



Adaptive direction-guided structure tensor total variation

Ezgi Demircan-Tureyen^{a,b,*}, Mustafa E. Kamasak^b

^a Department of Computer Engineering, Istanbul Kultur University, Istanbul 34156, Turkey

^b Department of Computer Engineering, Istanbul Technical University, Istanbul 34390, Turkey



ARTICLE INFO

Keywords:

Variational models
Image denoising
Directional total variation
Structure tensor
Orientation field estimation
Inverse problems

ABSTRACT

Direction-guided structure tensor total variation (DSTV) is a recently proposed regularization term that aims at increasing the sensitivity of the structure tensor total variation (STV) to the changes towards a predetermined direction. Despite of the plausible results obtained on the uni-directional images, the DSTV model is not applicable to the arbitrary (multi-directional and/or partly nondirectional) images. In this study, we build a two-stage denoising framework that brings adaptivity to the DSTV based denoising. We design a DSTV-like alternative to STV, which encodes the first-order information within a local neighborhood under the guidance of spatially varying directional descriptors (i.e., orientation and the dose of anisotropy). In order to estimate those descriptors, we propose an efficient preprocessor that captures the local geometry based on the structure tensor. Through the extensive experiments, we demonstrate how beneficial the involvement of the directional information in STV is, by comparing the proposed method with the state-of-the-art analysis-based denoising models, both in terms of quality and computational efficiency.

1. Introduction

Restoration of an image from its noisy linear transformations is an active field of research in imaging science. This field involves a variety of linear inverse problems; such as denoising, deblurring, inpainting, and super-resolution. Among them, the denoising problem is extensively studied; not only due to having a wide spectrum of application fields, but also because it serves as a building block for the other, more complex problems. The fundamental drawback of denoising (and the others) is ill-posedness. The knowledge deduced from the observed image is often insufficient and requires to take some prior information on the latent image into account. When such a problem is handled from a variational perspective, its reinterpretation in the form of an optimization problem involves a regularization term (i.e., regularizer), which is responsible for encoding that prior information.

A lot of effort has been dedicated to design good regularizers, which can characterize the unknown image well and in an efficiently solvable way. Rudin–Osher–Fatemi’s total variation (TV) [1] has been considered to be a *good* regularizer, due to its well-characterization of the images (as piecewise constant signals up to the edge regions), and its convex definition. Inevitably, TV’s piecewise constancy (PWC) assumption does not always hold. The real-world images may be piecewise smooth rather than constant, and when this is the case, TV yields artificial edges, known as staircase artifacts [2]. As being the initiator of the edge-preserving regularization notion, TV has given rise to a diverse range of TV inspired regularizers. Many of these regularizers

extend TV to stretch its PWC assumption, by bring anisotropy to the TV-term to better catch angled boundaries (e.g., [3–5]), and/or by incorporating the higher order information to promote piecewise smoothness (e.g., total generalized variation — TGV [6], and the other notable works [2,7,8]).

There are also TV extensions that steer anisotropy by a desired direction. Directional total variation (DTV) [9] is such a regularizer that aims at penalizing the local changes along a single dominant direction more than the others. For this purpose, it employs rotated and scaled gradients. Even though it performs well on the uni-directional images, DTV model fails to restore arbitrary, real-world images. It is applicable only when the image to be restored exhibits a single directional dominance. In [10], the authors suggested an edge-adaptive DTV (EADTV) to make DTV handle several directions. They proposed a two-stage approach: (1) extracting the tangential directions of the edges, (2) feeding them into the EADTV model. However, the noise heavily affects the edges and mostly the wrong directions are penalized. A very recent work [11] proposed an extended DTV model that follows a similar two-stage approach. Distinctively, to extract directions, they employed a more sophisticated orientation field estimation (OFE) method used for fingerprint identification in [12]. They also made use of the mixed $\ell_p - \ell_2$ norm, where $p \in (0, 1)$, while $p = 1$ in DTV.

The direction-guided regularizers mentioned above requires a preprocessor that provides the directions to them. So, the erroneous outputs of the preprocessor mislead the succeeding inversion process.

* Corresponding author.

E-mail addresses: e.demircan@iku.edu.tr (E. Demircan-Tureyen), kamasak@itu.edu.tr (M.E. Kamasak).

In [13], anisotropic TV filtering (ATV) was proposed as a combination of TV and anisotropic diffusion. It embodies the *diffusion tensor* designed according to the *structure tensor*. It intrinsically depends on the underlying image, rather than the observation. The downsides of ATV are its non-convexity and its heuristic design. The objective function mostly has multiple local or global minimizers, and it may result in a suboptimal solution. The recent structure tensor total variation (STV) [14], which forms the basis of this study in collaboration with DTV [9], proposed a convex objective by following the same idea behind ATV. It suggests to penalize the rooted eigenvalues of the structure tensor. In this way, it can capture the variations within a local neighborhood in contrast to TV and its localized extensions. In [14], its superiority has been shown over vectorial TV (VTV) [15], which extends TV to handle vector-valued images, and TGV [6].

Above mentioned regularizers are all local, except ATV and STV, which are classified as semi-local. There are also variational models that benefit from the non-local (NL) self similarity by adopting the idea of [16] (e.g., NLTV [17], NLSTV [18]). They usually perform better than their local or semi-local counterparts. Moreover, the regularizers mentioned here are all categorized as analysis-based priors, where the regularization directly runs on the image. Contrary to this, there are also synthesis-based priors that perform the regularization in a sparsifying transform domain (e.g., wavelet [19,20], shearlets [21]). Additionally, there is a recent paper [22] that proposes a semi-local regularizer along with its non-local version. Their prior depends on a new paradigm: it employs sparsity of the corners rather than the edges. They showed that their method (Noble's Corner Detector based Regularization — NCDR) is superior to STV (and NLSTV for the non-local versions) in terms of restoration quality; apart from the fact that it is non-convex, works heuristically and slow.

1.1. Contributions

Our intention is proposing a variational denoising framework which produces physically satisfying and statistically meaningful results that are superior to the state-of-the-art analysis-based variational methods. In this respect, our contributions are listed below:

1. We design an adaptive direction-guided penalty term by extending STV.
2. We propose a preprocessing method that estimates the parameters required by our regularizer.
3. We develop an efficient algorithm based on the proximal map of the proposed regularizer in order to solve the introduced convex optimization problem.
4. We assessed the performance of our framework by comparing the results with those obtained by TV, EADTV [9,10], STV [14], and NCDR [22]. We also include our preliminary work DSTV [23] to the experiments in order to clarify the contribution of the adaptivity.

Note that, we only focus on extending the semi-local STV to semi-local adaptive DSTV; thus, we will not consider designing a non-local counterpart of our regularizer, which can straightforwardly be accomplished by exploiting NLSTV [18]. It would surely boost the restoration quality.

2. Problem formulation

We deal with the recovery of an underlying function $f : \Omega \mapsto \mathbb{R}^C$ from an observation $g : \Omega \mapsto \mathbb{R}^C$, where the forward corruption is modeled as: $g(x) = f(x) + \eta(x)$, with $\eta : \Omega \mapsto \mathbb{R}^C$ referring to the additive noise. Here, C denotes the number of channels, which is more than one for the vector-valued images, and $x \in \Omega$, where the domain $\Omega \subseteq \mathbb{R}^2$ is a 2-dimensional image space. The noise at each pixel is modeled as an *i.i.d.* random variable that follows Gaussian distribution with $\eta(x) \sim \mathcal{N}(0, \sigma_\eta^2)$.

The variational models combine a data fidelity term with a regularization term $R(f)$ in the form of an optimization problem. For the Gaussian denoising, the fidelity term is nothing but the least squares fitting term to stick by the observed data, thus, one has the objective function of the form:

$$E(f) = \frac{1}{2} \|g - f\|_2^2 + \tau R(f) \quad (1)$$

to be minimized with respect to f . Here, τ is the regularization parameter that is used to tune the contribution of $R(f)$ to the cost. Note that, there are studies in the literature that makes τ adaptive to give rise to spatially dependent regularization (e.g., [24–26]).

2.1. Notation

Throughout the paper, both of the scalar-valued grayscale images ($C = 1$) and the vector-valued RGB images ($C = 3$) will be our concern. We will only focus on the discrete domain, where f is a discrete-space function, thus we switch the notation from $f(x)$ to $f[i]$, for $i \in \{1, 2, \dots, N\}$ referring to the index of a certain pixel, where N is the number of pixels in a single channel. Note that, we treat the images as being lexicographically stacked into vectors ($f \in \mathbb{R}^{NC}$).

$\|X\|_{p,q}$ stands for either the mixed $\ell_p - \ell_q$ norm defined as $(\sum_{i=1}^N \|X[i]\|_q^p)^{1/p}$, or the mixed $\ell_p - S_q$ norm when X is a matrix. Here S_q refers to the Schatten norm of order q .¹ The symbols \odot and \circ are used to denote entrywise product and entrywise power, respectively; while $\langle \cdot \rangle$ refers to the inner product. Furthermore, the vectors/matrices are denoted by using lowercase/uppercase bold letters herein after.

3. Background

The analysis-based regularizers can generically be defined as follows:

$$R(\mathbf{f}) = \|\phi((\Gamma\mathbf{f})[i])\|_p \quad (2)$$

TV seminorm is nothing but the ℓ_1 norm of the gradient magnitude, i.e., according to Eq. (2), $\Gamma = \nabla$, $\phi = \|\cdot\|_2$, and $p = 1$, where ∇ denotes the gradient operator. As mentioned in Section 1, many alternatives to this native isotropic TV have been proposed, e.g., $\phi = \|\cdot\|_1$ in [3], $\phi = \|\cdot\|_q$ and $q \in (0, 1)$ in [4], $\phi = w_1 \|\cdot\|_1 + w_2 \|\cdot\|_2$ in [5], where $\Gamma = \nabla$ and $p = 1$ were kept for all.

3.1. Directional total variation (DTV)

DTV [9] is a local analysis-based regularizer that simply changes the TV's rotation invariant ∇ in Eq. (2) with $\Gamma = \Lambda_\alpha \mathbf{R}_{-\theta} \nabla$, where $\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ and $\Lambda_\alpha = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix}$ with $\theta \in [0, \pi)$ corresponding to the dominant direction $(\cos \theta, \sin \theta)$, and α that is used to tune the dose of the penalization. In this way, it penalizes the magnitudes of the scaled and rotated gradients, towards a predetermined direction. As mentioned earlier, DTV can only be applied to the uni-directional images. EADTV [10] is its adaptive extension that suggests to use spatially varying angles $\theta[i]$. Even it is trivial to estimate a scalar θ from an observed uni-directional image, when θ represents an *orientation map*, it becomes a challenging problem. EADTV simply computes each $\theta[i]$ such that,

$$(\cos \theta[i], \sin \theta[i]) \perp (\nabla \mathbf{g}_\sigma)[i] \quad (3)$$

where the subscript σ indicates that the image \mathbf{g} is pre-smoothed by a Gaussian filter of standard deviation σ . However, the gradient is not a convenient descriptor under extensive amount of noise. The estimation of the gradient is mostly misguided, and since DTV is sensitive along

¹ Schatten norm of a matrix \mathbf{X} is ℓ_q norm of the vector $\sigma(\mathbf{X})$, where σ returns the singular values, i.e., $\|\mathbf{X}\|_{S_q} = \|\sigma(\mathbf{X})\|_q$ [27].

the angles $\theta[i]$, the structure is destroyed by the incorrect estimates. Moreover, rippled values of the angles cause unstable output.

By recalling Eq. (2), the recent paper [11] plugs $\Gamma = \Lambda_\alpha(\mathbf{g})R_{-\theta(\mathbf{g})}\nabla$, $\phi = \|\cdot\|_2$, and $p \in (0, 1)$ where Λ_α and θ are both functions of \mathbf{g} . Except the preference of the outer norm, this definition differs from the EADTV in the sense that it better recognizes the local edge orientations due to the more complex OFE method that it employs, and uses an adaptive stretching matrix weighted according to the first-order local edge information.

3.2. Structure tensor total variation (STV)

The *structure tensor* of an image \mathbf{f} at a point i is a 2×2 symmetric positive semi-definite (PSD) matrix of the form:

$$(S_K \mathbf{f})[i] = \mathbf{K}_\delta * ((J\mathbf{f})[i](J\mathbf{f})[i]^T) \quad (4)$$

where \mathbf{K}_δ is a Gaussian kernel of standard deviation δ centered at the i th pixel, and J denotes the Jacobian operator that extends the gradient for vector-valued images, i.e.,

$$(J\mathbf{f})[i] = [(\nabla \mathbf{f}^1)[i] \ (\nabla \mathbf{f}^2)[i] \ \dots \ (\nabla \mathbf{f}^C)[i]] \quad (5)$$

The superscripted \mathbf{f} 's are denoting the channels, and for scalar-valued images, $J(\cdot) = \nabla(\cdot)$. This semi-local descriptor can summarize all the gradients within the patch supported by \mathbf{K}_δ , thus provides a better way of characterizing the variations, when compared to the local differential operators. The eigenvectors of it correspond to the directions of maximum and minimum vectorial variations, and the rooted eigenvalues serve as a measure of variations towards the respective directions. At this point, the STV prior [14] presents an innovative way of describing the total amount of variations. It replaces the TV's ∇ in Eq. (2) with $\Gamma = \sqrt{\lambda(\cdot)}$, where $(\lambda \mathbf{f})[i]$ is a 2D vector composed of the maximum $\lambda^+(\cdot)$ and the minimum $\lambda^-(\cdot)$ eigenvalues of the structure tensor at the point i , i.e., $(\lambda \mathbf{f})[i] = [\lambda^+((S_K \mathbf{f})[i]), \lambda^-((S_K \mathbf{f})[i])]^T$. STV also considers a more general case by using $\phi = \|\cdot\|_q$ ($q > 1$). Since this design of STV is nonlinear and Eq. (4) involves a convolution kernel, the authors of [14] reformulated $(S_K \mathbf{f})[i]$ in terms of another operator that they named as *patch-based Jacobian* $J_K : \mathbb{R}^{N \times C} \mapsto \mathbb{R}^{N \times LC \times 2}$, which embodies the convolution kernel of size L , and enables to express the STV functional such that it can be decomposed into linear functionals. $(J_K \mathbf{f})[i]$ is defined as:

$$(J_K \mathbf{f})[i] = [(\tilde{\nabla} \mathbf{f}^1)[i], (\tilde{\nabla} \mathbf{f}^2)[i] \ \dots \ (\tilde{\nabla} \mathbf{f}^C)[i]]^T \quad (6)$$

where $(\tilde{\nabla} \mathbf{f}^c)[i] = [(\mathcal{T}_1 \nabla \mathbf{f}^c)[i], (\mathcal{T}_2 \nabla \mathbf{f}^c)[i], \dots, (\mathcal{T}_L \nabla \mathbf{f}^c)[i]]$. Each l th entity applies shifting and weighting on the gradient as $(\mathcal{T}_l \nabla \mathbf{f}^c)[i] = \sqrt{K_\sigma} [p_l](\nabla \mathbf{f}^c)[x_i - p_l]$, where x_i denotes the actual 2D coordinates of the i th pixel, and for $L = (2L_K + 1)^2$, $p_l \in \{-L_K, \dots, L_K\}^2$ is the shift amount. That way, the structure tensor in terms of the patch-based Jacobian is:

$$(S_K \mathbf{f})[i] = (J_K \mathbf{f})[i](J_K \mathbf{f})[i]^T \quad (7)$$

Now, the rooted eigenvalues of $(S_K \mathbf{f})[i]$ coincide with the singular values of $(J_K \mathbf{f})[i]$, and by employing S_q matrix norm, STV is redefined as:

$$STV(\mathbf{f}) = \sum_{i=1}^N \|(J_K \mathbf{f})[i]\|_{S_q} \quad (8)$$

This redefinition allows one to find an efficient numerical solution to STV regularized problems, by using convex optimization tools.

3.3. DTV versus STV

Before diving into the proposed method, let us first point out the relation between the DTV and STV measures. By expanding the DTV measure, one can draw the following conclusion:

$$DTV(\mathbf{f}) \triangleq \|\mathbf{A}_\alpha \mathbf{R}_{-\theta} \nabla \mathbf{f}\|_{1,2} = \left\| \begin{bmatrix} \alpha D_\theta \mathbf{f} \\ D_{\theta^\perp} \mathbf{f} \end{bmatrix} \right\|_{1,2} \quad (9)$$

where D_θ denotes the *directional derivative* of \mathbf{f} in the direction characterized by θ , and $\theta^\perp = \theta \pm \pi/2$. Eq. (9) shows that DTV actually works by increasing the dose of penalization (with the weight α) in the direction that it presumes the variation is minimum. The penalty dose in the direction of maximum variation, which is orthogonal to that of the minimum, is only determined by the regularization parameter τ . This idea of penalizing the maximum and minimum directional variations coincides with the STV. However, instead of making assumptions, STV codifies the maximum and the minimum directional variations with the maximum and the minimum eigenvectors and the square roots of the corresponding eigenvalues ($\sqrt{\lambda^+}$ and $\sqrt{\lambda^-}$) of the structure tensor. In other words, it deduces the directional variation from a summary of all derivatives within a local neighborhood.

$$STV(\mathbf{f}) \triangleq \left\| \begin{bmatrix} \sqrt{\lambda^+(S_K \mathbf{f})} \\ \sqrt{\lambda^-(S_K \mathbf{f})} \end{bmatrix} \right\|_{1,q} \quad (10)$$

From this point of view, the question that motivated us to design a direction-guided STV was: Can STV more accurately deduce the directional variation if the summarized derivatives are directional in a pre-determined direction?

4. Proposed method

4.1. Direction-guided STV (DSTV)

While it has been shown that the STV produces the state-of-the-art results among the other analysis-based local regularizers; under excessive amount of noise, it may not distinguish the edges and may smooth out them. At this point, the DSTV [23] comes into play. It aims at incorporating the prior knowledge on the underlying image's local geometry into STV. Under the inspiration of DTV, DSTV employs directional derivatives by applying the operator $\mathbf{II}_{(\alpha,\theta)}$ that can act on the gradient at each image point as $(\mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^c)[i] = \mathbf{A}_\alpha \mathbf{R}_{-\theta}(\nabla \mathbf{f}^c)[i]$, while gathering the neighboring gradients to constitute the patch-based Jacobian as:

$$(J_K^{(\alpha,\theta)} \mathbf{f})[i] = \begin{bmatrix} (\mathcal{T}_1 \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^1)[i] & (\mathcal{T}_2 \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^1)[i] & \dots & (\mathcal{T}_L \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^1)[i] \\ (\mathcal{T}_1 \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^2)[i] & (\mathcal{T}_2 \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^2)[i] & \dots & (\mathcal{T}_L \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^2)[i] \\ \vdots & \vdots & \vdots & \vdots \\ (\mathcal{T}_1 \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^c)[i] & (\mathcal{T}_2 \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^c)[i] & \dots & (\mathcal{T}_L \mathbf{II}_{(\alpha,\theta)} \nabla \mathbf{f}^c)[i] \end{bmatrix} \quad (11)$$

$J_K^{(\alpha,\theta)}$ is named as *directional patch-based Jacobian* [23], and involved by the DSTV functional, i.e.,

$$DSTV(\mathbf{f}) = \sum_{i=1}^N \|(J_K^{(\alpha,\theta)} \mathbf{f})[i]\|_{S_q} \quad (12)$$

By means of Eq. (12), one has the chance of leading the STV machinery in favor of a predetermined direction. It has been experimentally shown on uni-directional images in [23] that this incorporation produces prominently better results. Also, since the convexity is preserved in DSTV, the same convex optimization tools used to solve STV can be applied to DSTV based problems.

4.2. Adaptive DSTV (ADSTV)

In real-world images, the dominant directions vary in different regions, which makes DSTV inapplicable. In order to handle arbitrary images, we employ the same way that EADTV employed when bringing adaptivity to DTV. We consider a spatially varying map of orientation, i.e., $\theta : \Omega \mapsto [0, \pi)^C$, and this changes the rotation matrix to $R_{-\theta[i]}$.

Furthermore, in arbitrary images, the anisotropic structures are not homogeneously distributed. Also among the anisotropic structures, their degrees of anisotropy spatially change. DSTV's stretching matrix

A_α uses a constant factor α for the entire image. Just as done to the rotation parameter, one may employ a map of spatially changing stretching factors, so that the dose of penalization at a point can be tuned. However, since the regularization parameter τ is a constant scalar, with varying stretching factor the total amount of regularization changes from region to region. The fittest value of τ for a directional region that requires larger values of $\alpha[i]$ may produce poor results in the regions with isotropic structures, where the corresponding values of $\alpha[i]$ are small. To compensate it, we suggest to use a scaling matrix $\tilde{\Lambda}_{(\alpha^+, \alpha^-[i])}$ with two parameters, one of which is coming from a spatially varying map of $\alpha^- : \Omega \mapsto [1, \alpha^+]^C$, i.e.,

$$\tilde{\Lambda}_{(\alpha^+, \alpha^-[i])} = \begin{bmatrix} \alpha^+ & 0 \\ 0 & \alpha^-[i] \end{bmatrix} \quad (13)$$

While at a point i within a highly directional region, $\alpha^-[i] = 1$; in a flat region, it switches to isotropic behavior: $\alpha^-[i] = \alpha^+$. When it comes to the values between these two end points, as they get closer to one, the sensitivity to the changes towards $\theta[i]$ increases.

As a consequence, the directional patch-based Jacobian operator given in Eq. (11) is extended to its adaptive version (which will be denoted as $\tilde{J}_K^{(\alpha, \theta)}$) that employs $(\tilde{D}_{(\alpha, \theta)} \nabla \mathbf{f}^c)[i] = \tilde{\Lambda}_{(\alpha^+, \alpha^-[i])} \mathbf{R}_{-\theta[i]}(\nabla \mathbf{f}^c)[i]$ to perform directional derivation. Therefore, our ADSTV regularizer is defined as follows:

$$ADSTV(\mathbf{f}) \triangleq \sum_{i=1}^N \|(\tilde{J}_K^{(\alpha, \theta)} \mathbf{f})[i]\|_{S_q} = \|\tilde{J}_K^{(\alpha, \theta)} \mathbf{f}\|_{1, q} \quad (14)$$

4.3. Parameter estimation

Estimation of the parameters α^- and θ is not a trivial task. In a noisy image, line-like structures are often interrupted by the noise, which results in deviations from the correct direction. When the conventional smoothing filters are applied to suppress noise, the same line-like structures get thicker, and the edges are dislocated. The reconnection of the interrupted lines is concerned in many studies, such as the ones that employ the morphological operators for the curvilinear structures in biomedical images [28,29], or for the ridges in fingerprint images [30]. Their principle goal is detection rather than restoration. In the proposed method, we focus finding a layout for the orientations rather than a precise field of them. By means of that, the preprocessor does not put much computational overhead.

In our preprocessor, we employ eigenvectors and eigenvalues of the structure tensor to extract the knowledge about the orientations and the dose of anisotropy, in semi-local fashion, in contrast to EADTV. We also employ isotropic TV regularization to circumvent the problem of interrupted lines and to ignore the insignificant deviations. Below, we explain our parameter estimation procedure, which we will refer to as directional parameter estimator (DPE), step by step in detail. Note that our preprocessor employs Sobel operator as the discretization scheme used for the gradient. Also note that, our parameter estimator uses the luminance information of the input image (\mathbf{g}_L), therefore for the vector-valued images, each channel of the θ will be a copy of each other, i.e., $\theta^1 = \theta^2 = \dots = \theta^C$. The same applies to α^- .

As a measure of anisotropy, we equip a map of non-negative *linearity* factors which are computed as follows at each image point:

$$\mathbf{c}_\sigma[j] = \frac{\lambda_\sigma^+[j] - \lambda_\sigma^-[j]}{\lambda_\sigma^+[j]} \quad (15)$$

where $\lambda_\sigma^\pm[j] = \lambda^\pm((S_K \mathbf{g}_L)_\sigma)[j]$ and $j = \{1, 2, \dots, N\}$. Note that in this section, the variables subscripted by σ denote the pre-smoothed version of \mathbf{g}_L not only with the Gaussian kernel of standard deviation σ , but also with the support $\sigma^2 \times \sigma^2$. Moreover, the kernel $K_{\hat{\sigma}}$ embodied by the structure tensor that we employ here operates $\hat{\sigma}$ larger than $\hat{\sigma}$, again with the support $\hat{\sigma}^2 \times \hat{\sigma}^2$. Eq. (15) is nothing but a measure of the contrast between the directions of the highest and lowest fluctuations. For pure local orientations ($\lambda_\sigma^+ > 0, \lambda_\sigma^- = 0$) the linearity factor is large, while for isotropic structures ($\lambda_\sigma^+ = \lambda_\sigma^- > 0$) it is zero. We

provide Fig. 1(a) to illustrate the variations of the eigenvectors scaled by the squared eigenvalues, which actually visualizes the structure tensors. After computing \mathbf{c}_σ , by treating it as a corrupted image, we apply TV-based regularization. This way, the fluctuating intensities are suppressed, thus the uncertain responses are refused. The ROF model [1] that serves our purpose is given below:

$$\hat{\kappa}_\sigma = \operatorname{argmin}_{\kappa_\sigma \in [0, 1]^N} \|\mathbf{c}_\sigma - \kappa_\sigma\|_2^2 + TV(\kappa_\sigma) \quad (16)$$

To be able to detect all desired structures, the response of TV regularized linearity measure $\hat{\kappa}_\sigma$ is analyzed at different scales (i.e., $\sigma_k^2 = 2k - 1$, where $k = \{1, 2, \dots, K\}$). Since as the scale parameter σ_k gets larger, the false positive edge responses come into being; our method follows the iterative procedure shown below by giving more credits to the responses obtained at the fine-scales:

$$\kappa^{(k)}[j] = \begin{cases} \kappa^{(k-1)}[j], & \text{if } \hat{\kappa}_{\sigma_k}[j] \leq \kappa^{(k-1)}[j] \\ \frac{1}{2}(\kappa^{(k-1)}[j] + \hat{\kappa}_{\sigma_k}[j]), & \text{otherwise} \end{cases} \quad (17)$$

We suggest using $K = 2$ for the noise levels less than 0.2, while $K = 3$ for the higher ones. If the image at hand is thought to be either highly directional or highly nondirectional, our algorithm further improves the linearity values. It proceeds by enhancing the values higher than the mean in a directional image, while diminishing the values less than the mean in a nondirectional one. This procedure is also executed at each scale, i.e.,

$$\phi^{(k)}[j] = \begin{cases} \mathcal{V}(\kappa^{(k)}[j], 1), & \text{if } \gamma_1^{(k)} > 1 \\ \mathcal{V}(\sqrt[4]{\kappa^{(k)}[j]}, -1), & \text{else if } \gamma_1^{(k)} < -1 \\ \kappa^{(k)}[j], & \text{otherwise} \end{cases} \quad (18)$$

Here the function $\mathcal{V}(X, s) = \begin{cases} X^2, & \text{if } sX < s\mu^{(k)} \\ X, & \text{otherwise} \end{cases}$, where $\mu^{(k)}$ is the

mean of the TV regularized linearity map, i.e., $\mu^{(k)} = E[\hat{\kappa}^{(k)}]$, and the variable $\gamma_1^{(k)}$ denotes the skewness of the TV regularized linearity histogram at the scale σ_k , i.e., $\gamma_1^{(k)} = \text{Skewness}[\kappa^{(k)}]$. If the skewness is greater than 1, the distribution of the linearity values is considered to be highly skewed to the right, which implies that the nondirectional regions predominate the directional ones. If it is less than -1 , the distribution is highly left skewed and it can be inferred that the image is highly directional. Note that in our implementation, we used *skewness*(\cdot) function of Matlab.

Finally, our algorithm obtains the entries of α^- , by inverse scaling the entries of $\phi \in [0, 1]^N$ onto the range $[1, \alpha^+]$, so that the regions with higher response to the above mentioned linearity estimation procedure will be regularized by using smaller $\alpha^-[j]$.

$$\alpha^-[j] = \frac{\alpha^+ - 1}{\max(\phi) - \min(\phi)} (\max(\phi) - \phi[j]) + 1 \quad (19)$$

When it comes to the estimation of θ , we first consider the eigendirections of the structure tensor, computed at a scale with the highest response to Eq. (17), i.e.,

$$\vartheta[j] = \angle \mathbf{v}_{\hat{\sigma}_k}^-[j], \quad \hat{\sigma}_k = \operatorname{argmax}_{\sigma_1 < \dots < \sigma_K} \hat{\kappa}_{\sigma_k}[j] \quad (20)$$

where $\mathbf{v}_{\hat{\sigma}_k}^-[j]$ is the eigenvector corresponding to the smallest eigenvalue $\lambda_{\hat{\sigma}_k}^+[j]$. Next, as was done to the linearity factor, the estimated angles are TV regularized as follows:

$$\hat{\theta} = \operatorname{argmin}_{\theta \in [0, \pi]^N} \frac{1}{2} \|\vartheta - \theta\|_2^2 + \tau TV(\theta) \quad (21)$$

where we set $\tau = 0.02$ for all our experiments. Setting τ to a larger value may cause deviations from the correct directions due to the contrast loss.

In Fig. 1(b)–(e), we present colormaps taken from the different stages of our DPE procedure applied to the exemplary *Zebra* image from Berkeley Segmentation Dataset (BSD) [31], contaminated with the noise level $\sigma_\eta = 0.15$. Fig. 1(b) shows the linearity map extracted from

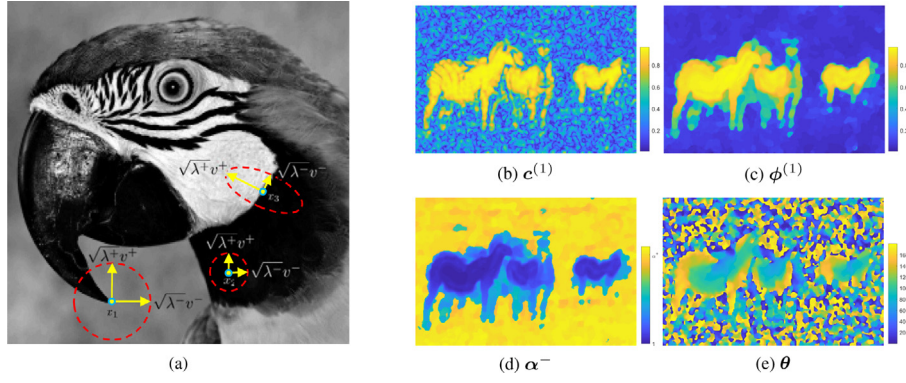


Fig. 1. Illustrations and exemplary outputs for the DPE procedure. (a) Schema for the variations of the eigenvalues and eigenvectors, which visualizes the structure tensors in ellipse form. The colormaps on the RHS respectively show (b) the linearity map of unsmoothed \mathbf{g}_L , (c) its TV regularized version, (d) the final weights, and (e) the final directions for the noise level $\sigma_\eta = 0.15$.

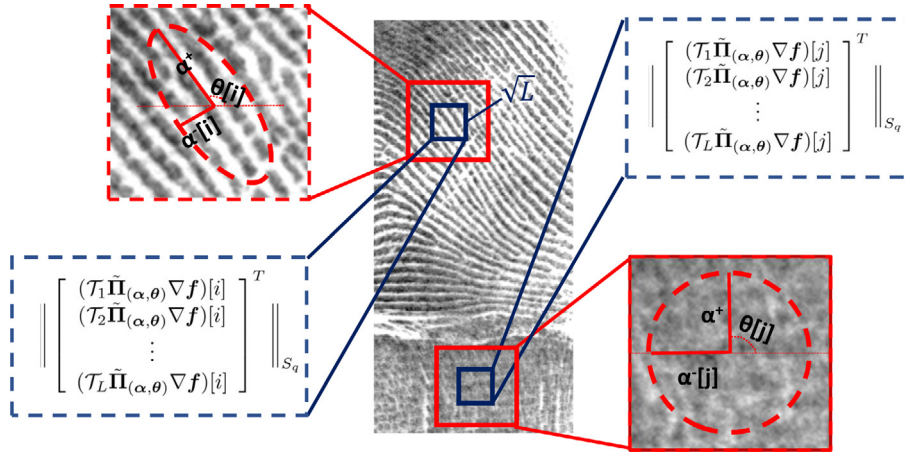


Fig. 2. A rough illustration of the DPE procedure vs. ADSTV denoiser. While the red frames refer to the DPE procedure, the navy blue indicates ADSTV based denoising.

the luminance image by using $\bar{\sigma} = \sqrt{11}$. Thanks to the neighborhood-awareness of the structure tensor, the directions of the small-scale structures are reliably distinguished without the need of pre-smoothing. Fig. 1(c) demonstrates the TV regularized linearity map after the application of Eq. (16). One can see that the image points within highly directional 11×11 neighborhood respond with very high linearity values (e.g., the bodies of the zebras), while the response attenuates as the point gets closer to a nondirectional region (e.g. the contours of the bodies). This gradual decrease prevents the affects of the sharp transitions from the anisotropic behavior to the isotropic one. Due to the uncertainties and the larger-scale directional structures that could not be extracted, the background is all suppressed. In Fig. 1(d), we show the colormap of the final α^- , obtained after applying one more sequence of Eqs. (15), (16), (17), and (18) at the scale with σ_2 , and finally Eq. (19). Additional to the result obtained at the scale σ_1 , one can see the inclusion of some directional structures at the background encoded in orangish colors. Eventually, Fig. 1(e) shows the final distribution of the orientations, i.e., $\theta = \hat{\theta}$. Furthermore, in Fig. 2, we roughly illustrate the idea behind our two stage framework. The directional descriptors are visualized as an ellipse, whose major and minor axes respectively corresponds to the α^+ and $\alpha^-[i]$ at a point i , and its orientation coincides with the orientation to be penalized.

4.4. Numerical optimization

According to the objective function given in Eq. (1), the problem to be solved is:

$$\hat{\mathbf{f}} = \underset{\mathbf{f} \in C}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{g} - \mathbf{f}\|_2^2 + \tau \|\tilde{J}_K^{(\alpha, \theta)} \mathbf{f}\|_{1,q} \quad (22)$$

where C is a set corresponding to an additional constraint ($C = \mathbb{R}^{NC}$ in unconstrained case). The solution to this problem corresponds to the proximal map of $ADSTV(\mathbf{f})$, i.e., $\operatorname{prox}_{\tau ADSTV}(\mathbf{g})$. In [23], $\operatorname{prox}_{\tau DSTV}(\mathbf{g})$ was solved through the derivation of dual problem in detail. Having ADSTV instead of DSTV does not change the primal problem, hence the way that we derive dual will remain same. However, DSTV was only considering Frobenius norm ($q = 2$). We define ADSTV in more general form, considering $q \geq 1$, as a direct extension of STV. In this respect, to come up with a concise derivation, using the fact that the dual of the norm $\|\cdot\|_{S_q}$ is $\|\cdot\|_{S_p}$, where $\frac{1}{q} + \frac{1}{p} = 1$, one can redefine ADSTV in terms of the support function of the form:

$$ADSTV(\mathbf{f}) = \sup_{\Psi[i] \in B_{S_p}} \langle \tilde{J}_K^{(\alpha, \theta)} \mathbf{f}, \Psi \rangle \quad (23)$$

by introducing a variable $\Psi \in \mathbb{R}^{N \times LC \times 2}$, and the set $B_{S_p} = \{\mathbf{X} \in \mathbb{R}^{LC \times 2} : \|\mathbf{X}\|_{S_p} \leq 1\}$. Here $\Psi[i]$ refers to the i th submatrix of Ψ . This paves the way for rewriting the problem in Eq. (22) as a minimax problem:

$$\min_{\mathbf{f} \in C} \max_{\Psi[i] \in B_{S_p}} \frac{1}{2} \|\mathbf{g} - \mathbf{f}\|_2^2 + \tau \langle \mathbf{f}, \tilde{J}_K^{(\alpha, \theta)*} \Psi \rangle \quad (24)$$

where $\tilde{J}_K^{(\alpha, \theta)*}$ arising after we leave \mathbf{f} alone in the second term denotes the adjoint. The adjoint of the patch-based Jacobian $J_K^* : \mathbb{R}^{N \times LC \times 2} \rightarrow \mathbb{R}^{NC}$ was defined in [14] with its derivation:

$$(J_K^* \mathbf{X})[t] = \sum_{l=1}^L -\operatorname{div}(\mathcal{T}_l^* \mathbf{X}[l, s]) \quad (25)$$

where $s = (c-1)L + l$ and $t = (c-1)N + n$ with $1 \leq n \leq N$ and $1 \leq c \leq C$. \mathcal{T}_l^* corresponds to the adjoint of \mathcal{T}_l , which scans the

$\mathbf{X}[i, s]$ in column-wise manner, where $\mathbf{X}[i, s] \in \mathbb{R}^2$ is the s th row of the i th submatrix of an arbitrary $\mathbf{X} \in \mathbb{R}^{N \times LC \times 2}$. Also, the operator div is discrete divergence, applied by using backward differences, since the gradient is discretized using forward differences, as in [14]. From this point of view, in order to define our $\tilde{J}_K^{(\alpha, \theta)^*} : \mathbb{R}^{N \times LC \times 2} \mapsto \mathbb{R}^{NC}$, one should only change the divergence operator in Eq. (25) with the directional divergence operator $\widetilde{\text{div}}_{(\alpha[i], \theta[i])}$, i.e.,

$$(\tilde{J}_K^{(\alpha, \theta)^*} \mathbf{X})[i] = \sum_{l=1}^L -\widetilde{\text{div}}_{(\alpha[i], \theta[i])} (\mathcal{T}_l^* \mathbf{X}[i, s]) \quad (26)$$

whose definition is given below:

$$\begin{aligned} \widetilde{\text{div}}_{(\alpha[i], \theta[i])}(\cdot) &\triangleq \text{div}(\tilde{\mathbf{H}}_{(\alpha[i], \theta[i])}^T(\cdot)) \\ &= \text{div}(\mathbf{R}_{\theta[i]} \tilde{\mathbf{A}}_{(\alpha^+, \alpha^-[i])}(\cdot)) \end{aligned} \quad (27)$$

Let us call the objective function in Eq. (24) as $\mathcal{L}(\mathbf{f}, \Psi)$. It is convex w.r.t. \mathbf{f} and concave w.r.t. Ψ . By following the fast gradient projection (FGP) method [32], which combines the dual approach introduced in [33] to solve TV-based denoising, and the fast iterative shrinkage-thresholding (FISTA) [34] to accelerate convergence, one can swap min and max, i.e.,

$$\min_{\mathbf{f} \in \mathcal{C}} \max_{\Psi[i] \in B_{S_p}} \mathcal{L}(\mathbf{f}, \Psi) = \mathcal{L}(\hat{\mathbf{f}}, \hat{\Psi}) = \max_{\Psi[i] \in B_{S_p}} \min_{\mathbf{f} \in \mathcal{C}} \mathcal{L}(\mathbf{f}, \Psi) \quad (28)$$

since the common saddle point is not affected. Maximization of the dual problem $d(\Psi) = \min_{\mathbf{f} \in \mathcal{C}} \mathcal{L}(\mathbf{f}, \Psi)$ at the right-hand side is same with the minimization of the primal problem $P(\mathbf{f}) = \max_{\Psi[i] \in B_{S_p}} \mathcal{L}(\mathbf{f}, \Psi)$ at the left-hand side. Therefore, finding the maximizer $\hat{\Psi}$ of $d(\Psi)$ serves to find the minimizer $\hat{\mathbf{f}}$ of $P(\Psi)$. One can rewrite $\hat{\mathbf{f}}$ in terms of Ψ , as:

$$\hat{\mathbf{f}} = \underset{\mathbf{f} \in \mathcal{C}}{\text{argmin}} \|\mathbf{f} - (\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)^*} \Psi)\|_2^2 - M \quad (29)$$

by expanding $\mathcal{L}(\mathbf{f}, \Psi)$ and collecting the constants under the term M . The solution to Eq. (29) is:

$$\hat{\mathbf{f}} = P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)^*} \Psi) \quad (30)$$

where P_C is the orthogonal projection onto the set \mathcal{C} . Then we proceed by plugging $\hat{\mathbf{f}}$ in $\mathcal{L}(\mathbf{f}, \Psi)$ to get $d(\Psi) = \mathcal{L}(\hat{\mathbf{f}}, \Psi)$, i.e.,

$$d(\Psi) = \frac{1}{2} \|\mathbf{w} - P_C(\mathbf{w})\|_2^2 + \frac{1}{2} (\|\mathbf{g}\|_2^2 - \|\mathbf{w}\|_2^2) \quad (31)$$

where $\mathbf{w} = \mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)^*} \Psi$. Dual problem is smooth with a well-defined gradient computed:

$$\nabla d(\Psi) = \tau \tilde{J}_K^{(\alpha, \theta)} P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)^*} \Psi) \quad (32)$$

based on the derivation in Lemma 4.1 of [32]. Therefore, in order to find the maximizer $\hat{\Psi}$ of $d(\Psi)$, one can employ the projected gradients [33]. It performs decoupled sequences of gradient descent/ascent and projection onto a set. For the gradient ascent in our case, an appropriate step size that ensures the convergence needs to be selected. For the gradient in Eq. (32), a $LC \times 2$ matrix filled by a constant step size $1/L[i]$ can be used for each image point, where $\mathbf{L} \in \mathbb{R}^{N \times LC \times 2}$ is a multidimensional array of Lipschitz constants each of which having the upper bound $L[i] \leq 8\sqrt{2}\tau^2((\alpha^+)^2 + (\alpha^-[i])^2)$. One can see Appendix for the derivation.

When it comes to the projection, as an efficient way, the authors of [14] reduced the projection of each $\Psi[i]$ onto B_{S_p} to the projection of the singular values onto the ℓ_p unit ball $B_p = \{\sigma \in \mathbb{R}_+^2 : \|\sigma\|_p \leq 1\}$. Therefore, for the SVD $\Psi[i] = \mathbf{U}\Sigma\mathbf{V}^T$ and $\Sigma = \text{diag}(\sigma)$, where $\sigma = [\sigma_1, \sigma_2]$, the projection was defined as:

$$P_{B_{S_p}}(\Psi[i]) = \Psi[i]\mathbf{V}\Sigma^+\Sigma_p\mathbf{V}^T \quad (33)$$

where Σ^+ is pseudoinverse of Σ and $\Sigma_p = \text{diag}(\sigma_p)$ for σ_p are the projected singular values of Σ . The readers are referred to [14] for the details about the derivation and the realization. As a consequence, the overall algorithm is shown in Algorithm 1.

Algorithm 1 Algorithm for ADSTV-based denoising

INPUT: \mathbf{g} , $\alpha^+ > 1$, $\tau > 0$, $p > 1$, P_C
INIT: $\Psi^{(0)} = \mathbf{0} \in \mathbb{R}^{N \times LC \times 2}$, $t^{(1)} = 1$, $i = 1$
 $(\theta, \alpha^-) \leftarrow \text{DPE}(\mathbf{g}, \alpha^+)$ [Section 4.3]
 $\mathbf{L} \leftarrow 8\sqrt{2}\tau((\alpha^+)^2 + (\alpha^-)^2)$
while stopping criterion is not satisfied **do**
 $\mathbf{z} \leftarrow P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)^*} \Psi^{(i-1)})$
 $\Psi^{(i)} \leftarrow P_{B_{S_p}}(\Psi^{(i-1)} + \mathbf{L}^{\circ-1} \odot (\tilde{J}_K^{(\alpha, \theta)} \mathbf{z}))$
 $(\Psi^{(i+1)}, t^{(i+1)}) \leftarrow \text{FISTA}(t^{(i)}, \Psi^{(i)}, \Psi^{(i-1)})$
 $i \leftarrow i + 1$
end while
return $P_C(\mathbf{g}^{(k)} - \tau \tilde{J}_K^{(\alpha, \theta)^*} \Psi^{(j)})$

function FISTA($t, \mathbf{f}^{(\text{cur})}, \mathbf{f}^{(\text{prev})}$) [34]
 $t^{(\text{next})} \leftarrow (1 + \sqrt{1 + 4t^2})/2$
 $\mathbf{f}^{(\text{next})} \leftarrow \mathbf{f}^{(\text{cur})} + (\frac{t-1}{t^{(\text{next})}})(\mathbf{f}^{(\text{cur})} - \mathbf{f}^{(\text{prev})})$
 return $(\mathbf{f}^{(\text{next})}, t^{(\text{next})})$
end

5. Experiments

In this section, we assess qualitative and quantitative performances of the proposed variational denoising framework. We compare it with the other related analysis-based regularizers: TV (as a baseline), EADTV (as a representative of previous attempts to make DTN [9] adaptive), STV [14] (as a predecessor of ADSTV), DSTV [23] (to clarify the contribution of the adaptivity), and NCDR [22] (as a recent analysis-based regularization scheme depending on a new paradigm). We do not include TGV [6] in the competing algorithms since, STV's superiority over it has already been demonstrated in [14]. Owing to the fact that TV, EADTV, DSTV, and NCDR are only applicable to the scalar-valued images, they are only involved in the grayscale environment. Thus, the experiments on the vector-valued images merely compare STV and DSTV regularizers. The source codes of STV² and NCDR³ that we use were made publicly available by the authors on GitHub. Our ADSTV is implemented on top of STV, while TV and EADTV are written from scratch. All methods were written in MATLAB by only making use of the CPU. The experiments were performed on a computer equipped with Intel Core Processor i7-7500U (2.70 GHz) with 16 GB of memory.

To assess the quantitative performances of the methods, we use peak signal-to-noise ratio (PSNR) in dB and structural similarity index (SSIM). The experiments are performed on the images shown in Fig. 3. The first row shows the thumbnails of seven classical grayscale test images of sizes 256×256 (*Monarch*, *Shells*, and *Feather*) and 512×512 (the others). In the other two rows, we have RGB color images. *Plant*, *Spiral*, *Rope* and *Corns* of sizes 256×256 are more textural with different directional characteristics. *Rope* and *Corns* are taken from the popular Berkeley Segmentation Dataset (BSD) [31], while the others are public domain images. The rest of the color images are natural images taken from BSD. They have either no or limited amount of directional parts. Note that, the intensities of all test images have been normalized to the range $[0, 1]$.

The algorithms under consideration are all aiming to minimize Eq. (1). Therefore, the critical regularization parameter τ is in common, and should be chosen precisely. For a fair comparison, we fine-tuned τ such that it leads to the best PSNR in all experiments. For STV, DSTV, ADSTV, and NCDR priors; one should also choose the convolution kernel. We used a Gaussian kernel of support 3×3 for all four algorithms. The standard deviations (σ) were set to 0.5 for STV, DSTV, and ADSTV, while $\sigma = 1$ was used for NCDR, as suggested in the

² https://github.com/cig-skoltech/Structure_Tensor_Total_Variation

³ <https://github.com/HUST-Tan/NCDR>

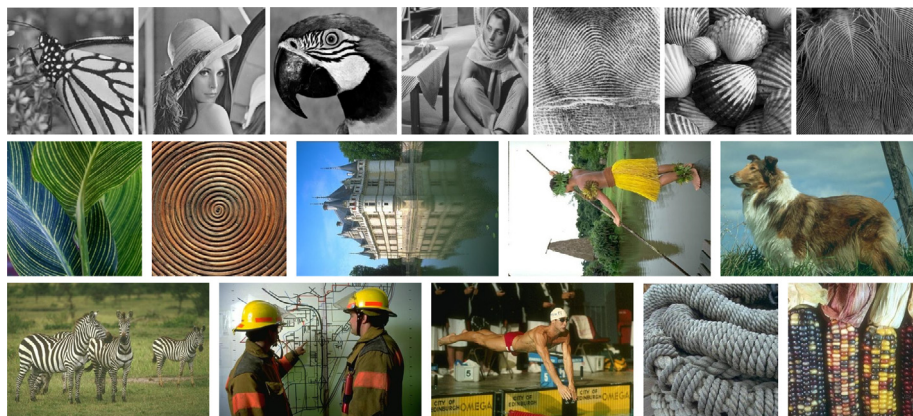


Fig. 3. Thumbnails of the grayscale and color images used in the experiments. From left to right and top to bottom: *Monarch*, *Lena*, *Parrot*, *Barbara*, *Fingerprint*, *Shells*, *Feather*, *Plant*, *Spiral*, *Chateau*, *Indigenous*, *Dog*, *Zebra*, *Workers*, *Swimmer*, *Rope*, and *Corns*.

algorithms' respective papers: [14] and [22]. All of STV, DSTV, and ADSTV use the nuclear norm of the patch-based Jacobian, since it was chosen as the best performing norm. Similar to the convolution kernel, all the other parameters of NCDR were kept same as they were used in [22]. α is fine-tuned by changing it from 2 (since it reduces to TV when $\alpha = 1$) to 30 with 1-unit intervals for both EADTV and DSTV. This is also the case for the ADSTV's α^+ , unless otherwise stated. Note that DSTV's α is almost always tuned to two, since with the larger values of α , DSTV model gains more anisotropy only in a dominant direction, which does not apply globally for our arbitrary images. ADSTV requires an additional convolution kernel too, to be used by the preprocessor DPE. We fixed it as a Gaussian kernel with variance and support $\bar{\sigma}^2 = 7$ for 256×256 images, while $\bar{\sigma}^2 = 11$ for 963×481 natural images from BSD, and $\bar{\sigma}^2 = 15$ for 512×512 images. For the optimization of all TV-based algorithms, the stopping criterion was either having a relative normed difference between two successive estimates that reaches a threshold of 10^{-5} , or fulfilling a number of iterations, equals to 100 in our case. When it comes to NCDR, since it uses gradient descent to solve its non-convex optimization problem, it is quite hard to ensure that whether a reported result is due to a local or a global minimum. The number of iterations for NCDR was experimentally set to 300.

We consider additive i.i.d. Gaussian noise with five noise levels of $\sigma_\eta = \{0.05, 0.1, 0.15, 0.2, 0.25\}$. In Table 1, we report PSNR and SSIM measures obtained by using TV, EADTV, STV, DSTV, NCDR, and ADSTV models applied to the grayscale images. From the results, we observe that the EADTV does not offer significant improvements over TV in the images with less directional patches such as *Lena* and *Parrot*. But, it even performs worse than TV for *Barbara* and *Feather* images, in which highly directional parts present, as the noise level increases. This is due to the fact that, those directional parts involve very high frequency components. Thus, even small deviations in the direction estimation extremely affects the result. In almost all of the experiments, EADTV preferred to use $\alpha = 2$ (except *Fingerprint* for which larger α values are selected), and this is also due to the need of compensating the sheer amount of mistakes made by its direction estimator. STV produced significantly better results when compared to TV and EADTV for all cases and images, except *Fingerprint*, where for all the noise levels $\sigma_\eta > 0.1$, EADTV outperformed STV. As might be expected, STV and DSTV produced very similar results, since $\alpha = 2$ was selected, which almost reduces the DSTV to STV. When it comes to NCDR, it yielded superior PSNR results to all competing algorithms including ADSTV in many experiments. However, as one may see in the last column, these results are obtained at the cost of computational load. The average runtimes of NCDR at all noise levels are pretty high. On the other hand, when the quality is measured by SSIM, NCDR (and the others) seems to be outperformed by our ADSTV. Fig. 4 is provided for the visual judgment. It can be inferred that NCDR smooths the structural regions

better than ADSTV at the risk of loss of details (See (d)–(e), (i)–(j), and the scarf of *Barbara* in (s)–(t)). NCDR also causes artifacts in the junctions more apparent than those of ADSTV (e.g., (d)–(e)). Also the ADSTV is better at smoothing flat region as one can see on *Lena*'s face in (n)–(o). These may clarify why the SSIM values obtained by ADSTV are higher. In Fig. 5, we present some visual results to compare DSTV and ADSTV. From the figure, one can infer what adaptivity brings. When $\alpha = 2$, the regions that are penalized in wrong directions are not that distinguishable in Fig. 5(b), but in Fig. 5(d), the deterioration in the dominant direction is considerably visible.

Table 2, on the other side, shows PSNR/SSIM values obtained by the application of STV and ADSTV to vector-valued images. According to the results, ADSTV systematically outperformed STV in terms of PSNR and SSIM measures, even in the images that do not exhibit directional dominance. However, in terms of the runtimes that are reported in the last row, ADSTV seems to bring almost twice as much load to the computation. Again in Fig. 6, we demonstrate exemplary detail patches cropped from the original images (first column), noisy versions (second column), and the restored versions by STV (third column) and ADSTV (last column). The results obtained by STV have oil-painting-like artifacts, whereas this effect is far less visible in ADSTV's results. With the incorporation of the directional information, the edges became more apparent, (see *Rope* and *Chateau*), more smoothing towards the desired direction (see *Spiral*), and the gaps between closely parallel lines could better be distinguished (e.g. the feather in *Dog* image, the details of the leaves in *Corns* image, and skirt in *Indigenous* image).

One of the ADSTV's disadvantages is introducing an extra free parameter α^+ that needs to be tuned. In Fig. 7, we present two graphs to show how sensitive the gain over STV by using our method with respect to α^+ is. Fig. 7(a) serves the purpose by testing varying α^+ values on five different grayscale and color images at the noise level $\sigma_\eta = 0.15$, while Fig. 7(b) examines them on *Barbara* image at five different noise levels. As one can infer from the results, even for the incorrect selections of α^+ , the gain has almost always been above zero, which shows the robustness of our approach. Yet a relevant research question here might be if it is possible to calculate α^+ automatically, as a function of an estimated noise level by using variance estimators and the linearity factors.

Finally, the graph in Fig. 8 shows the performances of the competing algorithms across the iterations. The y-axis is for the PSNR improvement with respect to the noisy input image. From the graph, one can imply that the convergence behaviors of the algorithms are similar. Note that, the reason why the NCDR starts from a better point (somewhat improved already) is that it preprocesses the image by running same number of NCDR iterations to adjust the parameters.

Table 1

PSNR/SSIM Comparison of TV, EADTV, STV, DSTV, NCDR, and ADSTV on grayscale images. Last column shows the average runtimes.

		Monarch	Lena	Parrot	Barbara	Fingerp.	Shell	Feather	Avg.	t (s)
$\sigma_\eta = 0.05$	TV	31.28/0.91	32.88/0.87	33.17/0.88	29.48/0.84	29.14/0.93	28.77/0.89	26.73/0.94	30.21/0.89	1.51
	EADTV	31.63/0.91	33.09/0.87	33.16/0.88	29.59/0.83	29.73/0.94	28.92/0.89	26.78/0.94	30.41/0.89	1.82
	STV	32.03/0.92	33.59/0.88	33.56/0.90	30.19/0.88	30.11/0.94	29.17/0.90	27.07/0.94	30.82/0.91	17.41
	DSTV	32.02/0.92	33.66/0.88	33.57/0.90	30.34/0.86	30.10/0.95	29.16/0.90	27.21/0.94	30.87/0.91	14.63
	NCDR	32.25/0.92	34.26/0.89	33.44/0.90	31.47/0.89	30.08/0.94	29.41/0.90	27.66/0.95	31.22/0.91	247.85
ADSTV	32.38/0.93	34.03/0.89	33.80/0.91	31.09/0.89	30.09/0.94	29.57/0.91	27.71/0.95	31.24/0.92	38.24	
$\sigma_\eta = 0.1$	TV	27.72/0.84	29.86/0.81	29.74/0.82	25.57/0.71	25.33/0.86	25.03/0.78	21.84/0.81	26.45/0.80	2.08
	EADTV	28.14/0.84	30.07/0.81	29.81/0.81	25.51/0.70	25.83/0.87	24.56/0.76	21.56/0.81	26.50/0.80	2.77
	STV	28.48/0.86	30.54/0.83	30.24/0.84	26.45/0.77	26.21/0.87	25.49/0.80	22.29/0.83	27.10/0.83	16.92
	DSTV	28.53/0.86	30.61/0.83	30.15/0.83	26.39/0.75	26.22/0.88	25.44/0.79	22.66/0.84	27.15/0.83	14.26
	NCDR	28.99/0.87	31.36/0.84	30.36/0.84	27.57/0.79	26.55/0.88	25.88/0.81	23.61/0.86	27.76/0.84	254.58
ADSTV	28.90/0.87	31.15/0.84	30.49/0.85	27.34/0.81	26.48/0.88	26.01/0.82	23.37/0.86	27.68/0.84	34.59	
$\sigma_\eta = 0.15$	TV	25.73/0.79	28.23/0.77	27.91/0.77	23.88/0.64	23.30/0.79	23.15/0.69	19.47/0.68	24.53/0.73	2.28
	EADTV	26.13/0.79	28.38/0.77	28.02/0.77	23.78/0.63	24.22/0.82	23.32/0.70	18.94/0.67	24.68/0.74	3.07
	STV	26.47/0.82	28.85/0.79	28.43/0.79	24.47/0.69	24.20/0.81	23.62/0.72	19.92/0.70	25.14/0.76	16.77
	DSTV	26.45/0.81	28.91/0.79	28.30/0.79	24.49/0.67	24.07/0.82	23.55/0.71	20.35/0.72	25.17/0.76	14.25
	NCDR	27.09/0.83	29.59/0.80	28.71/0.80	25.46/0.72	24.64/0.82	24.10/0.73	21.30/0.77	25.84/0.78	205.57
ADSTV	26.87/0.83	29.45/0.81	28.79/0.81	25.40/0.74	24.46/0.81	24.18/0.74	21.19/0.76	25.76/0.79	31.63	
$\sigma_\eta = 0.2$	TV	24.35/0.74	27.14/0.74	26.68/0.73	23.04/0.61	21.94/0.72	21.96/0.62	18.09/0.54	23.32/0.67	2.78
	EADTV	24.57/0.76	27.22/0.74	26.77/0.73	22.96/0.60	22.80/0.77	22.10/0.63	17.99/0.54	23.49/0.68	2.58
	STV	25.06/0.78	27.69/0.76	27.22/0.76	23.45/0.65	22.68/0.76	22.39/0.65	18.51/0.58	23.86/0.70	16.21
	DSTV	25.02/0.77	27.72/0.76	27.04/0.75	23.43/0.63	22.58/0.76	22.33/0.64	18.97/0.62	23.88/0.71	14.89
	NCDR	25.70/0.79	28.31/0.77	27.56/0.77	24.20/0.66	23.26/0.77	22.86/0.66	19.68/0.66	24.51/0.72	203.09
ADSTV	25.61/0.80	28.29/0.78	27.67/0.79	23.98/0.67	23.12/0.77	22.92/0.67	19.46/0.65	24.44/0.73	36.77	
$\sigma_\eta = 0.25$	TV	23.28/0.71	26.33/0.71	25.75/0.71	22.52/0.58	20.93/0.67	21.10/0.56	17.24/0.42	22.45/0.62	2.76
	EADTV	23.58/0.71	26.35/0.72	25.82/0.71	22.48/0.58	21.68/0.72	21.23/0.57	17.10/0.41	22.61/0.63	2.82
	STV	23.96/0.74	26.82/0.74	26.30/0.74	22.90/0.61	21.58/0.70	21.50/0.60	17.60/0.46	22.96/0.66	16.60
	DSTV	23.91/0.74	26.83/0.74	26.08/0.73	22.78/0.60	21.46/0.70	21.44/0.59	18.04/0.52	22.94/0.66	15.57
	NCDR	24.59/0.76	27.31/0.75	26.64/0.74	23.30/0.62	22.15/0.71	21.90/0.60	18.43/0.55	23.47/0.68	224.96
ADSTV	24.44/0.76	27.37/0.76	26.83/0.76	23.22/0.63	22.11/0.71	22.01/0.62	18.24/0.54	23.46/0.68	36.70	

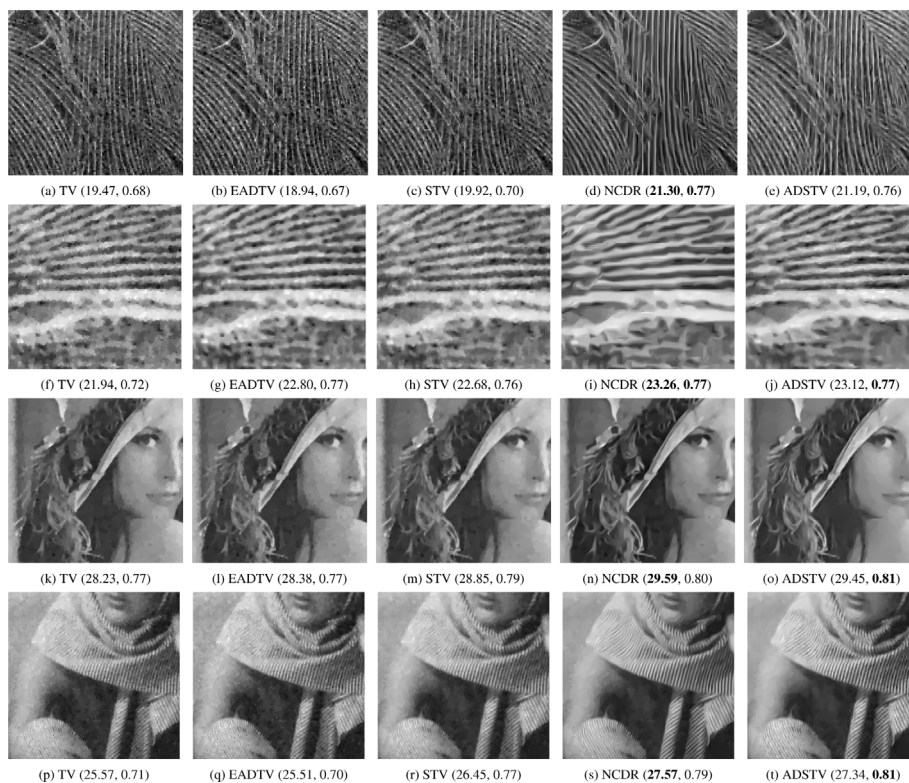


Fig. 4. The detail patches showing grayscale image denoising results. The noisy *Barbara* ($\sigma_\eta = 0.1$), *Feather*, *Lena* ($\sigma_\eta = 0.15$), and *Fingerprint* ($\sigma_\eta = 0.2$) images are restored by using TV (col-1), EADTV (col-2), STV (col-3), NCDR (col-4), and ADSTV (col-5) regularizers. The quantity pairs shown at the bottom of each image are corresponding to the (PSNR, SSIM) values.

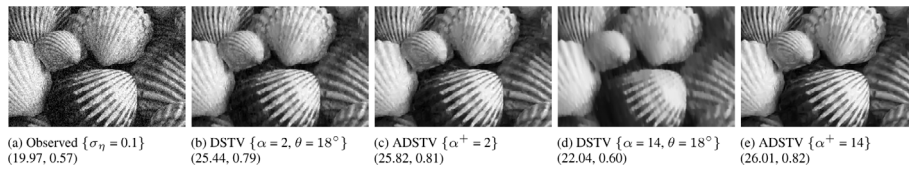


Fig. 5. The detail patches showing the denoising results for the noisy *Shell* image (a), by using DSTV (b and d) and ADSTV (c and e) regularizers with two different values of α and α^+ . The quantity pairs shown at the bottom of each image are corresponding to the (PSNR, SSIM) values.

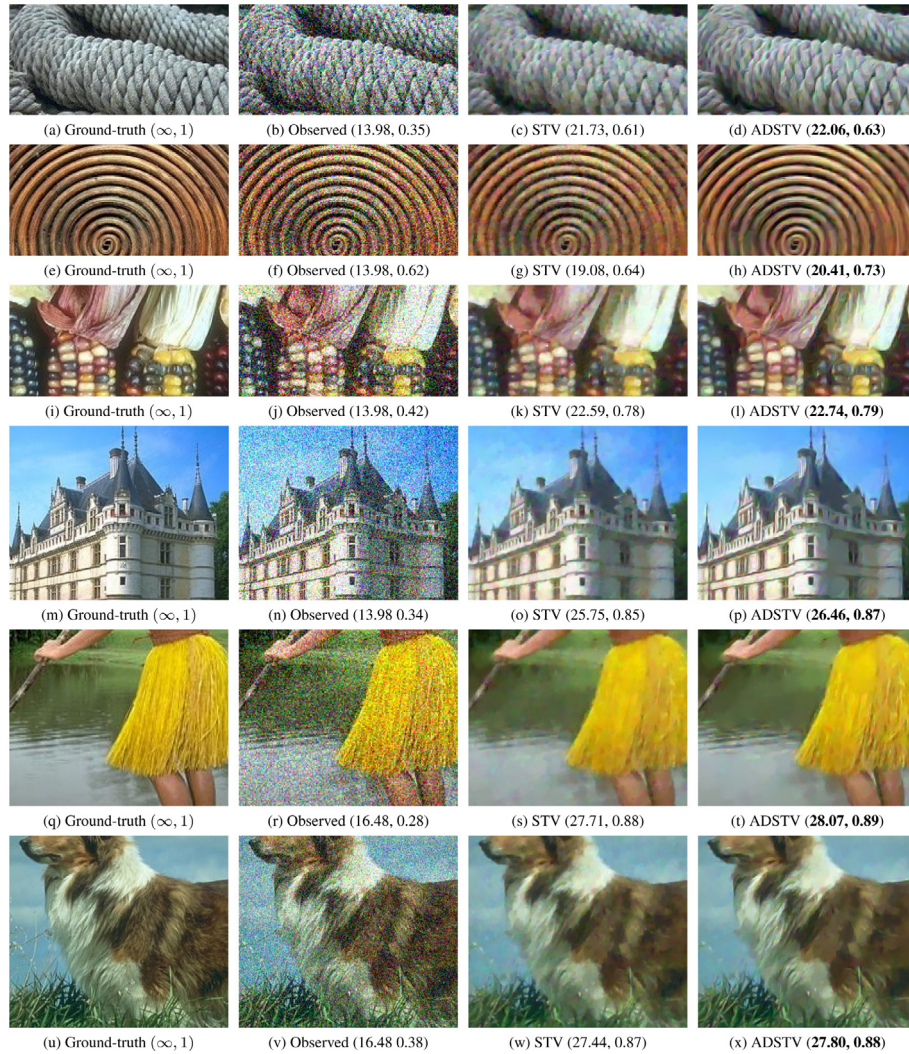


Fig. 6. The detail patches showing color image denoising results. The noisy *Indigenous*, *Dog* ($\sigma_\eta = 0.15$), and *Rope*, *Spiral*, *Corns*, *Chateau*, *Indigenous* ($\sigma_\eta = 0.2$) images (col-2) are restored by using STV (col-3) and ADSTV (col-4) regularizers. The quantity pairs shown at the bottom of each image are corresponding to the (PSNR, SSIM) values.

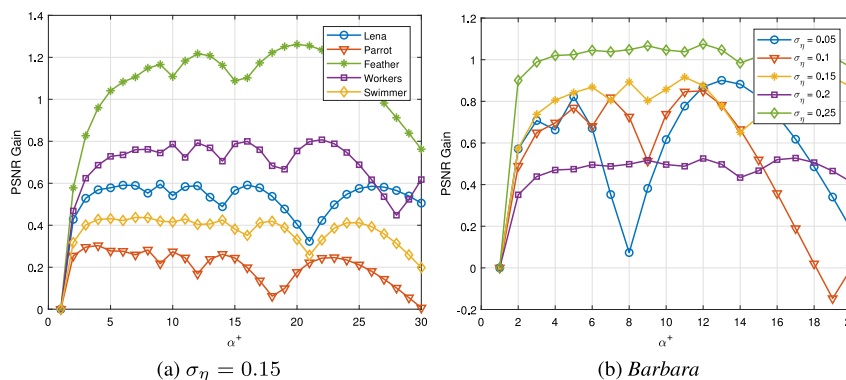


Fig. 7. PSNR gain of ADSTV over STV with respect to α^+ .

Table 2
PSNR/SSIM Comparison of STV and ADSTV on color image denoising. Last row shows the average runtimes.

	$\sigma_n = 0.05$		$\sigma_n = 0.1$		$\sigma_n = 0.15$		$\sigma_n = 0.2$		$\sigma_n = 0.25$	
	STV	ADSTV	STV	ADSTV	STV	ADSTV	STV	ADSTV	STV	ADSTV
Plant	29.23/0.96	30.32/0.97	25.27/0.90	26.63/0.93	23.28/0.85	24.60/0.89	22.05/0.80	23.24/0.85	21.21/0.76	22.27/0.81
Spiral	27.82/0.97	28.39/0.98	24.02/0.93	24.76/0.94	22.04/0.89	23.06/0.91	20.73/0.86	21.95/0.89	19.74/0.83	21.08/0.87
Chateau	32.43/0.95	32.74/0.96	28.83/0.91	29.39/0.92	26.93/0.87	27.63/0.89	25.75/0.85	26.46/0.87	24.88/0.83	25.51/0.85
Indigenous	32.70/0.95	33.04/0.96	29.33/0.90	29.74/0.92	27.71/0.88	28.07/0.89	26.70/0.85	26.99/0.87	25.99/0.84	26.25/0.85
Dog	32.19/0.95	32.53/0.96	29.00/0.90	29.40/0.91	27.44/0.87	27.80/0.88	26.46/0.84	26.76/0.85	25.78/0.82	26.05/0.83
Zebra	30.70/0.97	30.87/0.97	26.86/0.93	27.46/0.94	24.84/0.90	25.58/0.91	23.46/0.86	24.26/0.89	22.47/0.84	23.34/0.87
Workers	31.92/0.95	32.41/0.96	28.06/0.90	28.77/0.92	25.96/0.85	26.77/0.89	24.57/0.81	25.39/0.85	23.56/0.77	24.37/0.82
Swimmer	33.63/0.97	33.92/0.98	30.17/0.95	30.57/0.95	28.29/0.92	28.73/0.93	27.03/0.90	27.43/0.92	26.09/0.88	26.50/0.90
Rope	28.37/0.91	28.53/0.91	24.68/0.80	24.96/0.81	22.87/0.70	23.18/0.72	21.73/0.61	22.06/0.63	20.93/0.55	21.23/0.58
Corns	29.74/0.95	29.90/0.95	25.94/0.89	26.16/0.90	23.92/0.83	24.06/0.84	22.59/0.78	22.74/0.79	21.63/0.74	21.77/0.74
Avg.	30.87/0.95	31.27/0.96	27.22/0.90	27.78/0.91	25.33/0.86	25.95/0.88	24.11/0.82	24.73/0.84	23.23/0.79	23.84/0.81
t (s)	20.62	40.18	21.32	33.81	18.97	33.39	21.12	36.27	22.50	35.54

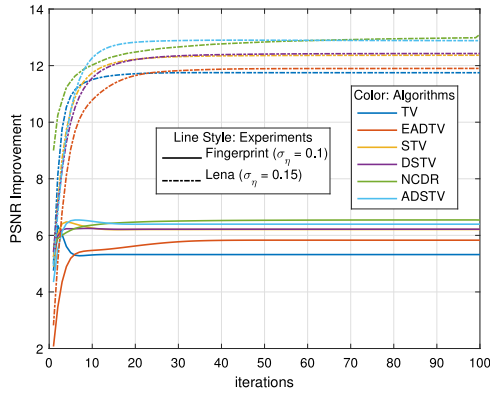


Fig. 8. PSNR improvement w.r.t. the noisy input across 100 iterations for all algorithms.

6. Discussion

In the previous section, we compared the proposed denoising framework with TV [1], EADTV [9,10], STV [14], DSTV [23], and NCDR [22] based denoising models. STV was holding the state-of-the-art records among the other TV-related local analysis-based priors. NCDR came up with a new paradigm regarding the sparsity of the corner points, in 2018. Although the NCDR puts plausible results out, its non-convexity and heuristic nature make it disadvantageous. Moreover, it is only applicable to the scalar-valued images. The qualitative results and SSIM measurements show the effectiveness of our ADSTV over NCDR in terms of restoration quality, while NCDR results in better PSNR. But when we consider the computational performances, NCDR machinery is around seven times slower than ADSTV. We believe that our framework provides a good balance between the restoration quality and the computational performance.

On the other hand, as mentioned in Sections 1 and 3.1, a closely related paper [11] was published as preprint (to be published in 2020) at the moment that we were studying on this paper. That work could not be involved in our experiments but, due to the common experiment performed on the same *Lena* image, we think it is worth talking over it here. The authors reported that the restored versions of *Lena* contaminated by Gaussian noise with $\sigma = \{0.05, 0.1, 0.15\}$ has the signal to noise ratio (SNR) around 17.5, 15 and, 13, respectively; whereas the corresponding results of ADSTV are 19.77, 16.84, and 15.07. We think this difference is due to the incorporation of structure tensor.

In order not to finish discussion without referring to today's learning based approaches on restoration, we want to touch upon briefly how close our results to those obtained by the popular feed-forward denoising convolutional neural networks (DnCNN) [35]. In Table 3, we

report the PSNR scores on the same grayscale denoising experiments conducted in Section 5. As might be expected, we observe that the results of DnCNN are 1 ± 0.2 dB better than ADSTV on average, except the cases where $\sigma_n = 0.25$. DnCNN's failure under this extensive amount of noise is due to the limited range of noise levels used to train the model, which was chosen as $[0, 0.22]$ in [35]. This dependency to the training set parameters is one of the downsides of the deep restoration approaches. In Fig. 9, one can see the visual outcomes of both algorithms for two different noise levels on the *Parrot* image. The performance drop of the DnCNN is visible in Fig. 9(e). Note that we preferred to use the publicly available PyTorch implementation⁴ of the testing codes for the blind version of DnCNN (DnCNN-B). Similar to the classification tasks, data-driven methods seems to predominate the model-driven ones in restoration tasks. They can themselves capture the underlying geometry within the data. Moreover, they can run very fast once trained. However, beside those advantages, they also have limitations such as the lack of the suitable ground truth images that would be used for training purpose, the difficulty of incorporating prior domain knowledge, and the strong dependency to the imaging parameters (e.g. noise level) of the training data, as can be seen from Table 3 and Fig. 9. To tackle with these limitations, new efforts try to involve the variational models in the learning-based methods [22,36,37]. Also as remarked in [38], the learning-based approaches comprise the variational approaches as special cases. These facts and the non-negligible drawbacks motivate the researchers to produce in both fields with the hope that their achievements might further be combined.

Finally, our ADSTV regularizer can be exploited in various restoration tasks, such as deblurring, inpainting, and destriping. ADSTV regularized model can easily be generalized to address such problems by using proximal splitting tools. However its performance highly depends on the performance of the directional parameter estimation procedure. The algorithm that estimates the directions in this work is suitable for the images corrupted by Gaussian noise. When it is applied to blurry images, edges are wrongly localized, therefore the directional parameters are wrongly estimated. When we apply it to images with missing pixels, it would not be able to estimate the underlying directional characteristics, especially when a large image portion is lost. When it comes to the removal of the stripe noise (a popular problem in remote sensing imagery, e.g., [39]), since our parameter estimator cannot distinguish the stripe component, it would be treated as a highly directional image component and promoted by the succeeding inversion process. A relevant research direction might be extending our approach to handle oblique stripe noise, which would require a new degradation model that takes stripes as a separate component into account. The penalization of the stripe component can be achieved by adding a DSTV term to the ADSTV regularized objective function, to

⁴ <https://github.com/SaoYan/DnCNN-PyTorch>

Table 3
PSNR Comparison of DnCNN and ADSTV on grayscale images.

		Monarch	Lena	Parrot	Barbara	Fingerp.	Shell	Feather	Avg.
$\sigma_\eta = 0.05$	DnCNN	33.05	35.12	33.99	33.04	31.25	29.80	28.44	32.10
	ADSTV	32.38	34.03	33.80	31.09	30.09	29.57	27.71	31.24
$\sigma_\eta = 0.1$	DnCNN	29.82	32.26	30.71	29.62	27.53	26.65	25.07	28.81
	ADSTV	28.90	31.15	30.49	27.34	26.48	26.01	23.37	27.68
$\sigma_\eta = 0.15$	DnCNN	28.05	30.43	29.08	27.60	25.44	24.92	23.17	26.96
	ADSTV	26.87	29.45	28.79	25.40	24.46	24.18	21.19	25.76
$\sigma_\eta = 0.2$	DnCNN	26.56	29.18	27.90	26.06	24.03	23.78	21.76	25.61
	ADSTV	25.61	28.29	27.67	23.98	23.12	22.92	19.46	24.44
$\sigma_\eta = 0.25$	DnCNN	23.97	24.89	25.08	23.07	22.04	21.80	20.01	22.98
	ADSTV	24.44	27.37	26.83	23.22	22.11	22.01	18.24	23.46

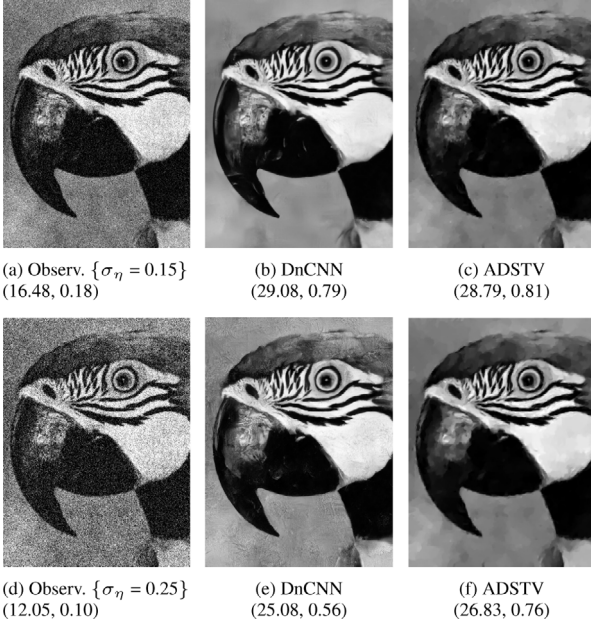


Fig. 9. The detail patches showing the denoising results for the noisy *Parrot* image with (a) $\sigma_\eta = 0.15$ and (b) $\sigma_\eta = 0.25$ by using (b) & (e) DnCNN and (c) & (f) ADSTV based denoising. The quantity pairs shown at the bottom are corresponding to the (PSNR, SSIM) values.

minimize the variations in a direction orthogonal to the stripes. Surely, the parameter estimation procedure should also be extended, maybe by categorizing the dominant one among the estimated orientations as the orientation of the stripes.

7. Conclusion

In this study, we have described a two-stage variational framework for denoising scalar- and vector-valued images. First stage is the pre-processor that estimates some directional parameters that will guide to the second stage, and the second stage is the one that performs denoising. Our preprocessor attempts to circumvent the chicken-and-egg problem of capturing local geometry from a noisy image to be used while denoise it, by employing the eigenvalues and the orthogonal eigenvectors of the structure tensor, and making assumptions on the piecewise-constancy of the underlying directional parameters. When it comes to the denoising stage, it encodes the problem as a convex optimization problem that involves the proposed adaptive direction-guided structure tensor total variation (ADSTV) as regularizer. Our ADSTV is designed by extending the structure tensor total variation (STV), so that it works biased in a certain direction varying at each point in the image domain. This approach results in improved restoration quality that has been verified by extensive experiments. We compared our method with the related denoising models involving various analysis-based priors.

We believe that the ADSTV regularizer can be used in a large spectrum of inverse imaging problems, even though the effectiveness of our parameter estimation process may change from problem to problem. As it is discussed in Section 6, one should work on the parameter estimation procedure to make it applicable for different problems, whereas the inversion step can easily be extended by using proximal splitting tools. These tools split the problem into smaller problems, one of which usually reduces to denoising. Therefore, another approach might be updating the directional descriptors from the estimated image just before the denoising iterations, at the cost of additional computational load. Furthermore, there might be appropriate application fields where the source image that is used to estimate the directional parameters differs from the target image to be restored (e.g., video restoration where the source and target frames are different, pansharpener where the chromatic image can be used as a source for the parameters). We plan to pursue these issues as future work.

CRedit authorship contribution statement

Ezgi Demircan-Tureyen: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Mustafa E. Kamasak:** Conceptualization, Supervision, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under 115R285.

Appendix

In order to find an upper bound for Lipschitz constants, we follow the derivation in [32] and adapt it to our formulation.

$$\begin{aligned}
 \|\nabla d(\mathbf{u}) - \nabla d(\mathbf{v})\| &= \tau \|\tilde{J}_K^{(\alpha, \theta)} P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)*} \mathbf{u}) - \tilde{J}_K^{(\alpha, \theta)} P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)*} \mathbf{v})\| \\
 &\leq \tau \|\tilde{J}_K^{(\alpha, \theta)}\| \|P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)*} \mathbf{u}) - P_C(\mathbf{g} - \tau \tilde{J}_K^{(\alpha, \theta)*} \mathbf{v})\| \\
 &\leq \tau \|\tilde{J}_K^{(\alpha, \theta)}\| \|\tau \tilde{J}_K^{(\alpha, \theta)*}(\mathbf{u} - \mathbf{v})\| \\
 &\leq \tau^2 \|\tilde{H}_{(\alpha, \theta)}\|^2 \|J_K\|^2 \|\mathbf{u} - \mathbf{v}\| \\
 &\leq \tau^2 \|\tilde{H}_{(\alpha, \theta)}\|^2 \|\nabla\|^2 \|T\|^2 \|\mathbf{u} - \mathbf{v}\|
 \end{aligned} \tag{34}$$

where $T = \sum_{l=1}^L (\mathcal{T}_l^* \mathcal{T}_l)$. Knowing from [32] that $\|\nabla\|^2 \leq 8$, from [14] that $\|T\|^2 \leq \sqrt{2}$, and further showing that $\|\tilde{H}_{(\alpha, \theta)}\|^2 \leq (\alpha^+)^2 + (\alpha^-)^2$, we come up with the spatially varying Lipschitz constants $\mathbf{L}[i] \leq 8\sqrt{2}\tau^2((\alpha^+)^2 + (\alpha^-)^2)$.

References

- [1] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1–4) (1992) 259–268.
- [2] T. Chan, A. Marquina, P. Mulet, High-order total variation-based image restoration, *SIAM J. Sci. Comput.* 22 (2) (2000) 503–516.
- [3] S. Esedoğlu, S.J. Osher, Decomposition of images by the anisotropic Rudin-Osher-Fatemi model, *Comm. Pure Appl. Math.* 57 (12) (2004) 1609–1626.
- [4] R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization, *IEEE Signal Process. Lett.* 14 (10) (2007) 707–710.
- [5] Y. Lou, T. Zeng, S. Osher, J. Xin, A weighted difference of anisotropic and isotropic total variation model for image processing, *SIAM J. Imaging Sci.* 8 (3) (2015) 1798–1823.
- [6] K. Bredies, K. Kunisch, T. Pock, Total generalized variation, *SIAM J. Imaging Sci.* 3 (3) (2010) 492–526.
- [7] S. Lefkimiatis, J.P. Ward, M. Unser, Hessian Schatten-norm regularization for linear inverse problems, *IEEE Trans. Image Process.* 22 (5) (2013) 1873–1888.
- [8] K. Papafitsoros, C.-B. Schönlieb, A combined first and second order variational approach for image reconstruction, *J. Math. Imaging Vision* 48 (2) (2014) 308–338.
- [9] I. Bayram, M.E. Kamasak, Directional total variation, *IEEE Signal Process. Lett.* 19 (12) (2012) 781–784.
- [10] H. Zhang, Y. Wang, Edge adaptive directional total variation, *J. Eng.* 2013 (11) (2013) 61–62.
- [11] Z.-F. Pang, H.-L. Zhang, S. Luo, T. Zeng, Image denoising based on the adaptive weighted tvp regularization, *Signal Process.* 167 (2020) 107325.
- [12] L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: Algorithm and performance evaluation, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (8) (1998) 777–789.
- [13] M. Grasmair, F. Lenzen, Anisotropic total variation filtering, *Appl. Math. Optim.* 62 (3) (2010) 323–339.
- [14] S. Lefkimiatis, A. Roussos, P. Maragos, M. Unser, Structure tensor total variation, *SIAM J. Imaging Sci.* 8 (2) (2015) 1090–1122.
- [15] P. Blongren, T.F. Chan, Color TV: Total variation methods for restoration of vector-valued images, *IEEE Trans. Image Process.* 7 (3) (1998) 304–309.
- [16] A. Buades, B. Coll, J.-M. Morel, Non-local means denoising, *Image Process. On Line* 1 (2011) 208–212.
- [17] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, *Multiscale Model. Simul.* 7 (3) (2008) 1005–1028.
- [18] S. Lefkimiatis, S. Osher, Nonlocal structure tensor functionals for image regularization, *IEEE Trans. Comput. Imaging* 1 (1) (2015) 16–29.
- [19] A. Chambolle, R.A. De Vore, N.-Y. Lee, B.J. Lucier, Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage, *IEEE Trans. Image Process.* 7 (3) (1998) 319–335.
- [20] M.A. Figueiredo, J.M. Bioucas-Dias, R.D. Nowak, Majorization–minimization algorithms for wavelet-based image restoration, *IEEE Trans. Image Process.* 16 (12) (2007) 2980–2991.
- [21] G. Kutyniok, M. Shahram, X. Zhuang, Shearlab: A rational design of a digital parabolic scaling algorithm, *SIAM J. Imaging Sci.* 5 (4) (2012) 1291–1332.
- [22] H. Liu, S. Tan, Image regularizations based on the sparsity of corner points, *IEEE Trans. Image Process.* 28 (1) (2018) 72–87.
- [23] E. Demircan-Tureyen, M.E. Kamasak, On the direction guidance in structure tensor total variation based denoising, in: *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2019, pp. 89–100.
- [24] G. Gilboa, N. Sochen, Y.Y. Zeevi, Variational denoising of partly textured images by spatially varying constraints, *IEEE Trans. Image Process.* 15 (8) (2006) 2281–2289.
- [25] M. Grasmair, Locally adaptive total variation regularization, in: *International Conference on Scale Space and Variational Methods in Computer Vision*, Springer, 2009, pp. 331–342.
- [26] Y. Dong, M. Hintermüller, Multi-scale total variation with automated regularization parameter selection for color image restoration, in: *International Conference on Scale Space and Variational Methods in Computer Vision*, Springer, 2009, pp. 271–281.
- [27] R. Bhatia, *Matrix Analysis*, Vol. 169, Springer Science & Business Media, 2013.
- [28] O. Merveille, B. Naegel, H. Talbot, L. Najman, N. Passat, 2D filtering of curvilinear structures by ranking the orientation responses of path operators (RORPO), 2017.
- [29] O. Merveille, H. Talbot, L. Najman, N. Passat, Curvilinear structure analysis by ranking the orientation responses of path operators, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2) (2017) 304–317.
- [30] M.A. Oliveira, N.J. Leite, A multiscale directional operator and morphological tools for reconnecting broken ridges in fingerprint images, *Pattern Recognit.* 41 (1) (2008) 367–377.
- [31] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (5) (2011) 898–916, <http://dx.doi.org/10.1109/TPAMI.2010.161>.
- [32] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, *IEEE Trans. Image Process.* 18 (11) (2009) 2419–2434.
- [33] A. Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vision* 20 (1–2) (2004) 89–97.
- [34] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* 2 (1) (2009) 183–202.
- [35] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, *IEEE Trans. Image Process.* 26 (7) (2017) 3142–3155.
- [36] S. Lefkimiatis, Universal denoising networks: A novel cnn architecture for image denoising, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3204–3213.
- [37] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.
- [38] M.T. McCann, K.H. Jin, M. Unser, Convolutional neural networks for inverse problems in imaging: A review, *IEEE Signal Process. Mag.* 34 (6) (2017) 85–95.
- [39] J.-L. Wang, T.-Z. Huang, T.-H. Ma, X.-L. Zhao, Y. Chen, A sheared low-rank model for oblique stripe removal, *Appl. Math. Comput.* 360 (2019) 167–180.