Learned Regularizers and Geometry for Image Denoising

Stacey Levine[1]
levines@duq.edu

Ryan Cecil[1]
cecilr@duq.edu

Marcelo Bertalmío[2]
marcelo.bertalmio@csic.es

[1] Department of Mathematics and
Computer Science
Duquesne University
Pittsburgh, PA, USA

[2] Instituto de Óptica
Spanish National Research Council
Madrid, Spain

Abstract

Recent frameworks for image denoising have demonstrated that it can be more productive to recover an image from a smoothed version of some geometric feature of the image rather than denoise the image directly. Improvements can be found both with respect to image quality metrics as well as the preservation of fine details. The challenge in working with this data is that mathematically sound mechanisms developed for handling natural image data do not necessarily apply, and this data itself can be quite ill behaved. In this work we learn both 'geometric' or nonlinear higher order features and corresponding regularizers. These approaches show improvement over recent model-based deep learning (DL) image denoising methods both with respect to image quality metrics as well as the preservation of fine features. Furthermore, the proposed approach for enhancing DL architectures by incorporating geometrically-inspired features is motivated by and has the potential to feed back into mathematically sound models for solving a variety of problems in image processing.

1 Introduction

Image denoising models have been foundational in a broad range of image processing tasks, including deblurring, superresolution, dejittering, and more. About a decade ago, image denoising had become so powerful that researchers began to study optimal denoising bounds [5, 16, 17], finding little room for improvement with respect to the root mean squared error (RMSE) over deterministic, patch based models for denoising an image directly. However, denoising results are still sub-optimal in applications where fine features must be preserved without introducing false data. Over the last decade, powerful GPUs have enabled supervised deep learning (DL) using massive amounts of data that have paved the way to outstanding denoising performance, pushing past these optimality bounds e.g. [6, 14, 15, 31]. A key challenge in working with learned approaches is the lack of predictability and interpretability of the models.

In a different direction, several image denoising approaches have been proposed that attempt to recover a denoised image from a smoothed version of some geometric feature of the image, including its unit normal vector field [18], level line curvature [3], or components in an orthonormal moving frame that encodes geometric information about the image [9]. While these approaches consistently improved over denoising the image directly using comparable approaches, the challenge in working with higher order geometrically inspired data is that mathematically sound mechanisms developed for handling natural image data do not necessarily apply, and this data itself can be quite ill behaved.

The crux of this work is to combine the power of DL approaches with the mathematical structure of geometrically motivated model based approaches to propose new image denoising models that take advantage of the best of both worlds. Specifically, the proposed approach *learns* both the geometric/nonlinear higher order features as well as corresponding regularizers that are motivated by and have the potential to feed back into mathematically sound models for image denoising. Ultimately this approach has the potential to serve as a feature preserving regularizer that can improve a variety of image processing problems either directly or as in e.g. [26, 30].

The remainder of this paper is as follows. Section 2 contains a discussion of the mathematical models and learning based algorithms that motivate the proposed approach, Section 3 introduces the proposed approach and includes analyses of the proposed architecture, Section 4 contains numerical results, and Section 5 concludes the paper. An Appendix contains details on the numerical implementation of the new approach.

2 Background

In the following, we will consider the degradation model of an image degraded by additive white Gaussian noise (AWGN). That is, we observe a noisy image $f = a + \eta$, where a is the true image and $\eta \sim \mathcal{N}(0, \sigma^2)$, and aim to recover an approximation for a.

2.1 Model based approaches for image denoising

In 1990, Perona and Malik [22] proposed denoising an image degraded by AWGN using the seminal edge-preserving anisotropic diffusion equation where $g(\cdot) \sim 0$ near likely edges,

$$\frac{\partial u}{\partial t} = div\,(g(|\nabla u|)\nabla u)\,,\ u(t=0) = f. \tag{1}$$

Around the same time, Rudin, Osher and Fatemi [27] proposed the equally pioneering denoising approach that seeks the solution of the minimization problem in the functions of bounded variation (BV)

$$\min_{u \in BV} \int |\nabla u| + \frac{\lambda}{2} \int (f-u)^2. \tag{2}$$

The flow of the Euler-Lagrange equation associated with the above minimization problem is the well-posed edge-preserving anisotropic diffusion equation which is closely related to (1)

$$\frac{\partial u}{\partial t} = div\left(\frac{\nabla u}{|\nabla u|}\right) + \lambda(f-u),\ u(t=0) = f. \tag{3}$$

Both (1) and (3) pair the linear divergence, *div*, and gradient ∇ operators, both of which can be discretized as convolution operators. These models also connect these linear operators through a nonlinear function $g(\nabla u)$. Interestingly, these are typical building blocks for convolutional neural networks (CNNs), discussed further in Section 2.3.

Both (1) and (3) were groundbreaking for their time in their ability to preserve sharp object boundaries throughout the noise removal process. On the flip side, when applied to images directly, these approaches suffer from artifacts arising in what should be smooth regions (this is often observed as 'staircasing,' where a noisy smooth region is mapped to a piecewise constant solution) and their inability to preserve fine scaled textures. However, they are still particularly appealing and still used as the basis for many new models for image processing given the wide reaching theory that enables their mathematical predictability.

2.2 Denoising framework using 'filtered' geometric features

In 2004, Lysaker, Osher and Tai proposed the following approach to image denoising. Instead of solving (2), to avoid staircasing, one can process the image surface more directly. Specifically, denoise the *unit normal field*, $\overrightarrow{\eta}(f) = \frac{\nabla f}{|\nabla f|}$, of the noisy image, f, by solving a constrained vectorial version of (2) to obtain $\mathcal{F}(\overrightarrow{\eta})$. Then the denoised image can be obtained via the following minimization problem which matches the orientation of the denoised unit normal vector field $\mathcal{F}(\overrightarrow{\eta})$ to the unit normal vector field of the denoised image, $\frac{\nabla u}{|\nabla u|}$,

$$\arg\min_{u \in BV} \int \left(|\nabla u| - \mathcal{F}(\overrightarrow{\eta}) \cdot \nabla u\right) + \lambda \int (f-u)^2. \tag{4}$$

This model is related to earlier inpainting work by Ballester et.al. 2002 [2], and notably led to Bregman iterations of Osher et.al. 2005 [21]. Alternate mathematically sound formulations for denoising the unit tangent vector field have also been proposed [11, 24].

In 2014, Bertalmío and Levine [3] proposed a general approach in which any image denoising method $\mathcal{F}$ intended for an observed noisy image f is instead applied to the noisy level line curvature, $\kappa(f) = \text{div}(\nabla f / |\nabla f|)$. Then a smoothed version of the curvature $\kappa_{\mathcal{F}} = \mathcal{F}(\kappa(f))$ is used to reconstruct a new image $\hat{f}_{\kappa_{\mathcal{F}}}$ whose curvature matches that of $\kappa_{\mathcal{F}}$ and whose average value along level lines matches that of f.

A simple, mathematically sound method for constructing the denoised image from its denoised curvature $\kappa_{\mathcal{F}}$ and a noisy observation $f = a + \eta$ is

$$\frac{du}{dt} = \underbrace{\kappa(u) - \kappa_{\mathcal{F}}}_{\text{preserves image 'geometry'}} + \underbrace{\lambda(f-u)}_{\text{preserves contrast along level lines}}, \quad u(t=0) = f_{in} \tag{5}$$

where $\lambda > 0$ depends only on the noise variance. As $t \to \infty$, $\frac{du}{dt} \to 0$, so $\kappa_{\mathcal{F}}$ becomes a good approximation for the curvature of the solution $\kappa(u)$ for which contrast is preserved along level lines. If one sets $f_{in} = \mathcal{F}(f)$, (5) leads to a kind of *boosting* algorithm [20, 25], where a denoised image $\mathcal{F}(f)$ is enhanced with a good approximation of the clean curvature, $\kappa_{\mathcal{F}}$. This is particularly appealing when $\mathcal{F}$ is a non-parametric approach which provides state of the art results with respect to image quality metrics, but does not naturally incorporate geometry, potentially resulting in loss of detail. Note also the close relationship between (3) and (5), the former of which can be written as

$$\frac{du}{dt} = \kappa(u) + \lambda(f-u), \quad u(t=0) = f. \tag{6}$$

Thus $\kappa(u)$ has a unique interpretation both as a *diffusion based regularizer* as well as *encoding critical geometry*.

The approaches in (4) and (5) are reasonable from several perspectives. An image can be reconstructed from its level sets (sets of pixels with the same intensity value), similar to how a three dimensional representation of surface altitude can be reconstructed from a topographic map, up to a change in contrast. Level sets are also useful for representing images, since the most important features within an image are the boundaries between objects which are typically level sets of the surface intensity. Furthermore, information describing contours, in this case level curves, is most concentrated at locations of highest curvature [1].

In addition to these geometric considerations, theoretical analyses [3] demonstrated that at likely edges, the point signal-to-noise ratio (PSNR) of these geometric features satisfy $PSNR(\kappa(f),\kappa(a)) > PSNR(\vec{\eta}(f),\vec{\eta}(a)) > PSNR(f,a)$, indicating that these geometric features might be more productive to denoise than the original image. Experimental results in [3] confirmed this, consistently showing an increase in PSNR and structured similarity index (SSIM) over denoising the image directly for both variational and discriminative denoisers $\mathcal{F}$ such as [4, 8, 11, 21, 27]. It is interesting to note that this framework falls outside of the ones studied in the most recent denoising optimality bounds [16, 17].

However, while the PSNR of $\kappa(f)$ is higher than that of f at locations where $|\nabla f|$ is sufficiently large, that is, locations that likely contain image detail such as edges and textures [3], finding an appropriate regularizer/denoiser intended directly for $\kappa(f)$ is not so straightforward. The underlying probability distribution for curvature patches is unknown, and it is not clear there is a good 'smoothness' space to work in. Thus at the moment, Bayesian approaches are guesstimates at best. Still, many denoising algorithms intended for natural image data carried over quite nicely to the unit normal and tangent vector fields as well as curvature data, with good results, so the potential for improvement is great if a more tailored regularizer could be used. Furthermore, with the powerful learning frameworks that have exploded in recent years, there is the potential to further learn this nonlinear geometric information that can aid in better reconstructions.

2.3 Mathematically motivated learning based approach for image denoising

The last five years have seen the emergence of deep learning (DL) architectures for solving inverse problems in imaging [19], including denoising, deblurring, super resolution, and related problems. One of the first such approaches, proposed by Chen and Pock [6], was a DL image denoising model based on successful reaction-diffusion equations which are inherently related to (1) and (6). They proposed the Trainable Nonlinear Reaction Diffusion (TNRD) model whose learned diffusion process (in the case of the AWGN) is

$$u_t = u_{t-1} - \left(\underbrace{\mathcal{G}_L^t(u_{t-1})}_{\text{diffusion term}} + \underbrace{\lambda^t(u_{t-1} - f)}_{\text{reaction term}} \right),\; u_0 = f,\; t = 1,..,T \tag{7}$$

with

$$\mathcal{G}_L^t(\cdot) = \sum_{i=1}^{N_k} \overline{k}_i^t * \phi_i^t(k_i^t * (\cdot)) \tag{8}$$

Here, k_i^t are learned kernels of some fixed size $m \times m$, $\overline{k}_i^t$ is the rotation of k_i by 180 degrees, $N_k = m^2 - 1$ is the number of filters at each iteration, ϕ_i^t are learned nonlinear functions parametrized by their coefficients in an appropriately defined basis (the authors in [6] used

64 radial basis functions), $\lambda^t > 0$, and T is a predetermined number of iterations. The parameters $\Theta = (k_i^t, \phi_i^t, \lambda^t)_{t=1}^T$ are learned from pairs of clean and noisy images (a^s, f^s) and updated by minimizing the loss function

$$L(\Theta) = \sum_{i=1}^{S} \frac{1}{2} ||u_T^s(\Theta) - a^s||_2^2 \tag{9}$$

where $u_T^s(\Theta)$ is the solution of (7) with initial data $f^s = a^s + \eta^s$ for each $s = 1, .., S$.

The reaction-diffusion equation (7) is similar to the forward problem in a typical convolutional neural network (CNN), but they are not precisely the same. CNNs do not typically pair k and $\bar{k}$ as in (7), which is more typical of partial differential equations. They also do not incorporate a reaction term unless it is a recurrent network [10], and the influence functions, ϕ, are typically fixed rather than learned. The solution of (7) is not a minimizer of an energy functional as it is intentionally run for a fixed number of iterations, T, which makes the problem tractable. The fixed iterations is not a problem since (7) is balanced with minimizing the loss function (9). The reaction term, $\lambda^t(u_{t-1} - f)$, is a smooth data fidelity term, also found in (6), which can readily handle the case of Gaussian denoising. The relationship of the forward problem (7) with nonlinear reaction-diffusion equations, particularly interpreted as a discrete generalized version of (1) and (3), makes this approach appealing from a mathematical perspective as the learning is both physically, mathematically, and geometrically motivated. The novelty in this approach is that it incorporates the power of DL, while potentially opening the door to establishing theoretical foundations as well as physical and geometric interpretations.

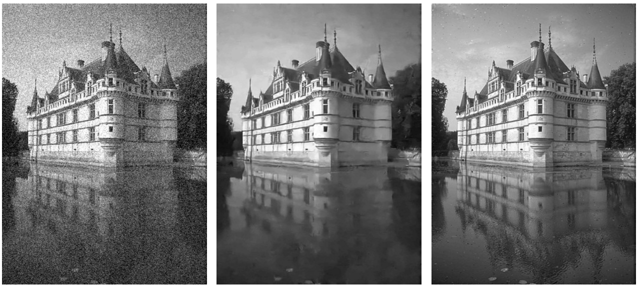

Figure 1: **Left**: Noisy image f with $\sigma = 25$; **Middle**: solution of (7), TNRD(f), with $u(t=0) = f$, PSNR=29.64, SSIM=0.8496, **Right**: solution of (5) with $\lambda = 0$, an oracle $\kappa_{\mathcal{F}} = \kappa(a)$ and $f_{in} = TNRD(f)$, PSNR=33.69, SSIM=0.9289.

TNRD benefitted from allowing the parameters to change over time making it different than a typical diffusion equation. The performance showed great promise for image denoising and it generalized to handle a range of image degradation tasks. Related model based DL approaches, including variational networks [14] and the Total Deep Variation model [15], produce even higher quality numerical results as well as provide additional interpretable mathematical structure. Should a good estimate for the image geometry such as level line curvature be attainable, the potential for improvement in combining one of these model based DL approaches with (5) is great. Fig. 1 contains an example of a noisy image and a TNRD [6] denoised image, the first and second images in the sequence. Using $u(t=0) = TNRD(f)$ (the denoised output of (7)) as an initial condition for (5), if the true clean level line curvature $\kappa(a)$ were accessible, the image is recovered almost perfectly, the result residing in the third image in the sequence. While an oracle $\kappa(a)$ is unattainable, a learned denoiser $\mathcal{F}_L$ might allow us to minimize the difference between $\kappa(a)$ and $\mathcal{F}_L(\kappa(f))$,

Table 1: PSNR comparing TNRD with the $\mathcal{GF}$-TNRD

	$\sigma = 15$				$\sigma = 25$				$\sigma = 50$			
stage 3	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9
TNRD	30.8	31.2	31.35	31.44	28.19	28.65	28.83	28.93	25.21	25.66	25.86	25.98
$\mathcal{F}_L$3x3	30.85	31.24	31.33	31.41	28.32	28.7	28.85	28.92	25.29	25.74	25.9	26.01
$\mathcal{F}_L$5x5	31.02	31.28	31.39	31.35	28.54	28.73	28.88	28.92	25.49	25.74	25.91	26.03
$\mathcal{F}_L$7x7	31.24	31.33	31.43	31.47	28.74	28.75	28.89	28.95	25.7	25.78	25.92	26.02
$\mathcal{F}_L$9x9	31.32	31.37	31.45	31.44	28.83	28.84	28.91	28.99	25.82	25.93	25.96	26.02
stage 5	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9
TNRD	30.92	31.29	31.42	31.5	28.36	28.75	28.92	29	25.42	25.77	25.97	26.05
$\mathcal{F}_L$3x3	31	31.33	31.45	31.52	28.44	28.8	28.95	28.99	25.47	25.83	26.01	26.11
$\mathcal{F}_L$5x5	31.16	31.35	31.47	31.5	28.62	28.82	28.97	29.04	25.65	25.85	26.02	26.08
$\mathcal{F}_L$7x7	31.29	31.42	31.5	31.5	28.75	28.89	28.98	29.04	25.78	25.92	26.02	26.1
$\mathcal{F}_L$9x9	31.39	31.45	31.5	31.54	28.87	28.92	29.02	29.05	25.93	26	26.03	26.1
stage 8	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9	$\mathcal{G}_L$ 3x3	$\mathcal{G}_L$ 5x5	$\mathcal{G}_L$ 7x7	$\mathcal{G}_L$ 9x9
TNRD	31.02	31.35	31.47	31.54	28.49	28.83	28.97	29.04	25.54	25.86	26.03	26.13
$\mathcal{F}_L$3x3	31.11	31.39	31.5	31.54	28.55	28.87	29.01	29.08	25.6	25.93	26.06	26.15
$\mathcal{F}_L$5x5	31.24	31.41	31.52	31.51	28.69	28.89	29.03	29.07	25.68	25.93	26.08	26.16
$\mathcal{F}_L$7x7	31.34	31.46	31.54	31.57	28.83	28.94	29.03	29.1	25.82	25.98	26.09	26.16
$\mathcal{F}_L$9x9	31.44	31.5	31.56	31.55	28.91	28.99	29.06	29.1	25.93	26.05	26.09	26.16

Table 2: Performance Comparison with State of the Art Models from [15]

Data Set	σ	BM3D [illegible]	TNRD from [illegible]	GF-TNRD (10)	DnCNN [illegible]	FFDNet [illegible]	N^{3}Net [illegible]	FOCNet [illegible]	TDV^3 [illegible]
Set 12	15	32.37	32.50	32.64	32.86	32.75	-	33.07	33.01
	25	29.97	30.05	30.21	30.44	30.43	30.55	30.73	30.66
	50	26.72	26.82	27.03	27.18	27.32	27.43	27.68	27.59
BSDS68	15	31.08	31.42	31.54	31.73	31.63	-	31.83	31.82
	25	28.57	28.92	29.05	29.23	29.19	29.30	29.38	29.37
	50	25.60	25.97	26.11	26.23	26.29	26.39	26.50	26.45
Urban100	15	32.34	31.98	32.22	32.67	32.43	-	33.15	32.87
	25	29.70	29.29	29.50	29.97	29.92	30.19	30.64	30.38
	50	25.94	25.71	25.97	26.28	26.52	26.82	27.40	27.04
# of parameters			26,645	89,605	555,200	484,800	705,895	53,513,120	427,330

more-so than model based approaches that were designed to handle natural image data. This is a key piece of the motivation for the work proposed in Section 3.

3 Structured learned denoisers and higher order features

In this work we propose a new approach for image denoising which combines the power of DL with the preservation of fine features that can be attained when incorporating image geometry. Specifically, we infuse learned denoisers *and* learned 'geometry' into (5). For the purpose of consistency and simplicity, we employ the Trainable Nonlinear Reaction Diffusion (TNRD) model of Chen and Pock [6] as a basis for comparison as well as a basic building block, given its innate connections to anisotropic diffusion (1) and (6) as well as more general reaction-diffusion equations. The intention is to conceptually demonstrate how this novel use of image geometry combined with the outstanding performance of learned models for directly denoising an image has the potential to give the best of both worlds.

To this end, as a natural extension of the curvature denoising in (5), we propose minimizing the loss function (9) to learn parameters for the diffusion model (10), which for simplicity we will call $\mathcal{GF}$-TNRD, written as

$$u_t = u_{t-1} - \left(\mathcal{G}_L^t(u_{t-1}) - \mathcal{F}_L^t(\mathcal{G}_L^t(f)) + \lambda^t(u_{t-1} - f)\right),\ u_0 = f,\ t = 1,..,T. \tag{10}$$

Note that (10) not only learns a geometrically inspired regularizer $\mathcal{G}_L^t$ (8), but it also learns a denoiser $\mathcal{F}_L^t$ for $\mathcal{G}_L^t(f)$, where

$$\mathcal{F}_L^t(\cdot) = \sum_{j=1}^{N_a} \overline{a}_j^t * \varphi_j^t(a_j^t * (\cdot)). \tag{11}$$

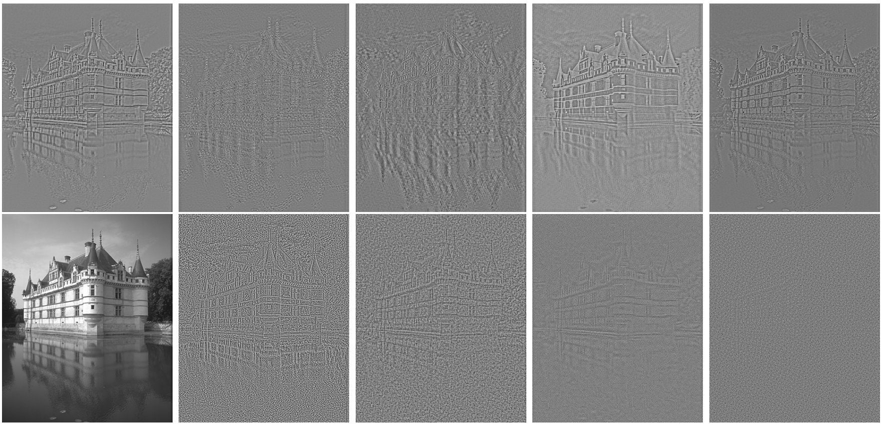

Figure 2: **Row 1**: $\mathcal{G}_L^t(a)$ for $t = 1,2,3,4,5$. Parameters learned for (10) with $T = 5$ and 80 9×9 filters for both $\mathcal{G}_L$ and $\mathcal{F}_L$. **Row 2**: clean image a followed by its level line curvature $\kappa(a)$. The last three images are the noisy geometry in stage 1, $\mathcal{G}_L^1(f)$, the denoised geometry in stage 1, $\mathcal{F}_L^1(\mathcal{G}_L^1(f))$, and the residual, $\mathcal{G}_L(f) - \mathcal{F}_L^1(\mathcal{G}_L^1(f))$

Thus, similar to κ in (5), $\mathcal{G}_L^t$ has a unique interpretation as both a *diffusion based regularizer* as well as *encoding geometry*. In fact, a discretized version of (5) can be considered a special case of (10) with $\mathcal{F} = TV$, $k_i^t = a_i^t = \nabla$, $\overline{k}_i^t = \overline{a}_j^t = -\text{div}$ and $\phi = \varphi = \rho'$ with $\rho(z) = |z|$. In [3] it was shown that this approach improved over total variation based denoising [27], Bregman iterations [21] and denoising unit normals [18] both with respect to image quality metrics as well as the preservation of fine features, thus optimizing these operators has the potential for further improvement. This should also serve to enhance a DL model such as TNRD since it ensures that higher order features, $\mathcal{G}_L^t(f)$, learned directly from f, are preserved; in a sense, this adds to the usual data fidelity term (modeling the additive noise degradation model) a second fidelity to the higher order, nonlinear, potentially geometric, image information.

Denoising effect of $\mathcal{F}_L^t$: The first row of Figure 2 contains learned 'geometric' regularizers $\mathcal{G}_L^t(a)$ for $t = 1,..,5$ applied to a clean image a obtained from (10). Similarities and complementary details with the level line curvature $\kappa(a)$, the second image in row 2, can be observed. To visualize the denoising effect of $\mathcal{F}$, the last three images in row 2 contain the noisy $\mathcal{G}_L^1(f)$ followed by the denoised $\mathcal{F}_L^1(\mathcal{G}_L^1(f))$ and finally the residual $\mathcal{G}_L^1(f) - \mathcal{F}_L^1(\mathcal{G}_L^1(f))$ which appears like random noise, indicating that $\mathcal{F}$ does in fact act as a denoiser. We tested this hypothesis over different test images using a range of model parameters and found that particularly after the first stage, the root mean squared error (RMSE) generally decreased after applying $\mathcal{F}$; that is, RMSE($\mathcal{F}_L^t(\mathcal{G}_L^t(f)), \mathcal{G}_L^t(a)$) < RMSE($\mathcal{G}_L^t(f), \mathcal{G}_L^t(a)$) for $t \geq 2$.

$\mathcal{F}_L^t$ filters -vs- $\mathcal{G}_L^t$ filters: It is interesting to visualize the learned filters for the geometric regularizers/geometries $\mathcal{G}_L^t$ and their denoisers $\mathcal{F}_L^t$, as they provide further insights into the proposed $\mathcal{GF}$-TNRD framework. Specifically, $\mathcal{G}_L^t$ filters often appear to be computing derivatives, which aligns with the idea that the $\mathcal{G}_L^t$ filters approximate features that are akin to curvature. On the other hand, many $\mathcal{F}_L^t$ filters appear to be computing averages along edges, in the center, or around the perimeter of the patch, operators which indicate denoising. Fig. 3 shows some example $\mathcal{G}_L^t$ filters in the top row and $\mathcal{F}_L^t$ filters in the bottom.

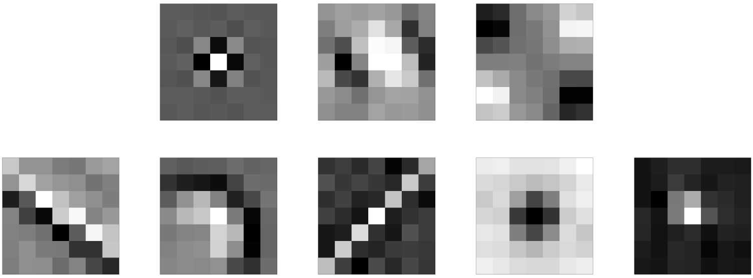

Figure 3: **Row 1**: sample $\mathcal{G}_L^t$ filters appear to compute derivatives, indicating curvature approximation. **Row 2**: sample $\mathcal{F}_L^t$ filters appear to compute local averages, indicating denoising.

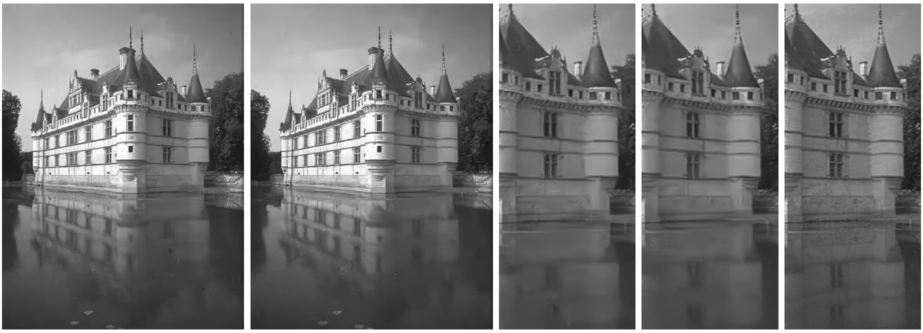

Figure 4: Solutions of (7) and (10) with $\sigma = 15$. (**L** to **R**): result of (7), result of (10) which retains more details in the windows and the water reflection with less artifacts in the sky, close up of (7) result, close up of (10) result, close up of the true image. Results for both (7) and (10) use $N_k = 80$ with 9×9 filters in (8) for $\mathcal{G}_L$ and $N_a = 8$ with 3×3 filters for $\mathcal{F}_L$.

4 Numerical Results

In what follows we compare $\mathcal{GF}$-TNRD with TNRD as an example of how geometric features can be carefully incorporated into a learned model for image denoising. Table 1 contains the average PSNR values of the $\mathcal{GF}$-TNRD model applied to the standard im68 database [28] for various filter sizes for both $\mathcal{G}_L$ and $\mathcal{F}_L$ computed for stages T=3,5,8. Ten noise samples were added to each image and the average output PSNR value for each image was computed. The PSNR reported in Table 1 is the average across the im68 database.

Note that for any given fixed filter size used for $\mathcal{G}_L$ in (7), a denoiser using a filter size $k \times k$ equal to or greater than $k \times k$ for $\mathcal{F}_L$ in $\mathcal{GF}$-TNRD always yields improved results over TNRD. The jump in PSNR begins modestly on the order of 0.05-0.1 but is much higher in the case of smaller filters for $\mathcal{G}$. In fact, for all models that showed any increase in PSNR, this increase was consistent across the entire database. Conducting a one-sided paired t-test

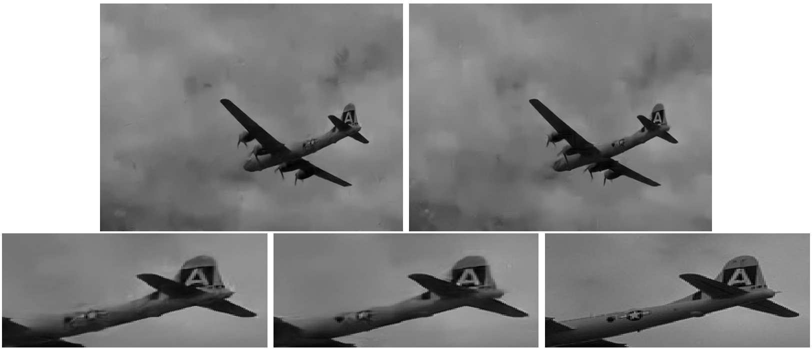

Figure 5: **Row 1**: **Left**: result of (7), **Right**: result of (10) which better preserves details on the plane body and tail with fewer artifacts in the sky. **Row 2**: close up of (7) result, close up of (10) result, close up of the true image. Same model parameters as Fig. 4 with $\sigma = 25$.

indicated that the increase in PSNR($\mathcal{GF}$-TNRD(f)) over PSNR(TNRD(f)) across the im68 database is statistically significant. SSIM showed similar trends.

It should also be noted that this increase in PSNR is not due simply to an increased number of parameters. In fact, note that filters of size 3×3 for $\mathcal{F}_L$ paired with 9×9 for $\mathcal{G}_L$ in $\mathcal{GF}$-TNRD consistently show a significant improvement over the reverse, 9×9 for $\mathcal{F}_L$ paired with 3×3 for $\mathcal{G}_L$ (see Table 1).

In addition to the increase in image quality metrics, the preservation of fine features can be observed in Figures 4 and 5. Specifically, the solution of $\mathcal{GF}$-TNRD (10) for the castle image retains more details in the windows and the water reflection, with less artifacts in the sky. The details on the body and the tail of the plane are also better preserved with (10) with fewer background artifacts (both of these examples used 9×9 filters for $\mathcal{G}_L$).

Table 2 compares the $\mathcal{GF}$-TNRD model with state of the art denoisers as reported in [15], including the number of parameters for each model. The $\mathcal{GF}$-TNRD model included in the chart uses $T = 5$ stages with 80 9×9 filters for both $\mathcal{G}_L$ and $\mathcal{F}_L$, with the number of parameters computed by $T * (1 + 31 * 80 + 9^2 * 80 + 31 * 80 + 9^2 * 80)$. In evaluating this table, we note that the intent of the proposed approach is to demonstrate how model based DL denoising can be improved both with respect to image quality metrics as well as the preservation of fine details by including a mechanism for enhancing local geometry/higher order features, such as the cascaded term $\mathcal{F}_L(\mathcal{G}_L(f))$ in (10). A similar term could be included in the frameworks in the state of the art approaches in [14] and [15] which we expect would further improve performance; this is the topic of future work. Still, the improvement is notable given the number of parameters required to obtain this improvement over TNRD is approximately fourfold, whereas the model parameters generally increase 20-40 times to achieve state of the art performance. The results are competitive with and almost uniformly show significant improvement over BM3D, which is still one of the top model-based approaches.

5 Conclusion

The purpose of this work is to demonstrate the value added by carefully incorporating denoised, geometrically-inspired features into model-based DL image denoising approaches.

The results show promise, as demonstrated by the performance of the $\mathcal{G}F$-TNRD model compared to TNRD, both with respect to image quality metrics as well as its ability to preserve fine features. The TNRD model was used as a prototype for this type of approach given the connections with (6) paired with the success of the curvature-based denoising (5). The mathematical basis for this added structure makes this type of approach ripe for physical and mathematical interpretation. In future work we will extend this general framework of incorporating geometry into more state of the art, model based DL denoising approaches, enabling a more in-depth analysis. Given denoising is a foundational tool in most image processing operations, including de-blurring, super-resolution, deblocking and others, this work will be further extended to handle more general inverse problems in imaging, including those with different types of noise as well as those with linear degradations.

Acknowledgements

The first and second author acknowledge support from the National Science Foundation DMS-1821342. The third author is supported by the H2020 Programme of the EU (project AdMiRe), and by the Spanish government and FEDER Fund (ref. PGC2018-099651-B-I00).

Appendix: Training Details

Implementation: The results reported here for both TNRD and $\mathcal{G}F$-TNRD were generated using Keras-Tensorflow [7] with a CUDA implementation of radial basis functions written for the denoising variational network in [14]. Both the TNRD and $\mathcal{G}F$-TNRD models were trained on 40x40 patches that were sampled, cropped, and transformed from the Train400 dataset used in [6, 28, 31]. The test metric performance did not show significant improvement with a sample complexity of more than 400 training pairs. Similar to [6, 14], the kernels were constrained to have zero mean and L^2 norm bounded by 1. Kernel weights were initialized using an 'orthogonal' initialization, the weights for the radial basis functions were initialized linearly, and all other scalar weights were initialized as 0.1. A batch size of 128 and the Adam optimizer [13] with an initial learning rate of 0.001 and Keras's Exponential Decay Scheduler were used to update the weights during training. Training TNRD for 16 epochs was found to achieve performance on par with the original implementations in [6, 14]. $\mathcal{G}F$-TNRD was trained for 32 epochs, and then for an additional 4 epochs using a cyclical learning rate [29] to further adjust the weights.

Training and Inference Times: Using an Nvidia RTX 3090 with Keras-Tensorflow, the training times for $\mathcal{GF}$-TNRD with $T = 5$ iterations ranged from 1068 seconds when using 8 3×3 filters for both $\mathcal{G}$ and $\mathcal{F}$ to 14501 seconds when training the model with 80 9×9 filters for both $\mathcal{G}$ and $\mathcal{F}$. In contrast, training for our implementation of the TNRD model ranged from 197 seconds for 8 3×3 filters for $\mathcal{G}$ to 2182 seconds for 80 9×9 filters for $\mathcal{G}$.

Once trained, the inference times of both models are comparable. The $\mathcal{GF}$-TNRD model with $T = 5$ stages and both $\mathcal{G}$ and $\mathcal{F}$ using 8 3×3 filters, 24 5×5 filters, 48 7×7 filters, and 80 9×9 filters has CPU inference times for a single image of 1.68, 4.60, 9.98, and 19.26 seconds respectively. The same parameters for TNRD processes a single image in 1.33, 3.67, 7.95 and 15.62 seconds. On the RTX 3090, all inference times for both models across all filter sizes range between 1.85-1.99 seconds.

References

[1] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

[2] Coloma Ballester, M. Bertalmío, V. Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process.*, 10(8):1200–1211, 2001.

[3] Marcelo Bertalmío and Stacey Levine. Denoising an image by denoising its curvature image. *SIAM J. Imaging Sci.*, 7(1):187–211, 2014. ISSN 1936-4954. doi: 10.1137/120901246. URL http://dx.doi.org/10.1137/120901246.

[4] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4(2):490–530 (electronic), 2005. ISSN 1540-3459.

[5] Priyam Chatterjee and Peyman Milanfar. Is denoising dead? *IEEE Transactions on Image Processing*, 19(4):895–911, 2010.

[6] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, June 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2596743.

[7] François Chollet et al. Keras. https://keras.io, 2015.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16 (8):2080–2095, 2007.

[9] Gabriela Ghimpeteanu, Thomas Batard, Marcelo Bertalmío, and Stacey Levine. A decomposition framework for image denoising algorithms. *IEEE Trans. Image Processing*, 25(1):388–399, 2016. doi: 10.1109/TIP.2015.2498413. URL http://dx.doi.org/10.1109/TIP.2015.2498413.

[10] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 545–552, USA, 2008. Curran Associates Inc. ISBN 978-1-6056-0-949-2. URL http://dl.acm.org/citation.cfm?id=2981780.2981848.

[11] Jooyoung Hahn, Xue-Cheng Tai, Sofia Borok, and Alfred Marcel Bruckstein. Orientation-matching minimization for image denoising and inpainting. *Int. J. Comput. Vis.*, 92(3):308–324, 2011. ISSN 0920-5691. doi: 10.1007/s11263-010-0371-5. URL http://dx.doi.org/10.1007/s11263-010-0371-5.

[12] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6047–6056, 2019. doi: 10.1109/CVPR.2019.00621.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[14] Erich Kobler, Teresa Klatzer, Kerstin Hammernik, and Thomas Pock. Variational networks: Connecting variational methods and deep learning. In Volker Roth and Thomas Vetter, editors, *Pattern Recognition - 39th German Conference, GCPR 2017, Basel, Switzerland, September 12-15, 2017, Proceedings*, volume 10496 of *Lecture Notes in Computer Science*, pages 281–293. Springer, 2017. ISBN 978-3-319-66708-9. doi: 10.1007/978-3-319-66709-6_23. URL https://doi.org/10.1007/978-3-319-66709-6_23.

[15] Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas Pock. Total deep variation for linear inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[17] A. Levin, B. Nadler, F. Durand, and W.T. Freeman. Patch complexity, finite pixel correlations and optimal denoising. Technical report, MIT - Computer Science and Artificial Intelligence Laboratory, 2012.

[18] Marius Lysaker, Stan Osher, and Xue-Cheng Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Process.*, 13(10):1345–1357, 2004. ISSN 1057-7149.

[19] M. T. McCann, K. H. Jin, and M. Unser. Convolutional Neural Networks for Inverse Problems in Imaging: A Review. *IEEE Signal Processing Magazine*, 34:85–95, November 2017. doi: 10.1109/MSP.2017.2739299.

[20] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, Jan 2013. ISSN 1053-5888. doi: 10.1109/MSP.2011.2179329.

[21] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.*, 4(2):460–489 (electronic), 2005. ISSN 1540-3459.

[22] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[23] Tobias Plötz and S. Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018.

[24] Talal Rahman, Xue-Cheng Tai, and Stanley Osher. A tv-stokes denoising algorithm. In Fiorella Sgallari, Almerico Murli, and Nikos Paragios, editors, *SSVM*, volume 4485 of *Lecture Notes in Computer Science*, pages 473–483. Springer, 2007. ISBN 978-3-540-72822-1.

[25] Yaniv Romano and Michael Elad. Boosting of image denoising algorithms. *SIAM J. Imaging Sciences*, 8(2):1187–1219, 2015. doi: 10.1137/140990978. URL http://dx.doi.org/10.1137/140990978.

[26] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017. doi: 10.1137/16M1102884. URL https://doi.org/10.1137/16M1102884.

[27] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[28] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781, June 2014. doi: 10.1109/CVPR.2014.349.

[29] Leslie N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017. doi: 10.1109/WACV.2017.58.

[30] Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *Proceedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 945–948, Austin, TX, USA, December 2013. doi: 10.1109/GlobalSIP.2013.6737048.

[31] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017. ISSN 1057-7149. doi: 10.1109/TIP.2017.2662206.

[32] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, PP, 10 2017. doi: 10.1109/TIP.2018.2839891.