rehman *et al.* [18] have tuned sparse modeling algorithms based on SSIM for

Manuscript received October 7, 2019; revised June 24, 2020 and October 27, 2020; accepted January 6, 2021. Date of publication January 27, 2021; date of current version February 5, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Damon M. Chandler. *(Corresponding author: Rajesh Bhatt.)*

The authors are with the Electrical Engineering Department, IIT Kanpur, Kanpur 208016, India (e-mail: bhatt@iitk.ac.in; nnaik@iitk.ac.in; venkats@iitk.ac.in).

Digital Object Identifier 10.1109/TIP.2021.3053369

denoising and super-resolution applications. Their observation revealed that the generated SSIM quality map of textured regions appears to be much less noisy than smooth regions. Unfortunately, their proposed modification works only for ℓ_0-based sparse representation. Another interesting observation from their work [19] based on non-local mean (NLM), when all weights are calculated separately using the ℓ_2 distance or the SSIM index from original image, is that NLM based denoising is better for latter weights. It is worth while to mention that SSIM index has shown sensitivity towards non-structural distortion like rotation, translation scaling and other misalignments. Sampat et. al. [20] were motivated by said sensitivities and introduce complex wavelet structural similarity index (CW-SSIM). The CW-SSIM index was further used in classification task [21]. For real world data, perceptual adequacy of SSIM index facilitated many image based applications range from compression [22], target recognition [23], classification [21], fusion [24], restoration [18], [25], camera design [26] and many more. Some interesting attempts have been made to kick-start the use of SSIM index directly in optimization driven applications. For example, Channappayya and Bovik [27] designed a linear equalizer by transforming problem into quasi-convex problem so that SSIM index could be optimized. Furthermore, Otero and Vrscay elegantly solved optimization of SSIM measure with ℓ_1-norm regularizer by deploying a related gradient scheme [28]. In another attempt, they derived a sequence of approximated smooth sub-problems using mollifiers [29]. Recently Otero *et. al.* [30] deployed Alternate Direction Method of Multipliers (ADMM) for solving an unconstrained SSIM-based optimization problem. Wang *et. al.* [31] came up with a new perspective and equipped NLH1 and NLTV regularization functionals with structural similarity information. Despite of mentioned attempts, there still exist many interesting questions, for example, can we formulate and solve optimization problem related to SSIM in a generalized setting? Can there be any learning or other boosting methods derived to explore full potential of SSIM index? Can a generalized solver be tuned to SSIM based optimization? The work reported here is an attempt to answer these questions. A simplified (pseudo SSIM) function (named as **pS**) is developed from SSIM, which aims to take the benefits of SSIM and be easy to fit with optimization framework. Additionally, components of alternating learning [32], [33] based on **pS**-function are developed. Organization of the paper will be explored in the coming section.

A. Paper Organization

Section II opens with a brief recap of the SSIM perceptual measure. Afterward, optimization plausible form of the SSIM, the pssim-function and its features are investigated. Section III discusses alternative learning and its components. In the next step, our aim is to develop the pssim-function based alternative learning components. For this, a convex program based on **pS**-function discrepancy for a linear model with an arbitrary choice of regularizer or constraint is derived. Further, we derive the corresponding dictionary learning scheme. In Section V, a stepwise clustering (used in later sections) is explored. Subsequently, in Section VI, denoising and deblurring problem with sparse regularized linear model under the roof of our pssim-function is investigated. For further calibration of the proposed scheme, an iterative scheme for smoothness correction is introduced using the statistical nature of the added noise. Section VII presents extensive experimental results and their comparison with other methods. The simulation results for denoising and deblurring applications show the effectiveness of the proposed scheme especially with respect to the SSIM index.

II. Problem Formulation

A. SSIM Index

As mentioned earlier, SSIM [8] index compares the reference $\mathbf{I}_R$ and test $\mathbf{I}_T$ images of size (say $N \times N$) in terms of localized luminance $l(\mathbf{z}, \mathbf{y})$, contrast $c(\mathbf{z}, \mathbf{y})$ and structural similarity $s(\mathbf{z}, \mathbf{y})$ information, where $\mathbf{z}$ and $\mathbf{y}$ are column major ordering of the square patch obtained from $\mathbf{I}_R$ and $\mathbf{I}_T$, respectively. Suppose, size of these patch vectors is m such that $m \ll N$. It is reasonable to compute luminance, contrast and structural information locally because images are considered to be spatially non-stationary. Additionally, luminance component uses mean (first order statistics) while contrast and structural components use second order statistics in their expressions. Finally, these three localized attributes are combined to produce local SSIM $S(\mathbf{z}, \mathbf{y})$,

$$\begin{aligned} S(\mathbf{z}, \mathbf{y}) &= l(\mathbf{z}, \mathbf{y})c(\mathbf{z}, \mathbf{y})s(\mathbf{z}, \mathbf{y}) \\ &= S_1(\mu_{\mathbf{z}}, \mu_{\mathbf{y}})S_2(\sigma_{\mathbf{z}}, \sigma_{\mathbf{y}}) \\ &= \frac{2\mu_{\mathbf{z}}\mu_{\mathbf{y}} + C_1}{\mu_{\mathbf{z}}^2 + \mu_{\mathbf{y}}^2 + C_1} * \frac{2\sigma_{\mathbf{zy}} + C_2}{\sigma_{\mathbf{z}}^2 + \sigma_{\mathbf{y}}^2 + C_2} \end{aligned} \tag{1}$$

where, $(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}})$ and $(\mu_{\mathbf{y}}, \sigma_{\mathbf{y}})$ are mean and standard deviation pairs for reference patch vector $\mathbf{z}$ and test patch vector $\mathbf{y}$, respectively and C_1, C_2 are stability constants. To obtain the final scalar score, locally assessed SSIM were pooled-up using averaging (also called as mean SSIM or MSSIM). Recently, Brunet *et. al.* [34] addressed mathematical nature of SSIM index and concluded its $S_1(\mu_{\mathbf{z}}, \mu_{\mathbf{y}})$ and $S_2(\sigma_{\mathbf{z}}, \sigma_{\mathbf{y}})$ components are non-convex but valid metrics, since translation invariance does not hold [35].

B. Simplification for Optimization

Our first wish is to make local SSIM in Eq. (1) compatible for optimization. Notice that local SSIM assessment $S(\mathbf{z}, \mathbf{y})$ ranges in between $[-1, 1]$ and equals to 1 if $\mathbf{z} = \mathbf{y}$. If we further assume reference and test patches as zero mean vectors then $S_1(\mu_{\mathbf{z}}, \mu_{\mathbf{y}})$ in Eq. (1) goes to 1 and vanishes. This assumption makes sense since the reference patch is inaccessible in model fitting and we directly estimate mean component of reference patch by the observed test patch. Many authors have adopted the said assumption in their model fitting framework [18], [27]–[29]. Finally, local assessing pair $S(\mathbf{z}, \mathbf{y})$

reduces to

$$S(\mathbf{z}, \mathbf{y}) = S_2(\sigma_{\mathbf{z}}, \sigma_{\mathbf{y}}) = \frac{2\sigma_{\mathbf{zy}} + C_2}{\sigma_{\mathbf{z}}^2 + \sigma_{\mathbf{y}}^2 + C_2} \tag{2}$$

$$T(\mathbf{z}, \mathbf{y}) = 1 - S(\mathbf{z}, \mathbf{y}) = \frac{\sigma_{\mathbf{z}}^2 + \sigma_{\mathbf{y}}^2 - 2\sigma_{\mathbf{zy}}}{\sigma_{\mathbf{z}}^2 + \sigma_{\mathbf{y}}^2 + C_2} \tag{3}$$

The range of T in Eq. (3) would be $[0, 2)$. As in [34], replacing second order statistical terms with their estimates and for $(m-1)C_2 = C$, Eq. (3) reduces to

$$T(\mathbf{z}, \mathbf{y}) = \frac{\|\mathbf{z} - \mathbf{y}\|^2}{\|\mathbf{z}\|^2 + \|\mathbf{y}\|^2 + C} \tag{4}$$

where we assume reference and test vectors are $\mathbf{z}$ and $\mathbf{y}$, respectively. Notice that when $T(\mathbf{z}, \mathbf{y})$ approaches zero, then reference and test vectors approach each other. The purpose of the C is to provide stability, we choose to use it latter and continue the analysis without it. Subsequently, Eq. (4) reduces to

$$T(\mathbf{z}; \mathbf{y}) = \frac{\|\mathbf{z} - \mathbf{y}\|^2}{\|\mathbf{z}\|^2 + \|\mathbf{y}\|^2} \tag{5}$$

If we restrict the range of $T(\mathbf{z}; \mathbf{y})$ in between interval $[0, 1]$, then this conditioned function for test vector $\mathbf{y}$ forms a convex set in variable $\mathbf{z}$. Therefore, set $\mathcal{Z} = \{\mathbf{z} \mid T(\mathbf{z}, \mathbf{y}) \in [0, 1]\}$ is a convex set. Notice that the mentioned restriction simply emphasizes further closeness in context of loss function. The following lemma encodes the above mentioned restriction in terms of a simple condition.

Lemma 1: For given $\mathbf{y} \in \mathcal{R}^m$, set $\mathcal{Z} = \{\mathbf{z} \mid T(\mathbf{z}, \mathbf{y}) \in [0, 1]\}$ and $\mathcal{Z}' = \{\mathbf{z} \mid \langle \mathbf{z}, \mathbf{y} \rangle > 0\}$ are equal.

Proof: Suppose, an arbitrary $\mathbf{z}$ obeys $\langle \mathbf{z}, \mathbf{y} \rangle > 0$, then also $-2\langle \mathbf{z}, \mathbf{y} \rangle < 0$, adding both side by $\|\mathbf{z}\|^2 + \|\mathbf{y}\|^2$ leads to $\|\mathbf{z} - \mathbf{y}\|^2 < \|\mathbf{z}\|^2 + \|\mathbf{y}\|^2$ and by Eq. (5), $T(\mathbf{z}, \mathbf{y}) < 1$ and by definition $T(\mathbf{z}, \mathbf{y}) \geq 0$. On the other hand, suppose $T(\mathbf{z}, \mathbf{y}) \in [0, 1]$ and Eq. (5) leads to $\|\mathbf{z} - \mathbf{y}\|^2 < \|\mathbf{z}\|^2 + \|\mathbf{y}\|^2$; therefore, $\langle \mathbf{z}, \mathbf{y} \rangle > 0$. This concludes the proof. □

Next, suppose we define function $f(\mathbf{z}; \mathbf{y})$ in variable $\mathbf{z}$ for given test patch $\mathbf{y}$ using $T(\mathbf{z}; \mathbf{y})$ as follows

$$f(\mathbf{z}; \mathbf{y}) = \frac{2T(\mathbf{z}; \mathbf{y})}{1 - T(\mathbf{z}; \mathbf{y})} = \frac{\|\mathbf{z} - \mathbf{y}\|^2}{\langle \mathbf{y}, \mathbf{z} \rangle} \tag{6}$$

In general, using $f(\mathbf{z}; \mathbf{y})$ instead of $T(\mathbf{z}; \mathbf{y})$ in minimization task is hopeless. For example, consider the case of regularized linear model for $p \geq 1$.

$$\underset{\mathbf{x}}{\text{minimize}} \;\; f(\mathbf{Dx}; \mathbf{y}) + \lambda \, \|\mathbf{x}\|_p$$

Above minimization is unbounded below. Suppose we include condition $\mathbf{z} \in \mathcal{Z}'$ in the definition of $f(\mathbf{z}; \mathbf{y})$. Such a stipulation eliminates unboundedness while using $f(\mathbf{z}; \mathbf{y})$ in the minimization procedure as well as makes $f(\mathbf{z}; \mathbf{y})$ non-negative. Additionally, it is easy to verify that both $f(\mathbf{z}; \mathbf{y})$ and $T(\mathbf{z}; \mathbf{y})$ are equivalent in terms of the optimal point for condition $\mathbf{z} \in \mathcal{Z}'$. Furthermore, by lemma 1, the condition $\mathbf{z} \in \mathcal{Z}'$ simply implies $T(\mathbf{z}; \mathbf{y}) \in [0, 1]$. We now add a positive constant c in the denominator for stability. We can also use c for tuning the algorithm for given task. With this, function $f(\mathbf{z}; \mathbf{y})$ is modified such that $\mathcal{Z}'$ equals to $\{\mathbf{z} \mid \langle \mathbf{z}, \mathbf{y} \rangle + c > 0\}$. Let us call this special function as a pseudo-SSIM function and represent it by **pS**

$$\mathbf{pS}(\mathbf{z}; \mathbf{y}) = \frac{p_S(\mathbf{z}; \mathbf{y})}{q_S(\mathbf{z}; \mathbf{y})} = \frac{\|\mathbf{z} - \mathbf{y}\|^2}{\langle \mathbf{z}, \mathbf{y} \rangle + c} \quad for \;\; \mathbf{z} \in \mathcal{Z}' \tag{7}$$

In summary, strictly speaking due to the restricted domain, it is not appropriate to use the **pS** function to measure similarity between two different arbitrary images. Our purpose in the development of the **pS** function is to efficiently embed SSIM nature in model fitting with the objective of yielding numerically stable schemes. For this, a solution in the suitably constrained space is searched. Additionally, it requires the provided test vector $\mathbf{y}$ to be centralized. It is interesting to note that **pS** function is a ratio of MSE to correlation term between test and unknown patch vector. In other words, **pS** function further enforces affinity in correlation between the unknown vector and the measured test vector. In the minimization procedure, this roughly implies that for a given MSE sphere of radius ϵ, centered at $\mathbf{y}$, the **pS** function emphasizes strong candidacy for a higher positively correlated vector $\mathbf{z}$.

III. From Modeling to Alternate Learning

Applications driven by the modeling framework require a learning procedure to further boost the performance [32], [36], [37]. Learning algorithms essentially determine the optimal setting of model parameters. For this reason, an alternate learning procedure is adopted by many authors [18], [33], [36], [38]–[44]. An alternate learning procedure is essentially a stepwise optimization procedure to capture the best hypothesis. We briefly describe this learning procedure in a general setting

Consider, $\mathbf{x}$ and α are latent (or representation) variable and model parameter respectively with a chosen model $g(\mathbf{x}; \alpha)$. Suppose the set Γ denotes restrictions on the latent variable and model setting, representing plausible knowledge of the problem. We are interested in solving the following problem,

$$\begin{aligned} &\underset{\mathbf{x}_r, \alpha}{\text{minimize}} \; \sum_{r=1}^{N} \mathsf{d}(\mathbf{y}_r, g(\mathbf{x}_r; \alpha)) \\ &\text{subject to } (\mathbf{x}_r, \alpha) \in \Gamma \; ; \; r = 1 \cdots N \end{aligned} \tag{8}$$

where d is a suitable discrepancy measure and $\mathbf{y}_r$ is the r^{th} observation from the training matrix $\mathbf{Y} = \left[\mathbf{y}_1 \cdots \mathbf{y}_r \cdots \mathbf{y}_N\right]$. It is worth mentioning that the restrictions (Γ) further translates into the set of regularizers or/and constraints according to the targeted application. The optimization problem (8) for variables α and $\mathbf{x}_r$ such that $r = 1 \cdots N$ is largely insolvable simultaneously. Oftentimes, either variable coupling raises a non-convex problem or simply, the problem is non-convex in one variable and admits multiple global optima, for instance [18], [36] [33], [38]. Consequently, one can only hope for a sub-optimal solution [32]. This gives rise to an alternating procedure to solve the optimization problem (8) and therefore solves the problem for one variable at a time for a given training set $\mathbf{Y}$. The Algorithm 1 provides further details of the process. Our attempted in this work, is to replace the fidelity measure d in Eq. (8) by the **pS** function for a general setting. For this, the corresponding solving strategies for the alternate

Algorithm 1 Pseudo Code for Alternate Learning

```
Input: Training Matrix: $\mathbf{Y}$, max Iterate = $K$, error tolerance = $\Delta$
Initialize: Learning Parameter: $\alpha := \alpha^0$, Iter: $k = 0$
repeat
    • Latent Coding: For fixed learning parameter set $\alpha$, solve Eq. (8) for latent code matrix $\mathbf{X}^k$.
    • Parameter Learning: For fixed latent code matrix $\mathbf{X}^k$, solve optimization Eq. (8) for learning parameters in set $\alpha^k$.
    • Compute error: $\delta_k = e(\mathbf{Y}, \mathbf{X}^k, \alpha^k)$
    • Update Iterate: $k := k + 1$
until $\delta_k \leq \Delta$ OR $k = K$
```

learning components namely, ***latent coding*** and ***parameter learning*** will be devised in the upcoming section.

IV. Convex Reformulation

In this section, our desire is to achieve the convex formulation/program for **pS** function in order to efficiently solve the latent coding stage in Algorithm 1. We will discuss two formulations to do so. One approach is specialized for the constrained case and the other approach leads to second order cone programing (SOCP) [45]. This would help any user to choose formulation according to their specification. The importance of regularized linear models is well known: they are very efficient both in computation and performance [46]–[48]. So, we restrict ourselves to linear regularized or constrained models for the rest of the paper. Mathematically, a regularized linear model is expressed as $f(\mathbf{x}; \alpha) = \mathbf{Dx}$ with regularizer or constraint on the latent variable $\mathbf{x}$, where $\mathbf{D}$ is a $\mathbb{R}^{n\times m}$ matrix. We further assume that the regularizer/constraint are convex. Theorem 1 translates the **pS** function based objective function under linear regularized model into the convex program. Note that the choice of convex regularizer/constraint is general, therefore the following convex program can be applicable for broader range of applications. In Section IV-B, an efficient solver is discussed to solve this convex program.

Theorem 1: Suppose model $f(\mathbf{x}; \alpha)$ *is* $\mathbf{Dx}$ *and the set* $\{\boldsymbol{\Omega}_i \mid \boldsymbol{\Omega}_i : \mathbb{R}^m \to \mathbb{R}_+, i = 1\cdots r\}$ *and* $\{\Gamma_i \mid \Gamma_i : \mathbb{R}^m \to \mathbb{R}, i = 1\cdots k\}$ *are convex regularizer and constraint set, respectively. If compact notation of* $\mathbf{D^T y}$ *is* $\mathbf{d}_y$ *then the following holds*

$$\begin{aligned}&\text{minimize } \mathbf{pS}(\mathbf{y}, \mathbf{Dx}) + \textstyle\sum_{i=1}^{r} \lambda_i \boldsymbol{\Omega}_i(\mathbf{x})\\ &\text{subject to } \mathbf{x} \in \mathbf{dom}(\mathbf{pS})\end{aligned}$$

$$\equiv \begin{aligned}&\text{minimize } p + q + \textstyle\sum_{i=1}^{r} \lambda_i \boldsymbol{\Omega}_i(\mathbf{x})\\ &\text{subject to } \mathbf{d}_y^T\mathbf{x} + c \geq 0,\ \ q - p \geq 0\\ &\qquad\qquad \left\| \begin{matrix}\mathbf{y} - \mathbf{Dx}\\ q\end{matrix} \right\| \leq p\end{aligned} \tag{9}$$

$$\begin{aligned}&\text{minimize } \mathbf{pS}(\mathbf{y}, \mathbf{Dx})\\ &\text{subject to } \mathbf{x} \in \mathbf{dom}(\mathbf{pS})\\ &\qquad\qquad \Gamma_i(\mathbf{x}) \leq L_i\\ &\qquad\qquad \textit{for } i = 1\cdots k\end{aligned} \equiv \begin{aligned}&\text{minimize } \frac{1}{w}\|w\mathbf{y} - \mathbf{Dz}\|^2\\ &\text{subject to } \Gamma_i(\mathbf{z}) \leq 0\\ &\qquad\qquad \textit{for } i = 1\cdots k\\ &\qquad\qquad w \geq 0\\ &\qquad\qquad \mathbf{d}_y^T\mathbf{z} + wc = 1\end{aligned} \tag{10}$$

Proof: See appendix A □

Algorithm 2 Dinkelbach's Algorithm [51]

```
1:  Aim: To minimize $P(\mathbf{d})/Q(\mathbf{d})$ for convex $P(\mathbf{d})$ and concave $Q(\mathbf{d})$ for some given constraint $\mathbf{d} \in \Psi$.
2:  Initialize: Max Iterate = $K$, error = $\Delta$
3:  Compute: $t_i^{(0)} = P_i(\mathbf{d}^{(0)})/Q_i(\mathbf{d}^{(0)})$ for $i = 1 \ldots N$
4:  For $k = 1$ to $K$ do
5:      • Solve:
          $\underset{\mathbf{d}}{\text{minimize}} \quad F_k \triangleq \sum_{i=1}^{N} P_i(\mathbf{d}^{(k)}) - t_i^{(k-1)} Q_i(\mathbf{d}^{(k)})$
          $\text{subject to} \quad \mathbf{d}^{(k)} \in \Psi$
6:      • Evaluate: $\delta = |F_k - F_{k-1}|$
7:      if $\delta \leq \Delta$ then
8:          break
9:      End if
10:     • Evaluate: $t_i^{(k)} := P_i(\mathbf{d}^{(k)})/Q_i(\mathbf{d}^{(k)})$ for $i = 1 \cdots N$
11: End For
```

A. Matrix/Dictionary Learning

Olshausen and Field [49], [50] first introduced dictionary learning procedure to understand HVS. Later, such MSE based matrix learning found numerous applications in image reconstruction and classification problems [33], [37], [38]. In this section, our focus is to develop the algorithm for the parameter (dictionary) learning stage in the Algorithm 1 for the **pS** function. For the sake of simplicity, we do not include any regularizer on dictionary. However, the applied procedure can easily be extended for such cases. Notice that equivalent convex forms (9) and (10) of the original problem are derived to solve latent variable **x** for fixed dictionary **D**. So, we directly optimize the original objective function for dictionary learning.

We further constrain atoms of $\mathbf{D} = [\mathbf{d}_1 \cdots \mathbf{d}_m]$ such that each atom has unity norm. This not only makes the optimization problem stable, but also reflects that we are interested only in direction and not in the magnitude. Further, dictionary learning is performed atom-wise separately. This sub-optimal dictionary learning approach is very common and has been explored by many authors [18], [33], [36] and obtained impressive results. Thus, for training data matrix $\mathbf{Y} = \begin{bmatrix}\mathbf{y}_1 \cdots \mathbf{y}_r \cdots \mathbf{y}_N\end{bmatrix}$ and for corresponding latent codes $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_r \cdots \mathbf{x}_N]$, the final learning problem boils down to

$$\begin{aligned}&\underset{\mathbf{d}_1\cdots\mathbf{d}_m}{\text{minimize}} \sum_{r=1}^{N} \mathbf{pS}(\mathbf{y}_r, \mathbf{Dx}_r)\\ &\text{subject to } \left\|\mathbf{d}_j\right\|_2 = 1 \ \text{ for } \ j = 1\cdots m\\ &\qquad\qquad \langle \mathbf{y}_r, \mathbf{Dx}_r \rangle + c > 0 \ \text{ for } \ r = 1\cdots N\end{aligned} \tag{11}$$

Clearly, the problem in Eq. (11) induces a non-linear fractional program. The pioneering work [51] of Dinkelbach, may be seen as a precursor of such fractional programing. In his seminal article, constrained convex by concave (natural extension for minimization task) program has been iteratively solved, as shown in Algorithm 2. In our case, another interesting benefit of using Algorithm 2 is that the operation 5 given below can be solved for entire dataset in order to learn dictionary **D** atom-wise.

In order to do so, our plan is to map the constrained problem in Eq. (11) to operation 5 of Algorithm-2. Then, operation 5 is converted into an unconstrained objective using method of multiplier (MOM) [52] and finally optimized using coordinate descent algorithm [53], [54].

By incorporating definition of **pS** function in the Operation 5 of Algorithm 2, we have following objective function in matrix form (for simplicity we omit iterate variable k)

$$F = \|\mathbf{Y} - \mathbf{DX}\|_F^2 - \mathbf{Tr}((\mathbf{YS}_t)^T \mathbf{DX}) \tag{12}$$

where $\mathbf{S}_t$ is a diagonal matrix $diag(t_1 \cdots t_r \cdots t_N)$ such that any $t_r = \mathbf{pS}(\mathbf{y}_r, \mathbf{Dx}_r)$. In terms of the ith atom contribution, Eq. (12) reduces to

$$\left\| \mathbf{Y} - [\mathbf{DX}]_{/i} - \mathbf{d}_i {\mathbf{x}^r}_i^T \right\|_F^2 - \mathbf{Tr}((\mathbf{YS}_t)^T ([\mathbf{DX}]_{/i} + \mathbf{d}_i {\mathbf{x}^r}_i^T)) \tag{13}$$

Here, $[\mathbf{DX}]_{/i}$ encodes $\mathbf{DX}$ without ith column and row of $\mathbf{D}$ and $\mathbf{X}$, respectively and ${\mathbf{x}^r}_i^T$ is the ith row of $\mathbf{X}$. Suppose, $\mathbf{E}_{/i} = \mathbf{Y} - [\mathbf{DX}]_{/i}$ and denoting ${\mathbf{x}^r}_i^T (\mathbf{YS}_t)^T$ by $\boldsymbol{\beta} \in \mathbb{R}^n$ (for simplicity we omit the index i), then Eq. (13) reduces to

$$\left\| \mathbf{E}^T - \mathbf{x}^r \mathbf{d}^T \right\|_F^2 - \boldsymbol{\beta}^T \mathbf{d} \tag{14}$$

Therefore, the term affecting only the jth component of $\mathbf{d}$ is

$$\left\| \mathbf{E}_j^T - \mathbf{x}^r d_j \right\|_2^2 - \beta_j d_j \tag{15}$$

where $\mathbf{E}_j^T$ is the jth column of $\mathbf{E}^T$ matrix. Next, we simplify the constrained part of operation 5 in Algorithm-2. For this, we first deal with the constrained set $\langle \mathbf{x}_r \mathbf{D}, \mathbf{y}_r \rangle + c > 0$ for $r = 1 \cdots N$. The constraint associated with arbitrary observation $\mathbf{y}$ and corresponding latent variable $\mathbf{x}$ can be written as

$$\xi \equiv \sum_{p \neq i}^{m} x_p (\mathbf{d}_p^{\mathrm{T}} \mathbf{y}) + x_i (\mathbf{d}_i^{\mathrm{T}} \mathbf{y}) + c = g + x_i (\mathbf{d}_i^{\mathrm{T}} \mathbf{y})$$

We see that the contribution of the jth coordinate is (for simplicity the index i is omitted)

$$\xi_j \equiv g' + x y_j d_j$$

where g' is equals to $g + x \sum_{k \neq j} d_k y_k$.

Suppose index set $\mathcal{I}$ is defined such that $\mathcal{I} \subseteq \{1 \cdots N\}$ and $r \in \mathcal{I}$ if rth latent-observation pair $(\mathbf{x}_r, \mathbf{y}_r)$ violates the constraint. Therefore, the augmented penalty term of MOM [52] for jth coordinate becomes

$$\sum_{r \in \mathcal{I}} [(g'_\rho + x y_j d_j)^2]_r - \rho_r^2 \tag{16}$$

where $[.]_r$ operator represents argument with respect to rth observation, $g'_\rho = g' + \rho$ and, ρ_r is the multiplier corresponding to rth observation. Considering only relevant in Eq. (16) containing optimization variables, we have

$$\begin{aligned} 2d_j \sum_{r \in \mathcal{I}} [g'_\rho x y_j]_r + d_j^2 \sum_{r \in \mathcal{I}} [(x y_j)^2]_r \\ \equiv 2 d_j \omega + d_j^2 \varphi \end{aligned} \tag{17}$$

where $\sum_{r \in \mathcal{I}} [g'_\rho x y_j]_r$ and $\sum_{r \in \mathcal{I}} [(x y_j)^2]_r$ are denoted by ω and φ, respectively. Finally, combining Eq. (15) and (17), we can easily obtain the MOM expression. The resulting MOM expression is finally optimized to get the update rule given below for the jth component of dictionary atom $\mathbf{d}$.

$$d_j = \frac{\langle \mathbf{E}_j^T, \mathbf{x}^r \rangle + \beta_j/2 - \lambda\omega}{\|\mathbf{x}^r\|_2^2 + \lambda\varphi} \tag{18}$$

where λ is the associated penalty parameter. The update rule for any rth multiplier ρ_r is

$$\rho_r^+ := \begin{cases} \rho_r^- + \langle \mathbf{x}_r \mathbf{D}, \mathbf{y}_r \rangle + c & r \in \mathcal{I} \\ \rho_r^- & else \end{cases} \tag{19}$$

After updating all components of $\mathbf{d}$ using Eq. (18), we finally normalize the atoms of the dictionary $\mathbf{D}$. In this way, after executing updates for each atom, we have one dictionary $\mathbf{D}$ update. Note that the above update rule can be easily implemented in the distributed computing environment. The performance of this learning technique will be discussed in application Section VI.

B. Choice of Solver

Choosing a solver to solve convex program in Eq. (9) or (10) for latent variable $\mathbf{x}$ is an important issue. Recently, Brendan *et. al.* [55] originated a first order split conic solver (SCS). Its attractive properties include low runtime complexity, efficient handling of a broad range of large scale problems and issuance of a token for infeasibility and unboundedness of the convex program. Briefly, the SCS forms a linear KKT system for the prototype convex problem and embeds two new variables, which decode the status of the solution. This system is proven to be self-dual and further posed as the feasibility problem. The alternative direction multiplier (ADMM) is used to solve posed feasibility problem. In order to improve the convergence rate, over-relaxation and relative scaling strategies have been adopted (for detail see [55]).

Computationally significant components in the SCS solver are initialization and the iterative computing module. Main modules of initialization are sparse LDL factorization [56] and relative scaling. Sparse LDL factorization is a direct method for efficient subspace projection and relative scaling benefits the convergence rate in practice. It is worth mentioning that if for different inputs, the problem in hand permits no alteration of data used in factorization, then the entire initialization component of the solver is cacheable. For such cases, the computational efficiency significantly improves, depending on problem structure. Later we use this observation for improving efficiency for a particular case.

V. Image Patches and Clustering

Patch based methods stack patches, e.g. , $\mathbf{y} \in \mathbb{R}^m$ (m is the size of image patch) into a *image to patch buffer* $\mathbf{Y} = \left[\mathbf{y}_1 \cdots \mathbf{y}_N\right]$. Each element $\mathbf{y}$ of this patch-buffer captures localized nature, for instance, homogeneous, texture, edges, etc of the given image. Milanfar [57] suggested that an image filter behaves differently for flat, edged or multi-oriented patches. Guoshen and Shapiro [40] used oriented nature of patches to initialize an mixture model. Wang and Moral [58] also included the patch nature in their Gaussian factor mixture and finally adapted filtering by SURE [59] aid. We wish to take benefit of such segregation. For this, we use oriented information to make clusters using Harris detector [60]. In detail, gradient pair $(\Delta \mathbf{y}_x, \Delta \mathbf{y}_y)$ of each patch $\mathbf{y}$ is computed to form 2×2 matrix $\mathbf{M_y} = [\Delta \mathbf{y}_x \ \Delta \mathbf{y}_y][\Delta \mathbf{y}_x \ \Delta \mathbf{y}_y]^T$. Afterward, the eigen value pair (λ_h, λ_l) such that $\lambda_h \geq \lambda_l$ is computed efficiently for each $\mathbf{M_y}$ using,

$$\lambda_h = 0.5\left(\mathbf{Tr}(\mathbf{M_y}) + \sqrt{\mathbf{Tr}^2(\mathbf{M_y}) - 2\,\mathbf{det}(\mathbf{M_y})}\right)$$

$$\lambda_l = 0.5\left(\mathbf{Tr}(\mathbf{M_y}) - \sqrt{\mathbf{Tr}^2(\mathbf{M_y}) - 2\,\mathbf{det}(\mathbf{M_y})}\right)$$

where **det** and **Tr** are the determinant and the trace operator respectively. The overall procedure for labeling each patch in $\mathbf{Y} = \left[\mathbf{y}_1 \cdots \mathbf{y}_N\right]$ is in two-steps. In the first step, we labeled any patch as oriented or non-oriented and in the second step, non-oriented is classified into flat or multi-oriented label. To execute the first step, the ratio λ_h/λ_l is used, since it captures orientation information [58], [60]. Intuitively, if this ratio is high, this means $\mathbf{M_y}$ approaches to rank 1 and therefore, the patch exhibits orientation. We adaptively computed a threshold value to declare a patch to be oriented. In our implementation, the threshold value is identified by first reducing the dynamic range of λ_h/λ_l for all vectors in $\mathbf{Y}$ using sigmoid function [61] followed by normalization. Afterward, the histogram is computed and finally Otsu's method [62] is applied to obtain the threshold value.

All patches having higher values than the obtained threshold are identified as oriented patches and rest as non-oriented. We have also tried various permutations to fit a sequence of fixed threshold values on different images taken from BSD dataset[1] for the iterative scheme given in the Algorithm 3. However, the threshold values computed from Otsu's method works better. Intuitively, it may be because Otsu's method uses statistical properties of available data. On the other hand, it is very hard to find an optimal sequence of fixed threshold values that performs well independently of the data characteristics, especially in the iterative algorithm settings. In the second step, we further classified non-oriented patches into flat and multi-oriented patch labels. For this, we collected λ_h for all non-oriented patches. Subsequently, normalized them and evaluated mean μ_{λ_h} and variance σ_{λ_h}. Finally, the threshold value was $\mu_{\lambda_h} - 3\sigma_{\lambda_h}$. If the normalized λ_h value of a non-oriented patch has lower value than the threshold, we assign its candidacy to the flat group, otherwise to the multi-oriented group. Furthermore, to find the exact membership of candidate $\mathbf{y}$ in the flat group, the following low cost test was applied

$$\sigma_{\mathbf{y}} \leq C\sigma_{-}$$

where $C = 1.6$ and $\sigma_{\mathbf{y}}$ and σ_{-} are the estimated variance for patch and image, respectively. Here, image variance σ_{-} can be computed using suitable algorithm, for instance Algorithm 3 in Section VI uses [63]. Only if the given patch passes the above test it is labeled as flat: otherwise it is labeled as multi-oriented. An alternative useful interpretation of $\sigma_{\mathbf{y}}$ is the deviation of $\mathbf{y}$ from its mean $\mu_{\mathbf{y}}$ value in the chi-square sense and therefore, it is a simple chi-square test against the expected DC value. An example of such a labeling is shown in figure 1.

Algorithm 3 Proposed Ps-Sparse Denoising Scheme

```
Input: Noisy Image: $\mathbf{I}_{ob}$, Dictionaries set $\mathbf{\Phi} = [\mathbf{D}_1\, \mathbf{D}_2 \cdots \mathbf{D}_{170}]$, Regularization parameter set: $\{\gamma_l\}_{l=1}^{170}$, Orientation map: $O(\mathbf{I}_{ob})_{map}$, Cluster centers set: $\mathbf{\Xi} = [\mathbf{c}_1\, \mathbf{c}_2 \cdots \mathbf{c}_{170}]$, Noise Sigma: $\sigma$, Max Iterate: $K$.
Initialize: Obtain patch buffer: $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_n \cdots \ \mathbf{y}_N]$ from noisy image $\mathbf{I}_{ob}$ and set $\sigma_{-} := \sigma$
For $k = 1$ to $K$ do
    • Clustering: Clustering on patch buffer $\mathbf{Y}$ to get $\{\mathbf{Y}^{(l)}\}_{l=1}^{170}$ using orientation map $\mathbf{O}_{map}$ and cluster centers set $\mathbf{\Xi}$.
    • Compute SSIM sparse codes: $\mathbf{X} = \{\mathbf{X}^{(l)}\}_{l=1}^{170}$ using Eq. (22) such that $\mathbf{D} := \mathbf{D}_l$ and $\lambda := \lambda_l = (\gamma_l \sigma_{-}^2)/\sigma_{\mathbf{y}}$ for each $l$.
    • SSIM estimate: $\mathbf{y'}_n = \mathbf{D}_l \mathbf{x}_n$ for corresponding cluster label $l$ and for $n = 1 \cdots N$.
    • Project on feasible-ball:
      For each $\mathbf{y'}_n$ , obtain
      $\mathbf{x'}_n \leftarrow \underset{\mathbf{x}}{\text{minimize}} \|\mathbf{x}\|_s \ s.t. \ \|\mathbf{y'}_n - \mathbf{D}_{sub}\mathbf{x}\|_2^2 \leq C_1\sigma_{-}^2$
      For each $\mathbf{r}_n = \mathbf{y} - \mathbf{y}'_{\mathbf{n}}$ , obtain
      $\mathbf{r'}_n \leftarrow \underset{\mathbf{x}}{\text{minimize}} \|\mathbf{x}\|_s \ s.t. \ \|\mathbf{r}_n - \mathbf{D}_{tiny}\mathbf{x}\|_2^2 \leq C_2\sigma_{-}^2$
      Therefore patch estimate,
      $\hat{\mathbf{Y}}' = [\hat{\mathbf{y}}'_1 \cdots \ \hat{\mathbf{y}}'_n = \mathbf{D}_{sub}(\mathbf{x'}_n + \mathbf{r'}_n) \ \cdots \hat{\mathbf{y}}'_N]$
    • Compute denoised image: $\hat{\mathbf{I}}_{ac}$ by performing patch averaging on patch set $\hat{\mathbf{Y}}'$
    • Update sigma: $\sigma_{-} := \sigma_{+}$ using [63]
    • Derive patch vector set: $\mathbf{Y}$ from $\mathbf{I}$ such that
          $\mathbf{I} = \hat{\mathbf{I}}_{ac} + \delta(\mathbf{I}_{ob} - \hat{\mathbf{I}}_{ac})$
    if $\sigma \geq 20$ && $k \leq 3$ then
        • Compute: $O(\mathbf{I})_{map}$
    End if
End For
```

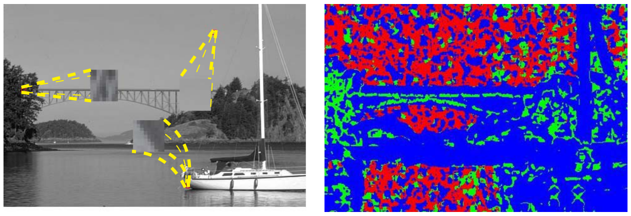

Fig. 1. Oriented patches example. *Left*: Zoomed (location marked by dotted yellow lines) oriented, flat and multi oriented patches in an Image. *Right*: Corresponding map (O_{map}) (a) Multi Oriented (green) (b) Oriented (blue) (c) flat (red).

We further localize patch-nature using clustering in the oriented and multi-oriented groups. The clustered patch methods have shown improvement in performance, for example, in the seminal work of Dong *et. al.* [41], [64]. We apply the well known K-mean [61] clustering technique in oriented and

[1] www.eecs.berkeley.edu/Research/Projects/CS /vision/bsds/

multi-oriented groups. However, instead of directly applying K-means on raw data, high frequency information, as suggested in [64], [65], is used in the K-means algorithm. In summary, the above mentioned clustering approach is stepwise and resembles divisive (top-down) type hierarchical clustering [61]. The main difference is that at each different level, the splitting algorithm is different. The applicability and robustness of the said clustering scheme will be discussed with respect to denoising application in Section VI.

VI. Experiment: Denoising and Deblurring

Denoising problem aims to find clean image $\mathbf{I}_{ac}$ from observable noisy image $\mathbf{I}_{ob}$. For our case we assume noise is additive white Gaussian noise $\mathcal{N}$ (AWGN) of zero mean and standard deviation σ. The mathematical representation of the problem is

$$\mathbf{I}_{ob} = \mathbf{I}_{ac} + \mathcal{N} \tag{20}$$

Notice that because of additive Gaussian corruption, it is hopeless to get actual clean image $\mathbf{I}_{ac}$ and therefore, at best anyone can only hope for its good estimate $\hat{\mathbf{I}}_{ac}$. The goodness of the estimate is measured by widely acceptable measures, e.g., PSNR, SSIM, etc. Researchers including Dabov et. al. [66], Wang and Morel [58], Elad and Aharon [33], Dong et. al. [41], [64], Yu et. al. [40], Zhang et. al. [44], Niknejad et. al. [43] and many more have shown great interest in this non-obvious problem. We further assume all image patches $\mathbf{y}$ obey the following data model.

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n} \tag{21}$$

where, $\mathbf{D}$ is the dictionary matrix and $\mathbf{x}$ is the latent variable. Powerful theoretical analysis from compressive sampling suggests to choose an overcomplete $\mathbf{D}$ and sparse $\mathbf{x}$, as the governing module.

Briefly, our patch based denoising procedure includes clustering as preprocessing (elaborately discussed in Section V) and sparsity prior general formulation discussed in Section IV (see details in the Subsection VI-A). Afterwards, statistical information about noise was also included in the scheme (discussed in the Subsection VI-B). A detailed description of the scheme is presented in Algorithm 3.

A. Formulation Fixing

In Section III we discussed embedding the **pS** function in the general alternate learning procedure. In this section we move one step further and apply ℓ_1-norm regularized linear overcomplete model on general formulation given in Eq. (9). Subsequently, the latent coding stage of learning Algorithm 1 boils down to sparse coding. The targeted optimization problem for our denoiser is translated into the following form

$$\begin{aligned} &\underset{\mathbf{D},\mathbf{x}}{\text{minimize}} \; p + q + \lambda \|\mathbf{x}\|_1 \\ &\text{subject to } {\mathbf{d_y}}^T\mathbf{x} + c \geq 0, q - p \geq 0 \\ &\qquad\qquad \left\| \begin{matrix} \mathbf{y} - \mathbf{D}\mathbf{x} \\ q \end{matrix} \right\| \leq p \end{aligned} \tag{22}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{D} \in \mathbb{R}^{m\times n}$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{d_y} \in \mathbb{R}^n$ are observable patch vectors, overcomplete dictionary ($m < n$), sparse vector and projection of the observation on dictionary rows, respectively. For our experiments, settings of m and n are 64 and 256, respectively. It is worthwhile to mention that the problem in (22) admits a non-unique solution similar to its MSE based counterpart. For a given measurement $\mathbf{y}$, if $(\mathbf{D}^*, \mathbf{x}^*)$ attains optimality, then by choosing $\mathbf{P} = diag(\mathbf{p})$ for $\mathbf{p} \in \{-1, +1\}^n$, $(\mathbf{D}^*\mathbf{P}, \mathbf{P}\mathbf{x}^*)$ are also optimal points. Note that the final estimate is the product of $\mathbf{D}$ and $\mathbf{x}$ and therefore, the mentioned ambiguity has no effect on the performance.

B. Algorithmic Calibration

For our denoiser, we assume that the variance σ of the added AWGN is known. We can therefore embed this statistical information in the denoiser to boost the performance. Many state-of-the-art denoisers leverage such information; e.g., the KSVD denoiser [33], the SA-BM3D [66] denoiser, etc. In our case, suppose the given patch vector (say $\mathbf{y}$) selects Eq. (22) after identifying its cluster label (for details see Section V) for further processing and the obtained latent (sparse) code is $\mathbf{x}$. As a result the patch estimate $\hat{\mathbf{y}}$ of patch vector $\mathbf{y}$ using pS-sparse data model given in Eq. (22) is $\mathbf{D}\mathbf{x}$. Consequently, the support of obtained sparse $\mathbf{x}$ encodes the required dictionary atoms in $\mathbf{D}$. So to enforce obtained sparsity from Eq. (22) a sub-dictionary $\mathbf{D}_{sub}$ is constructed using support of sparse $\mathbf{x}$. Further, based on noise characteristics of the given problem statement, the estimated patch $\hat{\mathbf{y}}$ is required to project on the variance-ball centered at $\hat{\mathbf{y}}$ using sub-dictionary $\mathbf{D}_{sub}$. In addition to this, we further enforce sparsity by including sparse promoting attributes. Consequently, the final projection equation becomes

$$\mathbf{x}' \leftarrow \underset{\mathbf{x}}{\text{minimize}} \|\mathbf{x}\|_s \quad s, t, \; \left\|\hat{\mathbf{y}} - \mathbf{D}_{sub}\mathbf{x}\right\|_2^2 \leq C_1\sigma^2 \tag{23}$$

where s is a sparse promoting norm. Indeed, to keep computation low and maintain sparsity criteria, we choose the value of s to be 0. Interested readers may refer to [67]–[69] for detailed discussion on ℓ_0 and ℓ_1 sparsity and their interconnection. This projection module includes necessary smoothness in the denoised image. Interestingly, a similar strategy has been explored by Wang and Simoncelli [70] in the gradient ascent/descent scheme for image synthesis. The inspiring work of Dong et. al. [41] has further motivated us to embed smooth sparse coefficients correction. For this, the sparse coefficients of the dominated atoms of $\mathbf{D}_{sub}$ are further corrected by patch residual $\mathbf{r}$. where $\mathbf{r}$ is equal to $\mathbf{y} - \hat{\mathbf{y}}$. We assume dictionary $\mathbf{D}_{sub}$ atoms corresponding to non-zero coefficient in $\mathbf{x}'$ (Eq. (23)) to be the dominated atoms (say $\mathbf{D}_{tiny}$). Corresponding sparse coefficients $\mathbf{r}'$ are computed using Eq. (23). Therefore, the final smooth sparse code becomes $\mathbf{x}' + \mathbf{r}'$. However, for further structural correction on the obtained patch estimate, we again back-project this estimate using Eq. (22). Retrospectively, this back and forth projection naturally boils down to an iterative scheme. A detailed description of the proposed SSIM denoiser scheme is presented in Algorithm 3. The coming sections discuss the learning procedure settings and denoiser settings in detail.

C. Comments on Learning Procedure and Setting

We have collected images from the BSD training dataset[2] and converted them into gray scale. Afterward, labeling of each patch for all images using stepwise clustering (explained in Section V) is performed. At the end of the clustering process, 70 subsets of patches for the oriented class and 100 patch subsets for the multi-oriented class are formed. Note that the number of subsets chosen is based on the elbow method.[3] These subsets are learned independently using Algorithm 1 of Section III. The latent coding step is processed using the smooth translated version of Eq. (22) given in Eq. (38) via SCS solver. Additionally, the parameter (dictionary) learning step is accomplished by Algorithm 2 for one iteration, where operation 5 is performed by Eq. (18) and (19) for three iterations.

The $\mathbf{d_y}$ combines training example $\mathbf{y}$ with optimization variable $\mathbf{x}_+$ and $\mathbf{x}_-$ in a linear equality constraint. Therefore, Eq. (38) becomes a non-separable linear equality constraint. The relative scaling parameters $\mathbf{D}, \mathbf{E}, \sigma$ and ρ of the SCS solver [55] depend on this linear equality constraint. Consequently, we have non-cacheable factorization. Therefore, a significant portion (15-25% in our case) of computing time per example patch is spent in the factorizyation.

This observation characterizes some relevant and interesting questions for Eq. (38). Is it really necessary to extract the scaling parameter for each example separately? If not, then how could we obtain a global parameter at least for an individual cluster? and what are the necessary changes we need to make in the optimizer? To answer these questions we studied 2 methods, namely (A) and (B). In method (A) learning process, we apply scaling and factorization for each example individually. In method (B), we pick the individual cluster center (since, in a mean square error sense, the cluster centers are the closest approximation to the respective cluster members) to generate scaling and factorization and use it for the learning of each cluster member. The said task also requires necessary modification in LDL factorization of the SCS solver which is derived in appendix B. We conduct extensive experiments for denoising problem under cacheable (method (B)) and non-cacheable (method (A)) initialization and report this in Table-I. Further discussion on these results will be carried out in the results Section-VII.

The Δ and maximum iterate K, settings of alternating learning Algorithm-1, are set to 10^{-2} and 80, respectively. The penalty parameter λ in Eq. (18) is set to 0.1. Additionally, the intrinsic parameters of the SCS solver for sparse (latent) coding stage (Eq. (38)) are 500, 10^{-3}, 5 and 1.8 for maximum solver iterate, solver tolerance, relative scaling and over relaxation, respectively for both (A) and (B) methods. Further, c and λ parameters of the Eq. (38) are set to 1 and 0.3, respectively. Finally, all learned dictionaries are initialized with the overcomplete DCT dictionary.

D. Comments on Denoiser Procedure and Setting

For an appropriate setting of the denoiser, in Eq. (24), we again analyze the denominator of Eq. (5) with respect to the data model Eq. (21). Notice that the Eq. (24) gives rise to some interesting and insightful thoughts.

$$\begin{aligned}\|\mathbf{y}\|_2^2 + \|\mathbf{z}\|_2^2 &= \langle \mathbf{y},\mathbf{y}\rangle + \langle \mathbf{z},\mathbf{z}\rangle \\ &= \langle \mathbf{y},\mathbf{Dx}+\mathbf{n}\rangle + \langle \mathbf{Dx},\mathbf{y}-\mathbf{n}\rangle \\ &= 2\langle \mathbf{y},\mathbf{Dx}\rangle + \langle \mathbf{y}-\mathbf{Dx},\mathbf{n}\rangle \\ (absorbing\ 2) &= \langle \mathbf{y},\mathbf{Dx}\rangle + 0.5\langle \mathbf{n},\mathbf{n}\rangle \\ (in\ numerator) &= \langle \mathbf{y},\mathbf{Dx}\rangle + 0.5\,\|\mathbf{n}\|_2^2 \end{aligned} \tag{24}$$

First, if ancillary scalars are omitted, we observe that $\mathbf{pS}$ and T given in Eqs. (5) and (7), respectively approximate each other for $c = 0.5\,\|\mathbf{n}\|_2^2$. Second, the data model and the $\mathbf{pS}$ function are applied locally: therefore, it is sufficient to know a 2^{nd} order characterization of the noise component $\|\mathbf{n}\|_2^2$ at the local level. Notice that number of patches for given image is large, for instance, 512×512 image has 255025 patches of size 8×8. Therefore, computational requirement to estimate $\|\mathbf{n}\|_2^2$ is high. Complexity in the estimation of $\|\mathbf{n}\|_2^2$ may further increase because of the iterative nature of the scheme. Therefore, we applied an heuristic i.e. we assume that the fraction of local observation $\epsilon\,\|\mathbf{y}\|_2^2$ approximates $0.5\,\|\mathbf{n}\|_2^2$. We obtain value of ϵ empirically and kept it constant for all test images in Fig-2. Corresponding results are reported in Table-II and Table-I for both methods (A) and (B). Flat patches represent homogeneous regions in real world images. As discussed in Subsection II-B, the $\mathbf{pS}$ function is applied on zero mean patch vectors and therefore suitable patches should have significant variation. On the contrary, flat patches have almost zero variation and thus, require a different mechanism to clean them. For this, we iteratively perform group-wise thresholding on such flat patches. First, inside a local window, similar flat patches are collected and then subjected to only group-wise thresholding in 3D transformed domain [17]. Finally, patches are aggregated in their respective locations.

The settings of intrinsic parameters of SCS solver denoiser 3 to perform sparse (latent) coding (38) are 300, 10^{-2}, 5 and 1.5 for maximum solver iterate, solver tolerance, relative scaling and over relaxation, respectively for both the proposed methods (A) and (B). The ϵ in Eq. (24) is set to 0.4 for lower noises ($\sigma \leq 40$), else 0.15. The regularization parameter for each cluster γ_l such that $1 \leq l \leq 170$ is obtained by cross-validation from cluster data set added with AWGN set at σ=20. Note that we have evaluated the regularization set independently for the methods (A) and (B). However, in method (B), the cluster center is used to set the relative scaling parameters of the SCS solver. In the upcoming section, a detailed comparison of proposed denoiser with other state-of-the-art denoising schemes is explored and discussed.

VII. Results and Discussions

To verify the performance of our proposed methodology, twenty test images (shown in Fig-2) were chosen. The proposed methods (A) and (B) were executed in Matlab environment. For dictionary learning, the method (B) was significantly faster than the method (A). This was due to the large training size of patch-buffer per cluster and subsequently, the cumulative initialization time was significant (approximately 15-20 min) per cluster. However, denoising time of

[2] www.eecs.berkeley.edu/Research/Projects/CS /vision/bsds/

[3] https://bl.ocks.org/rpgove/0060ff3b656618e9136b

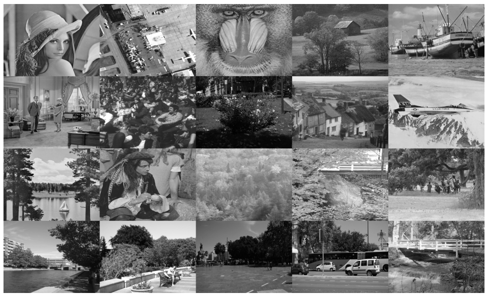

Fig. 2. Twenty test images labeled as 1 to 20. From left to right and top to bottom.

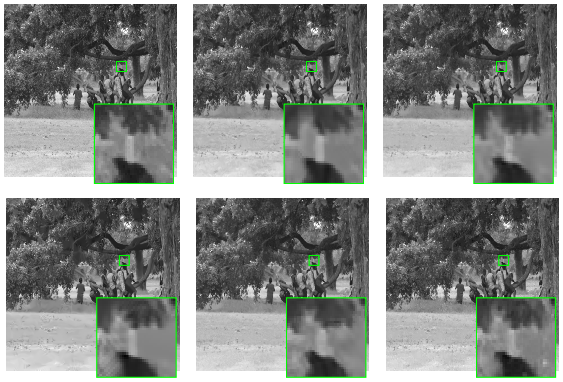

Fig. 3. Denoising comparison for 4th test image with moderate noise ($\sigma = 20$) corruption. *Top row*: original, the SA-BM3D [66] (SSIM = 0.781), the NCSR [41] (SSIM = 0.773). *Bottom row*: the LINC [43] (SSIM = 0.759), the GHP [44] (SSIM = 0.777) and the proposed method (SSIM = 0.779).

method (B) was 0.5-1.0 min faster than method (A). The average denoising time for method (A) to denoise 256 × 256 image was around 3.0 min on a PC (Intel i7-4790K at 4.4GHz). Our implementation is based on multi-core programming. Performance of method (A) and (B) is compared in the Table-I. It can be seen that method (A) performs better than method (B) for moderate noise. However, performances of both the methods in high noise are very close.

A. Comparison With Sparse Techniques

Table-II reports performance of the proposed method (A) and four other state-of-the-art denoisers. The other denoisers are the SA-BM3D denoiser [66], Gaussian mixture model (GMM) based linear estimator with neighborhood patch clustering (LINC) denoiser [43], the non-locally centralized sparse representation (NCSR) denoiser [41] and the gradient histogram estimation and preservation (GHP) denoiser [44]. Each state-of-the-art denoiser has different traits. For instance, Milad *et. al.* used Gaussian mixture model (GMM) with a novel patch weighting scheme in the LINC denoiser. Zhang *et. al.* pioneered the usage of a gradient histogram in the GHP scheme. Dong *et. al.* addressed the correction of sparse coefficients by non-local centralization in the NCSR method and Dabov *et. al.* embedded principal component analysis based spatial adaptation and collaborative filtering in the SA-BM3D denoiser. In Table-II, the best two performing methods' index are highlighted in the average row.

Additionally, the bottom left of the Fig-7 plots the average SSIM performance (last row of Table-II). Furthermore, one can observe that the proposed method achieves highly competitive denoising performance. In terms of average SSIM results, the proposed method (A) performs nearly to the SA-BM3D for small to moderate noise corruption. For higher noise NCSR and method (A) outperform.

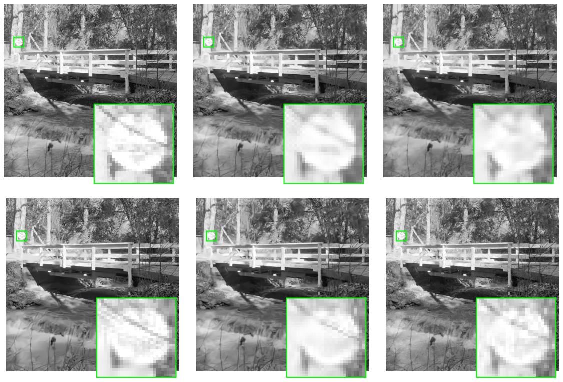

Fig. 4. Denoising comparison for 20th test image with moderate noise ($\sigma = 20$) corruption. *Top row*: original, the SA-BM3D [66] (SSIM = 0.808), the NCSR [41] (SSIM = 0.796). *Bottom row*: the LINC [43] (SSIM = 0.803), the GHP [44] (SSIM = 0.800) and the proposed method (SSIM = 0.812).

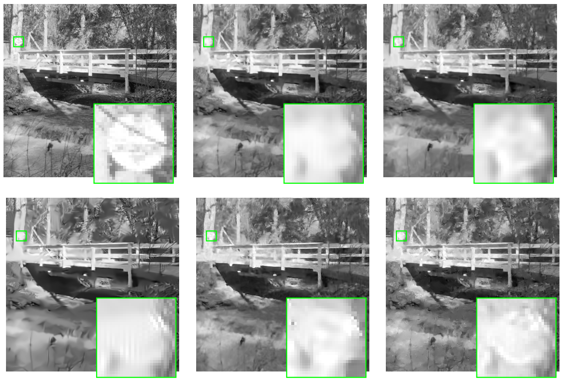

Fig. 5. Denoising comparison for 20th test image with moderate noise ($\sigma = 50$) corruption. *Top row*: original, the SA-BM3D [66] (SSIM = 0.599), the NCSR [41] (SSIM = 0.577). *Bottom row*: the LINC [43] (SSIM = 0.570), the GHP [44] (SSIM = 0.568) and the proposed method (SSIM = 0.601).

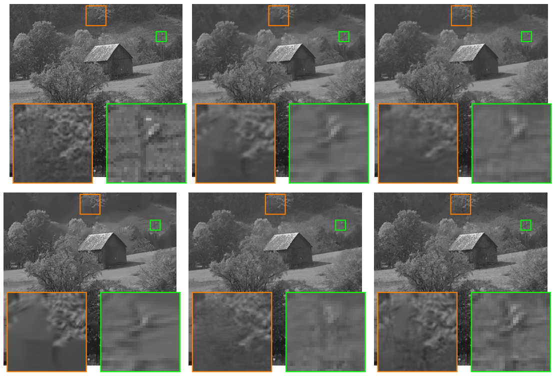

Fig. 6. Denoising comparison for 4th test image with moderate noise ($\sigma = 20$) corruption. *Top row*: original, the SA-BM3D [66] (SSIM = 0.714), the NCSR [41] (SSIM = 0.704). *Bottom row*: the LINC [43] (SSIM = 0.703), the GHP [44] (SSIM = 0.713) and the proposed method (SSIM = 0.732).

Fig-6, 3, 4 and 5 visually compare various denoiser responses for some test images. In Fig-6 (moderate corruption i.e. $\sigma = 20$), one can observe that SA-BM3D, GHP and the proposed method (A) perform well, corresponding to

TABLE I

THE PSNR (dB) AND SSIM VALUES OF PROPOSED METHOD(A) AND METHOD(B)

$\sigma \longrightarrow$	**Method(A)**				**Method(B)**			
	10	20	50	75	10	20	50	75
1	35.77 0.915	32.79 0.830	28.65 0.791	26.82 0.751	35.62 0.901	32.66 0.819	28.59 0.787	26.77 0.748
2	31.35 0.861	28.18 0.761	24.26 0.604	22.61 0.531	31.24 0.851	28.06 0.752	24.20 0.600	22.59 0.527
3	30.49 0.901	26.64 0.802	22.48 0.572	21.02 0.438	30.40 0.893	26.53 0.791	22.43 0.566	20.99 0.435
4	31.79 0.858	28.60 0.732	25.56 0.529	24.57 0.459	31.65 0.849	28.49 0.725	25.51 0.524	24.54 0.456
5	33.80 0.887	30.73 0.819	26.56 0.687	24.80 0.623	33.67 0.876	30.65 0.811	26.52 0.680	24.77 0.620
6	33.86 0.905	30.50 0.833	26.18 0.683	24.35 0.598	33.75 0.897	30.41 0.824	26.13 0.676	24.31 0.594
7	34.86 0.939	31.24 0.886	26.61 0.770	24.68 0.706	34.73 0.927	31.13 0.877	26.56 0.764	24.65 0.703
8	32.19 0.901	28.59 0.806	24.75 0.625	23.34 0.538	32.10 0.892	28.45 0.796	24.70 0.621	23.29 0.535
9	33.58 0.883	30.51 0.795	26.96 0.658	25.47 0.598	33.46 0.875	30.42 0.787	26.93 0.654	25.42 0.594
10	36.11 0.935	32.72 0.899	28.20 0.829	26.14 0.788	36.00 0.928	32.63 0.890	28.16 0.824	26.10 0.784
11	33.17 0.892	30.09 0.832	26.10 0.731	24.29 0.671	33.10 0.885	30.00 0.823	26.04 0.727	24.24 0.665
12	33.96 0.908	30.65 0.834	26.71 0.697	25.14 0.637	33.85 0.899	30.56 0.825	26.67 0.691	25.10 0.634
13	31.77 0.825	28.82 0.690	26.42 0.525	25.64 0.480	31.64 0.813	28.74 0.681	26.39 0.519	25.60 0.476
14	30.48 0.909	26.91 0.811	23.05 0.599	21.58 0.481	30.34 0.897	26.81 0.801	23.00 0.592	21.54 0.476
15	32.97 0.880	29.66 0.799	26.07 0.609	24.79 0.543	32.90 0.872	29.54 0.787	26.01 0.603	24.77 0.540

zoom-box orange. However, LINC recovers details present in the top right corner of the orange zoom-box but over-smooths the other part. The NCSR, although not aggressively smooth inside the orange zoom-box, still weakly preserves detail presented in the top right region of the orange zoom-box. On the other hand, if we compare visual performance in the green zoom-box of Fig-6, the method (A) excels in visual performance compared to other denoisers. It has been observed that if an image contains large smooth regions, (for instance 1st (Lena) test image) the performance of LINC and NCSR denoiser would be excelled as shown in the top right of Fig-7. On inspection of the green zoom-box region of Fig-3, we can identify its diverse nature, i.e. it contains texture portion at the top, almost uniform portion at bottom, connected thin white bar structures and a very fine detailed background. Due to such diverse information, it may shed light on the behavior of different denoisers. For example, the bottom region recovers better in cases of SA-BM3D, NCSR and GHP denoisers, since all these denoisers have correction strategies using neighboring information. However, the top texture portion and background details recover gracefully for the case of method (A) and GHP denoisers. On the other hand, structural information of the vertical bar is preserved in case of LINC, SA-BM3D and the proposed method (A). Finally, Fig-4 and 5 compare 20th (walkbridge) test image for the cases of moderate ($\sigma = 20$) and high noise ($\sigma = 50$), respectively. On inspection of zoom-box green in Fig-4, it is clear that for moderate noise, SA-BM3D, GHP, LINC and the proposed method (A) recover structural information. However, finer details are also recovered gracefully in method (A). On the other hand, for very high noise, the proposed method (A) preserves better structural information than other denoisers as shown in Fig-5. We believe that it is because the proposed algorithm iteratively builds mutual consent between smoothing and structural module

TABLE II

THE PSNR (dB) AND SSIM VALUES OF SA-BM3D, GHP, NCSR, LINC AND PROPOSED METHOD(A) ARE REPORTED. THE BEST 2 AVERAGE VALUES OF PSNR AND SSIM ARE HIGHLIGHTED

$\sigma \rightarrow$	**SA-BM3D[66]**						**GHP[44]**						**NCSR[41]**						**LINC[43]**						**Method(A)**					
	5	10	15	20	50	75	5	10	15	20	50	75	5	10	15	20	50	75	5	10	15	20	50	75	5	10	15	20	50	75
1	38.86 0.946	36.07 0.918	34.43 0.898	33.20 0.881	29.07 0.801	26.83 0.725	38.75 0.946	35.57 0.906	33.50 0.865	32.83 0.871	27.55 0.787	24.26 0.717	38.75 0.945	35.84 0.915	34.11 0.893	32.96 0.876	28.91 0.804	26.99 0.763	37.98 0.938	35.40 0.913	34.01 0.896	32.86 0.878	29.11 0.811	27.19 0.766	38.73 0.946	35.77 0.915	34.06 0.890	32.79 0.870	28.65 0.791	26.82 0.751
2	35.62 0.942	31.60 0.858	29.70 0.799	28.43 0.757	24.55 0.612	22.85 0.547	35.24 0.938	31.44 0.862	29.41 0.804	28.09 0.757	23.97 0.593	21.24 0.518	35.5 0.939	31.39 0.852	29.43 0.791	28.07 0.737	24.15 0.588	22.58 0.526	35.05 0.939	30.54 0.852	28.68 0.788	27.55 0.735	24.10 0.588	22.64 0.523	35.16 0.937	31.35 0.861	29.50 0.805	28.18 0.761	24.26 0.604	22.61 0.531
3	35.39 0.961	30.81 0.904	28.43 0.852	26.84 0.803	22.60 0.579	21.13 0.456	35.14 0.96	30.62 0.902	28.15 0.840	26.64 0.798	21.94 0.567	19.70 0.441	35.29 0.957	30.63 0.894	28.25 0.840	26.64 0.784	22.44 0.550	21.02 0.434	35.01 0.957	30.17 0.898	27.70 0.845	26.10 0.796	22.31 0.584	20.88 0.420	34.67 0.955	30.49 0.901	28.22 0.849	26.64 0.802	22.48 0.572	21.02 0.438
4	35.92 0.941	31.85 0.854	29.87 0.777	28.63 0.714	25.56 0.518	24.39 0.459	35.87 0.94	31.69 0.851	29.50 0.766	28.33 0.713	24.69 0.503	22.73 0.430	35.82 0.937	31.74 0.849	29.75 0.774	28.47 0.704	25.44 0.504	24.41 0.442	35.34 0.935	30.97 0.846	29.06 0.773	27.95 0.703	25.17 0.472	24.28 0.420	35.63 0.938	31.79 0.858	29.85 0.789	28.60 0.732	25.56 0.529	24.57 0.459
5	37.5 0.944	34.10 0.892	32.29 0.857	30.99 0.828	26.84 0.705	24.76 0.620	37.22 0.941	33.73 0.884	31.64 0.831	30.66 0.817	25.90 0.683	22.90 0.599	37.35 0.94	33.91 0.888	32.07 0.852	30.76 0.821	26.51 10.689	24.65 0.626	36.64 0.936	33.27 0.882	31.73 0.850	30.55 0.820	26.61 0.690	24.87 0.625	36.9 0.933	33.80 0.887	32.02 0.851	30.73 0.819	26.56 0.687	24.80 0.623
6	37.62 0.953	34.15 0.911	32.21 0.877	30.83 0.846	26.45 0.704	24.46 0.609	37.37 0.951	33.77 0.902	31.52 0.849	30.48 0.835	25.46 0.680	22.65 0.585	37.47 0.95	33.98 0.908	31.96 0.871	30.54 0.837	26.12 0.690	24.31 0.610	36.69 0.944	33.27 0.901	31.53 0.866	30.29 0.833	26.27 0.696	24.39 0.604	37.4 0.951	33.86 0.905	31.88 0.868	30.50 0.833	26.18 0.683	24.35 0.598
7	39.02 0.974	35.06 0.945	32.85 0.917	31.33 0.892	26.71 0.777	24.67 0.693	38.64 0.971	34.60 0.928	32.21 0.882	31.08 0.880	25.75 0.760	22.81 0.677	38.84 0.973	34.83 0.942	32.63 0.914	31.12 0.889	26.46 0.775	24.67 0.716	38.11 0.966	34.19 0.934	32.19 0.908	30.77 0.882	26.40 0.769	24.48 0.701	38.79 0.971	34.86 0.939	32.72 0.912	31.24 0.886	26.61 0.770	24.68 0.706
8	36.84 0.961	32.44 0.907	30.15 0.854	28.66 0.807	24.74 0.619	23.28 0.529	36.56 0.957	32.13 0.889	29.75 0.817	28.45 0.800	24.08 0.615	21.71 0.534	36.68 0.958	32.25 0.901	29.98 0.849	28.45 0.797	24.59 0.619	23.21 0.544	36.18 0.953	31.53 0.894	29.22 0.842	27.76 0.792	24.31 0.607	22.96 0.518	36.38 0.955	32.19 0.901	30.02 0.850	28.59 0.806	24.75 0.625	23.34 0.538
9	37.29 0.945	33.82 0.890	32.04 0.846	30.85 0.811	27.09 0.670	25.29 0.592	37.17 0.943	33.54 0.880	31.49 0.822	30.58 0.803	25.91 0.645	23.45 0.565	37.16 0.943	33.70 0.887	31.86 0.841	30.61 0.801	26.92 0.657	25.41 0.598	36.43 0.936	32.91 0.877	31.36 0.834	30.27 0.793	26.91 0.651	25.38 0.586	37.04 0.943	33.58 0.883	31.75 0.836	30.51 0.795	26.96 0.658	25.47 0.598
10	39.66 0.962	36.27 0.938	34.30 0.920	32.87 0.903	28.23 0.825	26.00 0.743	39.16 0.961	35.76 0.924	33.52 0.880	32.60 0.894	27.44 0.822	23.87 0.754	39.59 0.961	36.16 0.938	34.17 0.919	32.75 0.903	28.23 0.834	26.38 0.796	38.92 0.956	35.57 0.932	33.72 0.914	32.37 0.898	28.12 0.830	26.24 0.791	39.56 0.960	36.11 0.935	34.15 0.916	32.72 0.899	28.20 0.829	26.14 0.788
11	37.08 0.947	33.40 0.895	31.57 0.862	30.32 0.838	26.34 0.740	24.46 0.667	36.88 0.944	33.11 0.883	31.08 0.831	30.07 0.827	25.37 0.722	22.46 0.648	36.96 0.944	33.27 0.892	31.42 0.860	30.16 0.833	26.15 0.734	24.45 0.683	36.44 0.942	32.52 0.885	30.80 0.851	29.68 0.825	26.07 0.732	24.27 0.671	36.51 0.937	33.17 0.892	31.50 0.861	30.09 0.832	26.10 0.731	24.29 0.671
12	38.00 0.956	34.2 0.912	32.14 0.873	30.77 0.839	26.89 0.709	25.08 0.630	37.76 0.954	33.84 0.900	31.59 0.844	30.53 0.831	25.87 0.687	23.15 0.608	37.83 0.954	34.00 0.908	31.92 0.867	30.51 0.830	26.62 0.699	25.05 0.642	37.16 0.948	33.4 0.903	31.52 0.866	30.26 0.831	26.56 0.692	24.98 0.628	37.63 0.954	33.96 0.908	31.97 0.868	30.65 0.834	26.71 0.697	25.14 0.637
13	35.86 0.926	31.84 0.817	30.00 0.730	28.92 0.666	26.46 0.517	25.48 0.479	35.77 0.925	31.72 0.819	29.74 0.732	28.66 0.675	25.44 0.501	23.46 0.441	35.73 0.922	31.78 0.815	29.98 0.734	28.87 0.664	26.43 0.508	25.52 0.469	35.26 0.920	30.99 0.813	29.29 0.729	28.33 0.649	26.14 0.479	25.37 0.449	35.47 0.923	31.77 0.825	29.98 0.749	28.82 0.690	26.42 0.525	25.64 0.480
14	35.52 0.969	30.89 0.915	28.50 0.858	26.98 0.804	23.10 0.594	21.68 0.500	35.37 0.967	30.76 0.912	28.38 0.852	26.83 0.805	22.46 0.565	20.36 0.454	35.42 0.967	30.75 0.910	28.39 0.851	26.79 0.792	22.88 0.570	21.53 0.474	34.98 0.964	29.99 0.902	27.51 0.844	26.04 0.794	22.71 0.568	21.26 0.440	34.81 0.964	30.48 0.909	28.35 0.856	26.91 0.811	23.05 0.599	21.58 0.481
15	36.85 0.945	33.13 0.881	31.18 0.826	29.89 0.781	26.20 0.621	24.69 0.550	36.72 0.943	32.94 0.875	30.81 0.813	29.69 0.777	25.28 0.600	23.05 0.524	36.73 0.943	33.00 0.877	31.05 0.823	29.73 0.773	26.05 0.602	24.69 0.539	36.02 0.937	32.25 0.868	30.49 0.812	29.30 0.759	25.79 0.575	24.40 0.506	36.60 0.944	32.97 0.88	31.00 0.825	29.66 0.779	26.07 0.609	24.79 0.543
16	38.77 0.969	34.64 0.937	32.36 0.905	30.81 0.876	26.42 0.749	24.61 0.661	38.10 0.965	34.18 0.919	31.78 0.864	30.60 0.866	25.68 0.740	23.01 0.669	38.61 0.967	34.46 0.933	32.17 0.900	30.58 0.868	26.28 0.747	24.73 0.693	37.94 0.962	33.74 0.926	31.52 0.894	30.00 0.861	26.12 0.742	24.53 0.679	38.48 0.966	34.34 0.928	32.16 0.897	30.68 0.867	26.24 0.738	24.62 0.680
17	36.12 0.975	31.03 0.943	28.25 0.905	26.38 0.863	21.67 0.654	20.25 0.547	34.86 0.969	30.60 0.921	27.86 0.856	26.10 0.851	21.20 0.659	18.96 0.551	35.95 0.972	30.84 0.939	28.09 0.900	26.20 0.856	21.53 0.645	20.19 0.556	35.72 0.971	30.46 0.936	27.56 0.900	25.63 0.863	21.06 0.672	19.91 0.554	35.06 0.967	30.14 0.930	27.63 0.883	25.97 0.840	21.68 0.649	20.23 0.544
18	39.23 0.958	35.41 0.914	33.38 0.877	32.07 0.847	28.24 0.737	26.26 0.650	38.73 0.952	34.92 0.895	32.72 0.837	31.78 0.838	26.99 0.730	24.20 0.663	39.02 0.955	35.19 0.910	33.18 0.873	31.82 0.842	28.19 0.744	26.74 0.703	38.29 0.948	34.54 0.903	32.63 0.864	31.31 0.830	28.03 0.734	26.72 0.695	38.77 0.953	35.09 0.906	33.17 0.868	31.88 0.839	28.33 0.742	26.88 0.704
19	37.91 0.973	33.44 0.938	31.04 0.900	29.47 0.864	25.26 0.706	23.56 0.614	37.06 0.967	32.92 0.916	30.50 0.856	29.20 0.850	24.80 0.696	22.04 0.609	37.68 0.970	33.2 0.932	30.88 0.895	29.29 0.858	25.10 0.705	23.52 0.628	37.19 0.967	32.58 0.929	30.16 0.892	28.61 0.858	24.88 0.697	23.39 0.616	37.18 0.968	32.95 0.927	30.74 0.889	29.29 0.853	25.08 0.694	23.54 0.615
20	36.01 0.966	31.56 0.913	29.25 0.859	27.74 0.808	23.88 0.599	22.53 0.508	35.94 0.965	31.42 0.908	28.95 0.843	27.52 0.800	23.25 0.568	21.07 0.466	35.93 0.964	31.41 0.909	29.09 0.853	27.54 0.796	23.72 0.577	22.30 0.484	35.5 0.961	30.80 0.905	28.42 0.852	26.94 0.803	23.49 0.570	22.01 0.447	35.66 0.963	31.37 0.911	29.13 0.857	27.68 0.812	23.85 0.601	22.50 0.498
Avg	**37.25** **0.955**	**33.29** **0.904**	**31.20** **0.859**	**29.80** **0.821**	**25.82** **0.671**	24.11 0.589	36.92 **0.953**	32.96 0.894	30.71 0.834	29.54 0.814	24.95 0.656	22.35 0.573	**37.12** **0.953**	**33.12** **0.900**	**31.02** 0.855	29.59 0.813	25.64 0.662	**24.12** **0.595**	36.54 0.949	32.45 0.895	30.46 0.851	29.13 0.810	25.51 0.658	24.00 0.582	36.82 0.952	33.00 **0.900**	30.99 **0.856**	**29.61** **0.818**	**25.69** **0.667**	**24.15** **0.594**

as explained in Section-VI-B and thus, does not promote aggressive smoothing.

B. Comparison With Deep Learning Techniques

In the era of deep learning, it is interesting to compare proposed method with deep learning model based denoising schemes. Table-III reports performance of the proposed method (A) with other sparsity based schemes and the state-of-the-art deep learning based denoising schemes with respect to three different dataset, namely Kodak24,[4] Test20 (shown in Fig-2) and Test40[5] (shown in Fig-8). The comparing deep learning based frameworks reported in Table-III are Residual learning based CNN (DnCNN) [71] and Deep iterative down-up convolutional neural network (DIDN) [72]. In the DnCNN scheme, Zhang *et. al.* [71] elegantly applied residual learning to separate Gaussian noise using CNN in the discriminative fashion. On the other hand, Yu *et. al.* [72] modifies down-scaling and up-scaling layers of the U-Net [73] for image denoising task in the DIDN scheme. We see

[4] http://r0k.us/graphics/kodak

[5] http://decsai.ugr.es/cvg/CG/base.htm

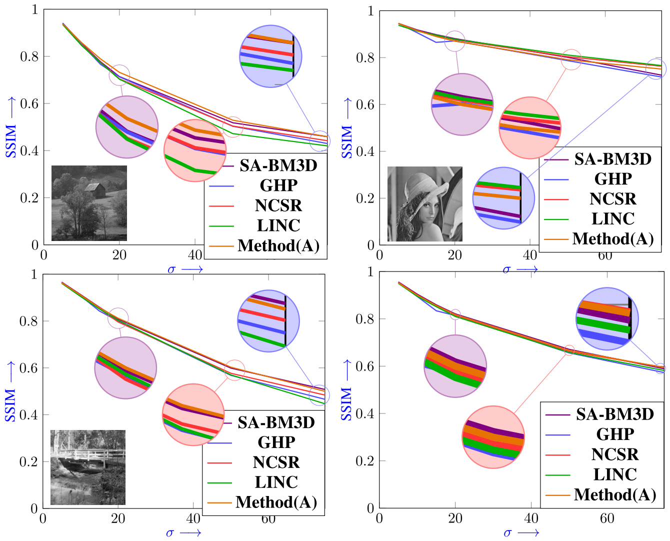

Fig. 7. SSIM vs corruption (σ) Plot: *Top-left*: 4th test image *Top-right*: 1st test image *Bottom-left*: 20th test image *Bottom-right*: Average.

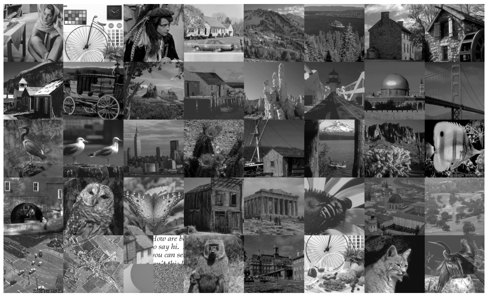

Fig. 8. Test40: Forty test images labeled as 1 to 40. From left to right and top to bottom.

from Table-III that the performance of proposed scheme is highly competitive to other schemes for small and moderate corruption. Fig-9 and 10 visually compare various denoiser responses for given test images. In Fig-9 (moderate corruption i.e. $\sigma = 20$), one can observe that DIDN, DnCNN and the proposed method (A) perform well. However, other methods weakly preserve details presented in the zoom-box. On the other hand, on inspection of the zoom-box region of Fig-10, again we can identify that structural contents are well preserved by DIDN, DnCNN and the proposed method (A). It is interesting to port the proposed **pS** function in the deep learning framework. This requires extensive study separately and will be explored in the near future.

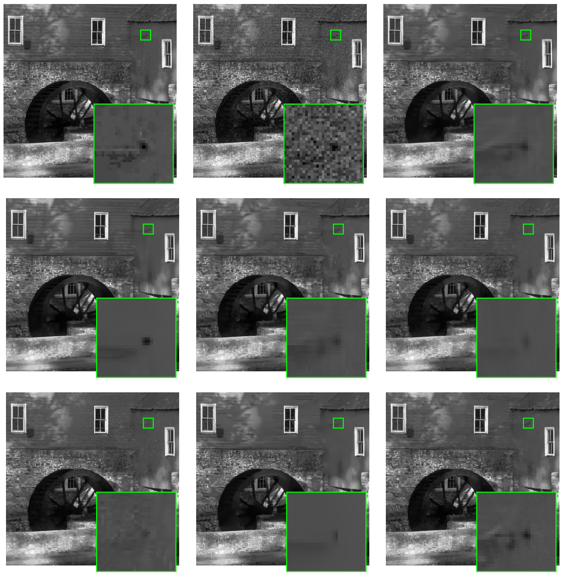

Fig. 9. Denoising comparison for 25th test image in Test40 dataset with moderate noise ($\sigma = 20$) corruption. *Top row*: original, noisy, the DnCNN [71] (SSIM = 0.781). *Middle row*: the DIDN [72] (SSIM = 0.786), the BM3D [17] (SSIM = 0.769), the NCSR [41] (SSIM = 0.767). *Bottom row*: the GHP [44] (SSIM = 0.768), the LINC [43] (SSIM = 0.762), and the proposed method (SSIM = 0.778).

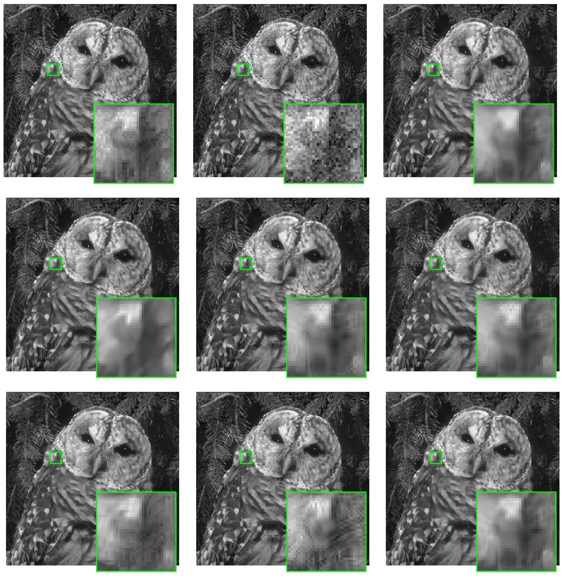

Fig. 10. Denoising comparison for 26th test image in Test40 dataset with moderate noise ($\sigma = 20$) corruption. *Top row*: original, noisy, the DnCNN [71] (SSIM = 0.795). *Middle row*: the DIDN [72] (SSIM = 0.789), the BM3D [17] (SSIM = 0.781), the NCSR [41] (SSIM = 0.777). *Bottom row*: the GHP [44] (SSIM = 0.784), the LINC [43] (SSIM = 0.776), and the proposed method (SSIM = 0.787).

C. Comparison With Other Related Techniques

Recently, Wang *et. al.* [31] have came up with a new perspective and equipped popular regularization functionals with structural similarity information in the nonlocal variational modeling. In details, they elegantly equipped NLH1 and NLTV regularization functionals with structural similarity information by incorporating it in the corresponding nonlocal gradients. Afterward, ADMM and CG based optimization routines were developed to solve corresponding problems (for detail see [31]). Additionally, authors had found that NLTV based variational model with CG solver i.e. SS-NLTV-CG [31] performed considerably better relative to other variants. Table-IV reports performance comparison of the proposed method(A) with available results of SS-NLTV-CG [31] scheme for various test images. Furthermore, one can observe that for low level corruption both techniques perform very closely. However, for moderate and high corruption proposed method achieves better performance. Again we see that the proposed method(A) is quite competitive.

D. Restoring Blur and Noisy Images

Deblurring problems are typically formulated as inverse problems and modeled as $\mathbf{I}_{ob} = H\mathbf{I}_{ac} + \mathcal{N}$, where $\mathbf{I}_{ac}$ is

TABLE III

THE AVERAGE PSNR (dB) AND SSIM VALUES OF DNCNN, DIDN, BM3D, NCSR, GHP, LINC AND PROPOSED METHOD(A) ARE REPORTED

$\sigma \longrightarrow$	Kodak24						Test20						Test40					
	5	10	15	20	50	75	5	10	15	20	50	75	5	10	15	20	50	75
DnCNN[71]	35.43	34.89	32.89	31.47	27.50	25.81	34.00	33.43	31.41	30.02	26.14	24.58	33.56	33.02	31.05	29.73	25.93	24.33
	0.911	0.922	0.889	0.857	0.737	0.675	0.892	0.907	0.866	0.829	0.689	0.617	0.894	0.905	0.866	0.831	0.693	0.617
DIDN [72]	38.83	35.17	33.16	31.82	27.96	26.27	37.12	33.45	31.50	30.17	26.34	24.66	36.13	32.76	31.00	29.81	26.25	24.69
	0.961	0.926	0.895	0.868	0.756	0.701	0.95	0.905	0.868	0.835	0.703	0.637	0.943	0.897	0.862	0.833	0.713	0.647
BM3D[17]	38.18	34.25	32.14	30.73	26.86	25.25	36.97	32.91	30.81	29.42	25.66	24.18	36.85	32.84	30.76	29.39	25.53	23.97
	0.957	0.911	0.871	0.836	0.709	0.647	0.952	0.896	0.849	0.809	0.662	0.594	0.952	0.901	0.857	0.820	0.670	0.596
NCSR[41]	38.35	34.47	32.36	30.92	26.84	25.16	37.12	33.12	31.02	29.59	25.64	24.12	36.94	32.96	30.90	29.50	25.51	23.91
	0.957	0.914	0.875	0.841	0.712	0.658	0.953	0.900	0.855	0.813	0.662	0.595	0.952	0.901	0.858	0.819	0.668	0.599
GHP[44]	37.90	34.16	31.92	30.86	26.18	23.14	36.92	32.96	30.71	29.54	24.95	22.35	36.37	32.59	30.48	29.36	24.78	22.14
	0.955	0.904	0.850	0.842	0.707	0.627	0.953	0.894	0.834	0.814	0.656	0.573	0.952	0.894	0.837	0.822	0.663	0.574
LINC[43]	37.78	33.94	31.96	30.61	26.90	25.37	36.54	32.45	30.46	29.13	25.51	24.00	36.40	32.31	30.32	29.02	25.38	23.84
	0.953	0.912	0.877	0.842	0.710	0.652	0.949	0.895	0.851	0.810	0.658	0.582	0.948	0.899	0.857	0.821	0.670	0.590
Method(A)	37.97	34.28	32.21	30.80	26.85	25.26	36.82	33.00	30.99	29.61	25.69	24.15	36.40	32.56	30.63	29.35	25.30	23.81
	0.957	0.913	0.875	0.847	0.712	0.658	0.952	0.900	0.856	0.818	0.667	0.594	0.950	0.900	0.860	0.825	0.667	0.598

TABLE IV

THE PSNR (dB) AND SSIM VALUES OF THE SS-NLTV-CG [31] SCHEME AND PROPOSED METHOD(A) ARE REPORTED

$\sigma \longrightarrow$	SS-NLTV-CG [31]				Method(A)			
	10	20	30	40	10	20	30	40
Parrot	35.03	31.93	29.77	27.77	35.70	32.33	30.11	28.80
	0.933	0.890	0.868	0.833	0.936	0.896	0.866	0.845
Barb	34.02	30.07	27.80	26.29	33.70	29.86	27.47	26.18
	0.942	0.885	0.821	0.757	0.931	0.869	0.800	0.757
C.man	33.47	30.07	27.99	26.32	33.95	30.31	28.29	26.32
	0.925	0.868	0.817	0.782	0.931	0.871	0.843	0.789
Hat	34.92	31.36	29.32	27.83	35.32	31.40	29.68	28.35
	0.931	0.867	0.819	0.783	0.940	0.865	0.839	0.790
Man	32.47	28.97	26.93	25.67	33.96	30.65	28.56	27.60
	0.905	0.814	0.738	0.679	0.908	0.835	0.759	0.730
Avg	33.98	30.48	28.40	26.78	34.53	30.91	28.82	27.45
	0.927	0.865	0.813	0.767	0.929	0.868	0.822	0.782

TABLE V

THE PSNR (dB) AND SSIM VALUES OF THE SS-NLTV-CG [31] SCHEME AND PROPOSED METHOD(A) FOR DEBLURRING ARE REPORTED

Blur →	SS-NLTV-CG [31]		Method(A)	
	G-Blur	M-Blur	G-Blur	M-Blur
Parrot	29.84	29.33	29.70	29.15
	0.886	0.882	0.884	0.879
Barb	25.81	25.68	24.96	24.67
	0.741	0.731	0.736	0.718
C.man	27.05	26.64	27.10	26.59
	0.833	0.820	0.839	0.829
Hat	30.12	29.70	29.98	29.60
	0.844	0.834	0.851	0.843
Man	28.06	27.59	29.62	29.30
	0.806	0.787	0.810	0.794
Avg	28.18	27.79	28.27	27.86
	0.822	0.810	0.824	0.813

the image to be estimated while $\mathbf{I}_{ob}$ is the observed degraded image. The degradation is caused due to linear blur operator H and added AWGN noise $\mathcal{N}$. Notice that if H is the identity operator then deblurring problem reduces to denoising problem. In this section, the proposed method is tested for non-blind image deblurring. Our simulation setting is similar to [31] i.e. Gaussian blur kernel (*fspecial('gaussian', 21, 1)*) and Moffat blurring function (*psfMoffat([21 21], 3, 5)*) were used. In detail, the reference image first convolves with the said blur kernel and subsequently Gaussian noise $\mathcal{N}$ (AWGN) of zero mean and standard deviation $\sigma = 10$ is added to generate final distorted image. Table-V lists out the PSNR and SSIM values of the proposed method and SS-NLTV-CG [31] scheme reported in [31]. We see that the proposed method achieves highly competitive performance and even better in some cases.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we developed the **pS** function that surrogates SSIM behavior as fidelity measure in model fitting and learning. Subsequently, an iterative procedure has been developed as denoiser that tries to reach mutual consent of both the smooth operation and the structural fidelity via the above learned model. Thanks to the power of SCS conic solver, this leads to a computationally efficient algorithm for learning and fitting tasks. Extensive experimental results have demonstrated that the proposed scheme can preserve the sharpness of the edges and suppress undesirable artifacts. This work clearly further indicates the importance of SSIM fidelity as extensively explained in [3]. In addition to Gaussian denoising, it would be interesting to study **pS** function with different noise distributions. It requires separate extensive study. For instance, what regularization should be taken and how to embed noise information in the corresponding scheme. This can be one of the possible research works in the future. The proposed framework has also been applied for deblurring task. However, it would be interesting to study **pS** function with other restoration tasks, for example, image inpainting, super-resolution, etc. The said applications will be further explored in the future work.

Appendix A
Proof of Theorem 1

It is obvious that for $\Phi' = \{\mathbf{x} \mid \mathbf{d}_y^T\mathbf{x} + c = 0\} \cup \Phi$ such that $\mathbf{dom}(\mathbf{pS})$ is equal to $\Phi = \{\mathbf{x} \mid \mathbf{d}_y^T\mathbf{x} + c > 0\}$, following $P1$ and $P2$ problems are equivalent

$$\begin{aligned} (P1) \quad & \text{minimize } \mathbf{pS}(\mathbf{y}, \mathbf{Dx}) + \sum\nolimits_{i=1}^{k} \lambda_i \Omega_i(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in \Phi \\ \equiv (P2) \quad & \text{minimize } \mathbf{pS}(\mathbf{y}, \mathbf{Dx}) + \sum\nolimits_{i=1}^{k} \lambda_i \Omega_i(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in \Phi' \end{aligned} \tag{25}$$

because the $P1$ equation is feasible by definition and above added set does not alter the optimal point. Subsequently, we rearrange the $P2$ equation in the epigraph using the definition of **pS** function

$$\begin{aligned} & \text{minimize } t + \sum\nolimits_{i=1}^{k} \lambda_i \Omega_i(\mathbf{x}) \\ & \text{subject to } \|\mathbf{y} - \mathbf{Dx}\|^2 \le tw \\ & \qquad \mathbf{d}_y^T\mathbf{x} + c = w, \; w \ge 0 \end{aligned} \tag{26}$$

Note that by definition t and w are non-negative and therefore, $tw = (t/2 + w/2)^2 - (t/2 - w/2)^2$. Suppose $t/2 + w/2 = p$ and $t/2 - w/2 = q$, then we have $w = q - p$, $t = q + p$ and

$$\|\mathbf{y} - \mathbf{Dx}\|^2 \le p^2 - q^2 \equiv \left\| \begin{matrix} \mathbf{y} - \mathbf{Dx} \\ q \end{matrix} \right\| \le p \tag{27}$$

Finally, substituting t and w and using (27), we establish SOCP equivalent given in (9).

In order to prove (10), suppose $(\mathbf{x}^*)$ and $(\mathbf{z}^*, w^*)$ are optimal points for the left and the right side of the equivalence in (10). The sets $\mathbf{X}$ and $\mathbf{Z}$, as defined below in the Eq. (28) for convex functions Γ_i; $1 \cdots k$, are convex. Since, $w\,\Gamma_i(\mathbf{z}/w)$ is perspective function of Γ_i convex function, hence $w\,\Gamma_i(\mathbf{z}/w)$ is also convex [45].

$$\begin{aligned} \mathbf{X} &= \{\mathbf{x} \mid \mathbf{x} \in \Phi, \; \Gamma_i(\mathbf{x}) \le L_i, \; i = 1 \cdots L\} \\ \mathbf{Z} &= \{(\mathbf{z}, w) \mid w\Gamma_i(\mathbf{z}/w) \le 0, \;\; i = 1 \cdots k, \\ & \qquad \mathbf{d}_y^T\mathbf{z} + wc = 1, \; w > 0\} \end{aligned} \tag{28}$$

Further, if any $\mathbf{x} \in \mathbf{X}$ obeys relation $\mathbf{z} = w\mathbf{x}$ such that $w = \frac{1}{\mathbf{d}_y^T\mathbf{x} + c} > 0$, then for $i = 1 \cdots k$ following holds

$$\Gamma_i(\mathbf{x}) \le 0 \Rightarrow w\,\Gamma_i(\mathbf{z}/w) \le 0 \tag{29}$$

$$\mathbf{d}_y^T\mathbf{z} + wc = 1 \tag{30}$$

$$\mathbf{pS}(\mathbf{y}, \mathbf{Dx}) = w\|\mathbf{y} - \mathbf{Dx}\|^2 = \frac{1}{w}\|w\mathbf{y} - \mathbf{Dz}\|^2 \tag{31}$$

Subsequently, any feasible point $\mathbf{x}$ in $\mathbf{X}$ translates into feasible point $(\mathbf{z}, w)$ in $\mathbf{Z}$. Furthermore, Eq. (31) implies that objective values are identical for the left and the right side of the equivalence in (10). Consequently, we have

$$\frac{1}{w^*}\|w^*\mathbf{y} - \mathbf{Dz}^*\|^2 \le \mathbf{pS}(\mathbf{y}, \mathbf{Dx}^*) \tag{32}$$

Conversely, let any $(\mathbf{z}, w)$ in $\mathbf{Z}$ obey relation $\mathbf{x} = \mathbf{z}/w$ then for $i = 1 \cdots k$ following holds,

$$\mathbf{d}_y^T\mathbf{z} + wc = 1 \Rightarrow w = \frac{1}{\mathbf{d}_y^T\mathbf{x} + c} > 0 \Rightarrow \mathbf{x} \in \Phi \tag{33}$$

$$w\,\Gamma_i(\mathbf{z}/w) \le 0 \Rightarrow \Gamma_i(\mathbf{x}) \le 0 \quad \text{for } i = 1 \cdots k \tag{34}$$

$$\frac{1}{w}\,\|w\mathbf{y} - \mathbf{Dz}\|^2 = w\,\|\mathbf{y} - \mathbf{Dz}/w\|^2 = \mathbf{pS}(\mathbf{y}, \mathbf{Dx}) \tag{35}$$

We conclude, any feasible point in $\mathbf{Z}$ mapped through relation $\mathbf{x} = \mathbf{z}/w$ is feasible to set $\mathbf{X}$. Furthermore, by relation (35) the objective values of both left and right side of Eq. (10) are identical. Therefore,

$$\mathbf{pS}(\mathbf{y}, \mathbf{Dx}^*) \le \frac{1}{w^*}\|w^*\mathbf{y} - \mathbf{Dz}^*\|^2 \tag{36}$$

Together with (32) and (36) we conclude the equivalence.

Appendix B
Simplification for Conic Solver

Consider optimization problem related to ℓ_1 norm regularized model with **pS** function recorded in Eq. (22). Suppose, we construct $\mathbf{x}_+$ and $\mathbf{x}_-$ from $\mathbf{x}$ in Eq. (22) as follows

$$x_{-i} = \begin{cases} |x_i| & x_i < 0 \\ 0 & else \end{cases}$$

$$x_{+i} = \begin{cases} x_i & x_i > 0 \\ 0 & else \end{cases}$$

Then, the following relations hold and are used to make Eq. (22) smooth for SCS conic solver [55].

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_+ - \mathbf{x}_- \\ \mathbf{0} &\preceq \mathbf{x}_+, \; \mathbf{0} \preceq \mathbf{x}_- \\ \|\mathbf{x}\|_1 &= \mathbf{1}^T(\mathbf{x}_+ + \mathbf{x}_-) \end{aligned} \tag{37}$$

Let $\mathbf{y} - \mathbf{Dx} = \mathbf{v}$, then smooth version of Eq. (22) is

$$\begin{aligned} & \text{minimize } p + q + \lambda\mathbf{1}^T(\mathbf{x}_+ + \mathbf{x}_-) \\ & \text{subject to } \mathbf{Dx}_+ - \mathbf{Dx}_- + \mathbf{v} - \mathbf{y} = 0 \\ & \qquad w = \mathbf{d}_y^T(\mathbf{x}_+ - \mathbf{x}_-) + c, q - p \ge 0 \\ & \qquad \left\| \begin{matrix} \mathbf{v} \\ q \end{matrix} \right\| \le p \;, \mathbf{x}_+ \succeq \mathbf{0}, \mathbf{x}_- \succeq \mathbf{0} \end{aligned} \tag{38}$$

Clearly, optimization variables $(\mathbf{x}_+, \mathbf{x}_-)$ in Eq. (38) decode solution to original problem (22) by using relation $\mathbf{x}_+ - \mathbf{x}_- = \mathbf{x}$. Additionally from theorem 1 and relation $\mathbf{d}_y = \mathbf{D}^T\mathbf{y}$ implies that for each new test patch $\mathbf{y}$, SCS solver requires to compute sparse permuted LDL factorization [56] for linear constraint in Eq. (38). Subsequently, execution of initialization step for each test vector in SCS solver is inevitable. For details interested readers may refer to Brendan *et. al.* [55]. In order to overcome this problem in caching process, we first note definition of matrix $\mathbf{M}$ from [55] as follows

$$\mathbf{M} = \begin{bmatrix} \mathbf{I} & \mathbf{A}^T \\ -\mathbf{A} & \mathbf{I} \end{bmatrix} \tag{39}$$

where, $\mathbf{A}$ is constituted from the linear part of Eq. (38). By rearranging this linear part, one can represent linear constrained related to $\mathbf{d}_y^T$ as first row of matrix $\mathbf{A}^T$. Subsequently, matrix $\mathbf{M}$ can be represented as

$$\mathbf{M} = \begin{bmatrix} 1 & \mathbf{a}^{\mathbf{T}}(t) \\ -\mathbf{a}(t) & \mathbf{M}' \end{bmatrix} \tag{40}$$

where,

$$\mathbf{a}(t) = \begin{bmatrix} \mathbf{d}_y^T \ \mathbf{d}_y^T \ -1 \ \ \mathbf{0} \end{bmatrix} \quad \& \quad \mathbf{M}' = \begin{bmatrix} \mathbf{I}_{\mathbf{(N-1)}} & \mathbf{A}^{\mathbf{T}}_{\mathbf{(N-1)}} \\ -\mathbf{A}_{\mathbf{(N-1)}} & \mathbf{I} \end{bmatrix}$$

Note that vector $\mathbf{a}$ is represented with argument t to emphasis its varying nature with different observations $\mathbf{y}$. In order to deduce new global caching factorization for Eq. (38), we start with borrowing Eq.(27) from [55],

$$\begin{bmatrix} \mathbf{I} & \mathbf{A}^{\mathbf{T}} \\ -\mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{A}^{\mathbf{T}} \\ \mathbf{A} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{bmatrix} = \begin{bmatrix} \mathbf{w}_x \\ -\mathbf{w}_y \end{bmatrix} \tag{41}$$

Considering Eq. (39) and (40) and simplifying above, we get

$$\begin{bmatrix} 1 & \mathbf{a}^{\mathbf{T}}(t) \\ \mathbf{a}(t) & \mathbf{M}' \end{bmatrix} \begin{bmatrix} z_0 \\ \begin{bmatrix} \mathbf{z}_{x\backslash 0} \\ \mathbf{z}_y \end{bmatrix} \end{bmatrix} = \begin{bmatrix} w_0 \\ \begin{bmatrix} \mathbf{w}_{x\backslash 0} \\ -\mathbf{w}_y \end{bmatrix} \end{bmatrix} \tag{42}$$

Applying *Sherman-Morrison-Woodbury* formula and *Schur* complement [74], we have following equation set

$$\begin{bmatrix} \mathbf{z}_{x\backslash 0} \\ \mathbf{z}_y \end{bmatrix} = \mathbf{S}^{\mathbf{-1}} \left(\begin{bmatrix} \mathbf{w}_{x\backslash 0} \\ -\mathbf{w}_y \end{bmatrix} - w_0 \mathbf{a}(t) \right)$$

$$z_0 = w_0 - \mathbf{a}^{\mathbf{T}} \begin{bmatrix} \mathbf{z}_{x\backslash 0} \\ \mathbf{z}_y \end{bmatrix}$$

$$\mathbf{S}^{-1} = (\mathbf{M}' - \mathbf{a}\mathbf{a}^{\mathbf{T}})^{-1} = \mathbf{M}'^{-1} + \frac{\mathbf{M}'^{-1}\mathbf{a}\mathbf{a}^{\mathbf{T}}\mathbf{M}'^{-1}}{1 - \mathbf{a}^{\mathbf{T}}\mathbf{M}'^{-1}\mathbf{a}^{\mathbf{T}}} \tag{43}$$

The first two equations in the above set are used to obtain unknown coefficients of Eq. (42) using the last equation in the set.

Conclusively, global caching factorization is still achievable but with respect to $\mathbf{M}'$ rather than $\mathbf{M}$. In other words, for given observation computationally non-iterative quantities ($e.g.$ $\mathbf{M}^{\mathbf{-1}}\mathbf{h}$) and iterative one ($e.g.$ $Eq.$ (41)) in [55] can be made cacheable with respect to $\mathbf{M}'$ for different test vector $\mathbf{y}$. Additionally, $\mathbf{M}$ is computed for different test vector using Eq. (43). In this way, we can avoid initialization process of the solver for each new observation for improving computational efficiency.

Acknowledgment

The authors would like to thank the reviewers for their patiently reading and helpful comments which improves the quality of this manuscript a lot.

References

[1] Z. Wang and A. C. Bovik, "Modern image quality assessment," *Synth. Lectures Image, Video, Multimedia Process.*, vol. 2, no. 1, pp. 1–156, Jan. 2006.

[2] B. Girod, *Digital Images and Human Vision*. A. B. Watson, Ed. Cambridge, MA, USA: MIT Press, 1993, pp. 207–220. [Online]. Available: http://dl.acm.org/citation.cfm?id=197765.197784

[3] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[4] Z. Wang, "Applications of objective image quality assessment methods [Applications Corner]," *IEEE Signal Process. Mag.*, vol. 28, no. 6, pp. 137–142, Nov. 2011.

[5] D. M. Chandler, "Seven challenges in image quality assessment: Past, present, and future research," *ISRN Signal Process.*, vol. 2013, pp. 1–53, Feb. 2013.

[6] Z. Wang and A. C. Bovik, "Reduced- and no-reference image quality assessment," *IEEE Signal Process. Mag.*, vol. 28, no. 6, pp. 29–40, Nov. 2011.

[7] W. Lin and C.-C. Jay Kuo, "Perceptual visual quality metrics: A survey," *J. Vis. Commun. Image Represent.*, vol. 22, no. 4, pp. 297–312, May 2011.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[9] D. M. Chandler and S. S. Hemami, "VSNR: A wavelet-based visual signal-to-noise ratio for natural images," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2284–2298, Sep. 2007.

[10] H.-W. Chang, H. Yang, Y. Gan, and M.-H. Wang, "Sparse feature fidelity for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4007–4018, Oct. 2013.

[11] A. Saha and Q. M. J. Wu, "Full-reference image quality assessment by combining global and local distortion measures," *Signal Process.*, vol. 128, pp. 186–197, Nov. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016516841 6300196

[12] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3350–3364, Dec. 2011.

[13] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1733–1740.

[14] A. Rehman and Z. Wang, "Reduced-reference image quality assessment by structural similarity estimation," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3378–3389, Aug. 2012.

[15] Y. Zhang, T. D. Phan, and D. M. Chandler, "Reduced-reference image quality assessment based on distortion families of local perceived sharpness," *Signal Process., Image Commun.*, vol. 55, pp. 130–145, Jul. 2017.

[16] M. Hasan and M. R. El-Sakka, "Improved BM3D image denoising using SSIM-optimized Wiener filter," *EURASIP J. Image Video Process.*, vol. 2018, no. 1, p. 25, Dec. 2018.

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[18] A. Rehman, M. Rostami, Z. Wang, D. Brunet, and E. R. Vrscay, "SSIM-inspired image restoration using sparse representation," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, p. 16, Dec. 2012. [Online]. Available: http://asp.eurasipjournals.com/content/2012/1/16

[19] A. Rehman and Z. Wang, "SSIM-based non-local means image denoising," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 217–220.

[20] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey, "Complex wavelet structural similarity: A new image similarity index," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2385–2401, Nov. 2009.

[21] Y. Gao, A. Rehman, and Z. Wang, "CW-SSIM based image classification," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1249–1252.

[22] Z. Wang and X. Shang, "Spatial pooling strategies for perceptual image quality assessment," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 2945–2948.

[23] H. Chang and J. Zhang, "New metrics for clutter affecting human target acquisition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 361–368, Jan. 2006.

[24] G. Piella and H. Heijmans, "A new quality metric for image fusion," in *Proc. Int. Conf. Image Process.*, vol. 3, Sep. 2003, pp. III–173.

[25] A. A. Dovganich and A. S. Krylov, "A nonlocal image denoising algorithm using the structural similarity metric," *Program. Comput. Softw.*, vol. 45, no. 4, pp. 141–146, Jul. 2019.

[26] W. Yu, "Practical anti-vignetting methods for digital cameras," *IEEE Trans. Consum. Electron.*, vol. 50, no. 4, pp. 975–983, Nov. 2004.

[27] S. S. Channappayya, A. C. Bovik, C. Caramanis, and R. W. Heath, "Design of linear equalizers optimized for the structural similarity index," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 857–872, Jun. 2008.

[28] D. Otero and E. R. Vrscay, "Unconstrained structural similarity-based optimization," in *Proc. Int. Conf. Image Anal. Recognit.* Springer, 2014, pp. 167–176.

[29] D. Otero, D. La Torre, and E. R. Vrscay, "Structural similarity-based optimization problems with L^1 -regularization: Smoothing using mollifiers," in *Proc. Int. Conf. Image Anal. Recognit.* Springer, 2015, pp. 33–42.

[30] D. Otero, D. L. Torre, O. V. Michailovich, and E. R. Vrscay, "Alternate direction method of multipliers for unconstrained structural similarity-based optimization," in *Image Analysis and Recognition*, A. Campilho, F. Karray, and B. ter Haar Romeny, Eds. Cham, Switzerland: Springer, 2018, pp. 20–29.

[31] W. Wang, F. Li, and M. K. Ng, "Structural similarity-based nonlocal variational models for image restoration," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4260–4272, Sep. 2019.

[32] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.

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

[34] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1488–1499, Apr. 2012.

[35] T. Norfolk, *When Does a Metric Generate Convex Balls*. Akron, OH, USA:, 1991.

[36] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1033–1040.

[37] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

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

[39] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[40] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2481–2499, May 2012.

[41] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[42] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3501–3508.

[43] M. Niknejad, H. Rabbani, and M. Babaie-Zadeh, "Image restoration using Gaussian mixture models with spatially constrained patch clustering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3624–3636, Nov. 2015.

[44] W. Zuo, L. Zhang, C. Song, D. Zhang, and H. Gao, "Gradient histogram estimation and preservation for texture enhanced image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2459–2472, Jun. 2014.

[45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[46] P. Siarry, *Optimisation in Signal and Image Processing*, vol. 46. Hoboken, NJ, USA: Wiley, 2010.

[47] T. F. Chan and J. J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Philadelphia, PA, USA: SIAM, 2005.

[48] M. Zibulevsky and M. Elad, "L1-L2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.

[49] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996.

[50] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.

[51] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[52] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. New York, NY, USA: Academic, 2014.

[53] F. Bach, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, 2011.

[54] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[55] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," 2013, *arXiv:1312.3039*. [Online]. Available: http://arxiv.org/abs/1312.3039

[56] T. Davis, *Direct Methods for Sparse Linear Systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006.

[57] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[58] Y.-Q. Wang and J.-M. Morel, "SURE guided Gaussian mixture image denoising," *SIAM J. Imag. Sci.*, vol. 6, no. 2, pp. 999–1034, Jan. 2013.

[59] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, Nov. 1981.

[60] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, vol. 15, 1988, p. 50.

[61] J. Franklin, "The elements of statistical learning: Data mining, inference and prediction," *Math. Intelligencer*, vol. 27, no. 2, pp. 83–85, Mar. 2005.

[62] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[63] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2209–2216.

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

[65] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar. 2002.

[66] 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*. 2009.

[67] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[68] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer, 2010.

[69] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[70] Z. Wang and E. P. Simoncelli, "Maximum differentiation (MAD) competition: A methodology for comparing computational models of perceptual quantities," *J. Vis.*, vol. 8, no. 12, p. 8, Sep. 2008, doi: 10.1167/8.12.8.

[71] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[72] S. Yu, B. Park, and J. Jeong, "Deep iterative down-up CNN for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2095–2103.

[73] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Springer, 2015, pp. 234–241.

[74] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2012.