PDE-based Image Denoising

Mansur Shakipov
Department of Mathematics
University of Maryland
College Park, MD, USA
shakipov@umd.edu

***Abstract*—Image denoising is an essential part of image processing which entails ubiquitous applications. There are different philosophies when it comes to tackling this problem. One of the more mathematically backed-up approaches is based on Partial Differential Equations (PDEs). This paper is going to give an overview on some of the most important PDE-based methods with the visual comparison of the results.**

***Index Terms*—image processing, image denoising, partial differential equations, Tikhonov regularization, total variatioon minimization, Perona-Malik diffusion, anisotropic diffusion**

I. Introduction

With an ever increasing digitization of the modern world, the demand for "high-quality" images is higher than ever. While the techniques of image acquisition have become not only more sophisticated but also more ubiquitous, nearly all the images get inevitably degraded by noise. Removing noise without deteriorating image features (i.e. edges) is a highly nontrivial problem in image processing. The applications of image denoising are vast: from achieving purely aesthetic goals in fine-art photography to resolving the first image of a black hole [1]. These factors, together with many others, lead to image denoising being an active area of research for many decades.

Many different approaches have been introduced to tackle this problem. The earliest approaches employed filtering in spatial and frequency domains. With the development of the theory of partial differential equations (PDE) and their corresponding numerical methods, PDE-based approaches were formulated. With the advent of powerful multi-core CPUs and GPUs, Neural Network-based algorithms have become feasible and highly successful.

Although PDE-based methods may not be the state of the art anymore, they are still extremely important to study because of the mathematical rigor behind those techniques. This paper will give a short introduction to some of the most fundamental PDE-based image denoising methods, including the Heat kernel [2], [3], Tikhonov regularization [2], [4], [5], Total Variation (TV) minimization [6], [7], Perona-Malik diffusion [8], and a more complex model like Fourth-order You-Kaveh filter [9].

In mathematical terms, the problem of image denoising can be formulated in the following way. Let $I \in \mathbb{R}^{n\times m}$ be an m-by-n clean image, $n \in \mathbb{R}^{n\times m}$ be additive Gaussian noise. Then the noisy image is $I_n = I + n$. The problem then reads

$$\text{Given } I_n, \text{ recover } I. \tag{1}$$

The difficulty of solving (1) comes from the fact that we want to (a) preserve edges, (b) keep flat areas smooth, and (c) do not add new artifacts [10]. (1) is an example of an inverse problem, which does not in general admit a unique solution, so we want to construct a procedure that produces a good estimate of the original clean image.

II. PDE-based formulations of the problem

One of the first approaches in image denoising are based on filtering. While linear filters do hit points (b,c), they fail to recognize edges and thus fail to satisfy (c). This realization created the need for better algorithms, and this is where PDE-based models came into place.

PDE, short for Partial Differential Equations, are equations that involve partial derivatives of a function to be found. They are great at describing physical phenomena, like, for instance, diffusion or wave propagation, though they do account for many other phenomena [3].

One of the most striking facts about PDEs is that they are closely related to energy minimization problems via Euler-Lagrange equations. This allows, depending upon a situation, to formulate the same problem in two different ways, which gives a lot of flexibity, as we will see shortly.

A. The Heat Kernel

The heat (or diffusion) equation is one of the most fundamental PDEs.

$$\begin{cases} u_t - \Delta u = f & \Omega \times \{t > 0\} \\ u = u_0 & \Omega \times \{t = 0\} \\ au + b\frac{\partial u}{\partial n} = g & \partial\Omega \times \{t > 0\} \end{cases} \tag{2}$$

The heat equation in the general form is given by (2). Here, $\Omega \subset \mathbb{R}^n$ is the spatial domain of the problem, $\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2}$ is the Laplace operator, $\frac{\partial}{\partial n}$ is the normal derivative at the boundary. By setting $a = 0$ we get a pure Neumann problem, while by setting $b = 0$ we obtain a pure Dirichlet problem. When neither a nor b are zero, this problem is called the Robin problem [3].

For the purposes of this project, we need to impose several assumptions on (2). First, we want the denoised image to converse its "mass", i.e. $\int_\Omega u = \int_\Omega u_n$. The proper boundary condition for this assumption is given by the *homogeneous Neumann condition*, i.e. $\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$. This means that $g = 0$ in equation (2). Note that Neumann problems have compatibility condition $\int_{\partial\Omega} g = \int_{\Omega} f$ which means that f should have vanishing mean. For simplicity, set $f = 0$. Then, the corresponding heat equation has the form given in (3), where u_0 is the noisy image, and u is the denoised image.

$$\begin{cases} u_t = \Delta u & \Omega \times \{t > 0\} \\ u = u_0 & \Omega \times \{t = 0\} \\ \frac{\partial u}{\partial n} = 0 & \partial\Omega \times \{t > 0\} \end{cases} \tag{3}$$

The solution to (3) is given by

$$u(x,t) = G_{\sqrt{2t}} * u_0(x) \tag{4}$$

where $*$ denotes convolution, and $G_{\sqrt{2t}}$ is the Gaussian with mean zero and standard deviation $\sigma = \sqrt{2t}$. This solution is unique provided $u \in \mathcal{O}(\exp(a|x|^2)), a > 0$ [2]. Thus, according to (4) as the time goes forward, the initial data is convolved with a Gaussian with increasing standard deviation resulting in increasingly smoothed images. As $t \to \infty$, $u(x,t)$ converges to the (constant) mean of $u_0(x)$.

Numerically (4) can be handled by convolving u_0 (i.e. the noisy image) with a Gaussian kernel (i.e. $3 \times 3, 5 \times 5$, etc) of varying standard deviation and symmetric boundary option. On the other hand, if we apply Gaussian filter with standard deviation σ to u_0, then the result should be equivalent to solving (3) until $T = \frac{1}{2}\sigma^2$, for which Finite Difference Method (FDM) and Finite Element Method (FEM) are common choices. Thus, we have established that there is an equivalence between spatial Gaussian filtering and the solution of the Heat Equation.

As was stated earlier, spatial filtering does not respect edges, and so does Heat Equation. In addition, there isn't definitive stopping criteria in this approach, if we let the heat equation solver run indefinitely, it will converge to the mean of u_0, which is not desirable.

B. Tikhonov Regularization

Another way to formulate (1) is as an energy minimization problem. The simplest such formulation is called Tikhonov regularization [4] and is defined as follows [5]

$$\min_{u \in V} E(u) = \min_{u \in V} \int_{\Omega} |\nabla u|^2 + \frac{\lambda}{2} \int_{\Omega} |u - u_0|^2, \quad \lambda > 0. \tag{5}$$

Here, V is an appropriate space of functions, i.e. $H^1(\Omega) = \{u \in L^2(\Omega) : |\nabla u| \in L^2(\Omega)\}$. What (5) is telling here is that we want to find an image u which is the closest to u_0 in the L^2 sense, and also whose H^1-seminorm is minimized. Note also that $\lambda > 0$ here is the parameter that governs the ratio between the two energies and thus controls the amount of smoothing that u gets.

If we employ calculus of variations and the method of *gradient flows*, we can convert (5) to the following parabolic PDE (See APPENDIX A for derivation).

$$\begin{cases} u_t = 2\Delta u - \lambda(u - u_0) & \Omega \times \{t > 0\} \\ u = u_0 & \Omega \times \{t = 0\} \\ \frac{\partial u}{\partial n} = 0 & \partial\Omega \times \{t > 0\} \end{cases} \tag{6}$$

All three formulations (5, 12, 6) can be solved numerically. (5) is a smooth convex functional, so after discretizing the space V, tools of convex optimization can be utilized. The elliptic problem given in (12) can be solved by FDM or FEM resulting in a linear system, which can solved by either Gauss-Seidel or Multigrid. The last, parabolic formulation (6) can be solved in the same fashion as (3) by the means of FDM and time marching (note that in the parabolic case, no linear system needs to be solved assuming explicit scheme is used). In all cases, we observe that the Gaussian smoothing is positively correlated with λ.

Again, because the energy has $\int_{\Omega} |\nabla u|^2$ term in it (u is "too regular" in a sense), Tikhonov Regularization much like the Heat Kernel, does not respect edges and blurs everything out. However, the underlying philosophy of formulating the problem by the means of energy minimization has led to the development of very effective schemes, like Total Variation minimization.

C. Total Variation Minimization

In 1992, Rudin, Osher and Fatemi [6] proposed to replace the $\int_{\Omega} |\nabla u|^2$ term in (5) by the *total variation* of ∇u:

$$\min_{u \in W^{1,1}} \int_{\Omega} |\nabla u| + \frac{\lambda}{2} \int_{\Omega} |u - u_0|^2, \quad \lambda > 0. \tag{7}$$

This change entails profound implications: the variation of u can now be supported on a set of Lebesgue measure zero, meaning that ∇u is allowed to have jumps (which was not the case with Tikhonov regularization), so u can recover edges.

By going through the same steps as before, (7) can be reformulated as a nonlinear diffusion equation

$$\begin{cases} u_t = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \lambda(u - u_0) & \Omega \times \{t > 0\} \\ u = u_0 & \Omega \times \{t = 0\} \\ \frac{\partial u}{\partial n} = 0 & \partial\Omega \times \{t > 0\} \end{cases} \tag{8}$$

This formulation of the problem produced truly remarkable results. The difficulty with this approach is that (7) while still being a convex functional, is no longer smooth because of the total variation term, furthermore (8) is nonlinear (and so computationally expensive) with the norm $|\nabla u|$ in the denominator causing numerical difficulties in FDM discretization. Finding efficient algorithm for solving either of (7) or (8) has been a challenge for more than a decade. The state of the art routine called "Split Bregman Iteration" for solving (7) and in fact a general class of L^1-regularized minimization problems has been proposed by Goldstein and Osher in 2009 [11]. An efficient Matlab implimentation of the Split Bregman method can be found in [12].

TV minimization method hits all three points mentioned at the end of introduction, making it one of the best PDE-based image denoising models.

D. Perona-Malik Diffusion

Another powerful PDE-based approach, introduced in 1987 by Perona and Malik, is Perona-Malik diffusion. The model is given by (9)

$$\begin{cases} u_t = \nabla \cdot \left(\begin{bmatrix} g(u_x) & 0 \\ 0 & g(u_y) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \right) & \Omega \times \{t > 0\} \\ u = u_0 & \Omega \times \{t = 0\} \\ \frac{\partial u}{\partial n} = 0 & \partial\Omega \times \{t > 0\} \end{cases} \tag{9}$$

where Perona and Malik suggest using the following two kernels $g(x)$ (the first one is considered standard):

$$g(x) = \frac{1}{1 + (\frac{x}{\lambda})^2}, \text{ or } g(x) = e^{-\left(\frac{x}{\lambda}\right)^2}, \quad \lambda > 0. \tag{10}$$

By taking a closer look at (6) and (8) and comparing them to (9), we notice that the first two models are *isotropic*, meaning that the diffusion process does not prefer any direction, while the latter is *anisotropic*, since the diffusion matrix is not a scalar multiple of the identity. This is why Perona-Malik model is also called anisotropic diffusion.

The advantage of Perona-Malik model over TV minimization is that it is much easier to implement via FDM (a standard implementation in Matlab can be found in [13]). In addition to the two kernels $g(x)$ proposed in [8], other variants have been proposed, see [14]–[18]. The disadvantage is that it can produce artifacts reminiscent of salt-and-pepper noise, and result in loss of detail.

Despite its subtle drawbacks, the simplicity of this model makes it of the most widely used image denoising filters. Matlab has an extremely efficient implementation of the standard Perona-Malik filter under `imdiffusefilt`.

E. Fourth-Order Filter

Fourth-order PDEs are widely used in elasticity theory. The simplest fourth-order operator called *Bilaplacian* (or Biharmonic) is defined as $\Delta^2 := \Delta(\Delta u)$. In 2000, You and Kaveh have proposed [9] to minimize the energy associated with the biharmonic operator, which leads to the following PDE

$$\begin{cases} u_t = -\Delta \left(g(\Delta u)\Delta u\right) & \Omega \times \{t > 0\} \\ u = u_0 & \Omega \times \{t = 0\} \\ \frac{\partial u}{\partial n} = \frac{\partial (g(\Delta u)\Delta u)}{\partial n} = 0 & \partial\Omega \times \{t > 0\} \end{cases} \tag{11}$$

wherein $g(x) = \lambda^2/(\lambda^2 + x^2)$. One thing to take note here is that (11) is an *isotropic* process. In 2003, Lysaker, Lundervold, and Tai have proposed an anisotropic variation of the YK model [19].

In terms of implementation, the computation of the biharmonic part can actually be done very efficiently via filtering. The problem is the time marching: since this is a fourth-order problem, in order to use an explicit FDM scheme, the time step should satisfy the CFL condition, meaning that time steps needs to be taken very small compared to the spatial mesh size. If, for instance Perona-Malik diffusion works fine with $dt \lesssim 1/8$, then the You-Kaveh model requires $dt \lesssim 1/64$ (assuming $dx = dy = 1$), resulting in more iterations until convergence.

III. Results

Heat equation solver, Tikhonov regularization, TV minimization, Perona-Malik diffusion and You-Kaveh fourth-order filter have been implemented in Matlab. For the Perona-Malik filter, aside from the two kernels in (10), three more kernels were implemented: by Charbonnier [16], by Weickert [17], and by Guo [18], they are given in Appendix B. The Matlab code is provided in [20].

After experimentation it was established that the kernels that have the best performance are the first PM kernel from (10), as well as the Charbonnier and the Guo functions.

As we have established in section A that the heat equation solver is equivalent to Gaussian filtering, we will consider Gaussian filter instead. Two types of analysis were conducted: quantitative (peak signal-to-noise-ratio, or PSNR), and qualitative (visual comparison). We refrain from analyzing time efficiency of the methods because not all implementation that we have used are state-of-the-art.

A. PSNR Analysis

PSNR is a natural metric for assessing the “quality” of an image. We have run all the algorithms on Lena and Cameraman images contaminated with Additive Gaussian Noise of two levels: “low” with $\sigma_n = 0.005$, and “high” with $\sigma_n = 0.02$ (these standard deviations are as per Matlab’s `imnoise`), determined the optimal parameter (if any) that gives the highest PSNR for each of them, and summarized the results in table (I).

Overall, it seems that TV minimization and PM-Charbonnier have the best performance of the bunch, followed by PM-1, PM-Guo, then followed by You-Kaveh, and lastly Gaussian filter and Tikhonov Regularization.

B. Visual Comparison

The denoised images corresponding to the optimal parameters from the PSNR analysis are depicted on Figure (1). It is evident that the Gaussian filter (b) and the Tikhonov regularization (c) produce equivalent results, this is because both of them are based on linear diffusion. You-Kaveh filter (e) is able to recognize edges (see cameraman with high noise level), but fails to draw the boundaries. TV minimization (f) and PM-Charbonnier (h) produce comparable, great results. PM-1 (g) and PM-Guo (i) are great at smoothing flat regions and preserving edges, but they seem to lose a lot of detail and introduce salt-and-pepper-noise-like artifacts. Overall, TV minimization and PM-Charbonnier seem to be the best models from the visual standpoint as well.

IV. Conclusion

To summarize, PDE-based image denoising models can be very effective at removing additive Gaussian noise. They are not as computationally expensive as Deep Learning models, making them a nice tradeoff between efficiency and effectiveness.

Image	Noise level	Noisy	Gauss	Tikh Reg	You-Kaveh	TV Min	Per.-Mal.	Per.-Mal.-Char.	Per.-Mal.-Guo
Lena	low	23.034	30.580	30.196	30.762	31.327	30.759	**31.562**	30.851
	high	17.182	27.753	27.588	27.454	28.405	27.726	**28.685**	27.861
Cameraman	low	23.370	27.097	26.874	28.048	**29.461**	29.238	29.442	29.188
	high	17.518	23.754	23.624	24.190	**25.476**	25.162	25.444	25.133

TABLE I

PSNR VALUES FOR DIFFERENT PDE-BASED IMAGE DENOISING METHODS.

(a) Lena, low noise level, $\sigma_n = 0.005$.

(b) Cameraman, low noise level, $\sigma_n = 0.005$.

(c) Lena, high noise level, $\sigma_n = 0.02$.

(d) Cameraman, high noise level, $\sigma_n = 0.02$.

Fig. 1. Visual comparison of the PDE-based methods. The layout is $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$, where (a) is original clean image, (b) is noisy image, (c) is Gaussian filter, (d) is Tikhonov regularization, (e) is You-Kaveh 4-th order filter, (f) is TV minimization, (g) is standard Perona-Malik diffusion, (h) is Perona-Malik diffusion with Charbonnier kernel, (i) is Perona-Malik diffusion with Guo kernel.

References

[1] Institute for Pure & Applied Mathematics, "Rudin-Osher-Fatemi Model Captures Infinity and Beyond." [Online]. Available: https://www.ipam.ucla.edu/news/rudin-osher-fatemi-model-captures-infinity-and-beyond/

[2] J. Weickert, "Anisotropic Diffusion in Image Processing," 1998. [Online]. Available: https://www.mia.uni-saarland.de/weickert/Papers/book.pdf

[3] L. C. Evans, *Partial differential equations*, 2nd ed., ser. Graduate studies in mathematics. Providence, R.I: American Mathematical Society, 2010, no. v. 19, oCLC: ocn465190110.

[4] A. Tikhonov, "On The Stability of Inverse Problems," *Proceedings of the USSR Academy of Sciences*, 1943.

[5] E. Rusu, "A Variational Approach to Signal Denoising," 2016. [Online]. Available: https://github.com/gjlper/Tikhonov-Denoising/blob/master/Project1_Report.pdf

[6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/016727899290242F

[7] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, Feb. 1997. [Online]. Available: https://ieeexplore.ieee.org/document/551699/

[8] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, Jul. 1990. [Online]. Available: http://ieeexplore.ieee.org/document/56205/

[9] Y.-L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1723–1730, Oct. 2000. [Online]. Available: http://ieeexplore.ieee.org/document/869184/

[10] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief Review of Image Denoising Techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, 12 2019.

[11] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009. [Online]. Available: https://doi.org/10.1137/080725891

[12] B. Tremoulheac, "Split Bregman method for Total Variation Denoising," 2022. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/36278-split-bregman-method-for-total-variation-denoising

[13] D. Lopes, "Anisotropic Diffusion (Perona Malik)," 2022. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/14995-anisotropic-diffusion-perona-malik

[14] B. J. Maiseli, "On the convexification of the Perona–Malik diffusion model," *Signal, Image and Video Processing*, vol. 14, no. 6, pp. 1283–1291, Sep. 2020. [Online]. Available: http://link.springer.com/10.1007/s11760-020-01663-x

[15] V. Kamalaveni, R. A. Rajalakshmi, and K. Narayanankutty, "Image Denoising Using Variations of Perona-Malik Model with Different Edge Stopping Functions," *Procedia Computer Science*, vol. 58, pp. 673–682, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050915021985

[16] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.

[17] J. Weickert, "Coherence-enhancing diffusion filtering," *International Journal of Computer Vision*, vol. 31, pp. 111–127, 2004.

[18] Z. Guo, J. Sun, D. Zhang, and B. Wu, "Adaptive perona–malik model based on the variable exponent for image denoising," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 958–967, 2012.

[19] M. Lysaker, A. Lundervold, and Xue-Cheng Tai, "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1579–1590, Dec. 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1257394/

[20] M. Shakipov, "PDE-based Image Denoising." [Online]. Available: https://github.com/chromomons/umd-enee631-pde-based-image-denoising

Appendix A
Optimization of the Tikhonov functional and Gradient Flow

The functional $E(u)$ in (5) is minimized when its "derivative" is equal to 0. Let $\alpha > 0, v \in V$, then $u + \alpha v$ is a perturbation of u. Thee Gateaux derivative of $E(u)$ in the direction v is defined as

$$\lim_{\alpha \to 0} \frac{E(u+\alpha v) - E(u)}{\alpha} = \frac{d}{d\alpha} E(u+\alpha v)|_{\alpha=0}.$$

We now compute the derivative.

$$E(u+\alpha v) = \frac{\lambda}{2} \int_\Omega (u + \alpha v - u_0)^2 + \int_\Omega |\nabla(u+\alpha v)|^2$$

$$\frac{d}{d\alpha} E(u+\alpha v) = \lambda \int_\Omega (u + \alpha v - u_0) v + 2 \int_\Omega \nabla(u+\alpha v) \cdot \nabla v$$

Set $\alpha = 0$:

$$\frac{d}{d\alpha} E(u+\alpha v)|_{\alpha=0} = \int_\Omega (u - u_0) v + 2 \int_\Omega \nabla u \cdot \nabla v.$$

We want $\frac{d}{d\alpha} E(u+\alpha v)|_{\alpha=0} = 0$, as we are seeking the minimizer, so we obtain the so-called *weak formulation* of the problem

$$\int_\Omega (u - u_0) v + 2 \int_\Omega \nabla u \cdot \nabla v = 0, \quad \forall v \in V.$$

Assuming extra regularity on u, using integration by parts, we get

$$\int_\Omega \left[\lambda(u - u_0) - 2\Delta u\right] v = 0, \quad \forall v \in V.$$

Since the above equality is true for any $v \in V$, we get the elliptic PDE given in (12).

$$\begin{cases} -2\Delta u + \lambda(u - u_0) = 0 & \Omega \\ \frac{\partial u}{\partial n} = 0 & \partial\Omega \end{cases} \tag{12}$$

(12) can be solved via *gradient flows*, which a generalization of gradient descent to infinite dimensional vector spaces. We convert (12) to a parabolic problem of the form given in (6) s.t. as $t \to \infty$ the solution of (6) approaches the solution of (12) in the appropriate sense.

Appendix B
Charbonnier, Weickert and Guo kernels for the Perona-Malik diffusion

$$g_{ch}(x) = \sqrt{\frac{1}{1 + (\frac{x}{\lambda})^2}}, \quad g_{we}(x) = 1 - e^{-3.31488 |\frac{\lambda}{x}|^8}.$$

$$g_{guo}(x) = \frac{1}{1 + (\frac{x}{\lambda})^{\alpha(x)}}, \quad \alpha(x) = 2 - \frac{2}{1 + (\frac{x}{\lambda})^2}.$$