Edge-Aware Image Denoising via Adaptive Weighted Fractional-Order Total Variation under Cauchy Noise

Hamdi Braiek[1,*]

[1]ESPRIT School of Engineering, Ariana, Tunisia

ICRAMI 2025 / Sousse, Tunisia

*hamdi.houichet@gmail.com

November 23, 2025

Outline

Real word applications

Cauchy noise appears everywhere in real imaging systems

- Radar & SAR imagery $\longrightarrow$ speckle $+$ impulsive bursts
- Ultrasound & sonar $\longrightarrow$ heavy-tailed interference
- Atmospheric turbulence, MRI artifacts, quantum-limited imaging

Real word applications

Cauchy noise appears everywhere in real imaging systems

- Radar & SAR imagery $\longrightarrow$ speckle + impulsive bursts
- Ultrasound & sonar $\longrightarrow$ heavy-tailed interference
- Atmospheric turbulence, MRI artifacts, quantum-limited imaging
- Classical Gaussian assumption $\longrightarrow$ catastrophic failure
- Cauchy distribution: infinite variance, α-stable with $\alpha = 2 \longrightarrow$ Gaussian, $\alpha \to 0 \longrightarrow$ extremely impulsive

Real SAR image corrupted by heavy-tailed noise

Best model appropriats to denoising problem

Why current methods fail on Cauchy noise?

Method	Fidelity	Regularization	Major drawback
ROF (TV-L2)	Gaussian	1st-order TV	Stair-casing + outlier collapse
TV-L1 / Median	Laplace	1st-order TV	Texture loss
Non-local Means / BM3D	Gaussian	Patch-based	Slow, fails on extreme outliers
Unweighted FOTV	Various	Fractional TV	Isotropic smoothing
Cauchy + standard TV	Cauchy	1st-order	Severe stair-casing
Weighted integer-order TV	Cauchy	Adaptive TV	Still piecewise constant

Best model appropriats to denoising problem

Why current methods fail on Cauchy noise?

Method	Fidelity	Regularization	Major drawback
ROF (TV-L2)	Gaussian	1st-order TV	Stair-casing + outlier collapse
TV-L1 / Median	Laplace	1st-order TV	Texture loss
Non-local Means / BM3D	Gaussian	Patch-based	Slow, fails on extreme outliers
Unweighted FOTV	Various	Fractional TV	Isotropic smoothing
Cauchy + standard TV	Cauchy	1st-order	Severe stair-casing
Weighted integer-order TV	Cauchy	Adaptive TV	Still piecewise constant

No existing model combines:

- Robust Cauchy fidelity
- Adaptive + fractional-order regularization ($1 < \alpha < 2$)
- Automatic hyper-parameter tuning

Our proposal

Proposed energy

$$\mathcal{J}(u) = \lambda \underbrace{\int \log\left(1 + \frac{(f-u)^2}{\gamma^2}\right)}_{\text{Cauchy fidelity (robust to outliers)}} + \underbrace{\int w(x)\,|\nabla_C^\alpha u|(x)\,dx}_{\text{Weighted Caputo FOTV (adaptive + nonlocal)}}$$

Our proposal

Proposed energy

$$\mathcal{J}(u) = \lambda \underbrace{\int \log\left(1 + \frac{(f-u)^2}{\gamma^2}\right)}_{\text{Cauchy fidelity (robust to outliers)}} + \underbrace{\int w(x)\, |\nabla_C^\alpha u|(x)\, dx}_{\text{Weighted Caputo FOTV (adaptive + nonlocal)}}$$

Four major advantages of the model

1. Extreme robustness to impulsive heavy-tailed noise
2. Excellent preservation of edges and fine details
3. Superior global-to-local smoothness control
4. High stability and quantitative performance

Caputo fractional gradient & weighted FOTV

Caputo directional derivative ($1 < \alpha \leq 2$)

$$ {}^{C}D_x^{\alpha} u(x,y) = \frac{1}{\Gamma(2-\alpha)} \int_a^x (x-t)^{1-\alpha} \partial_{tt} u(t,y)\, dt $$

Weighted Fractional Total Variation

$$ \mathrm{FOTV}_w^{\alpha}(u) = \sup_{\|v\|_\infty \leq w} \int -u\, \mathrm{div}^{\alpha} v\, dx \qquad \longrightarrow \qquad \int w(x)\, |\nabla_C^{\alpha} u|(x)\, dx $$

Key properties

Convex, lower-semi-continuous, coercive in $BV_w^{\alpha}(\Omega) \longrightarrow$ existence of minimizers

Adaptive weight: $w(x) = \exp\big(-|\nabla(G_\sigma * u(x))|\big)$, $\longrightarrow$ edge-preserving

Efficient Split Bregman Solver with Half-Quadratic Reformulation

Introduce auxiliary field $\mathbf{d} \approx \nabla_C^\alpha u$
Constrained problem $\rightarrow$ Augmented Lagrangian:

$$\mathcal{L}(u, \mathbf{d}, \mathbf{b}) = \lambda \sum_{i,j} \log\left(1 + \frac{(f_{i,j} - u_{i,j})^2}{\gamma^2}\right) + \sum_{i,j} w_{i,j} |\mathbf{d}_{i,j}| + \frac{\beta}{2} \sum_{i,j} \|\mathbf{d}_{i,j} - \nabla_C^\alpha u_{i,j} + \mathbf{b}_{i,j}\|_2^2$$

Split Bregman iteration (k):

1. u-subproblem (nonconvex Cauchy term)

$$u^{k+1} = \operatorname*{argmin}_u \lambda \sum \phi_\gamma(f - u) + \frac{\beta}{2} \|\mathbf{d}^k - \nabla_C^\alpha u + \mathbf{b}^k\|_2^2$$

   $\rightarrow$ Half-quadratic reformulation:

$$\phi_\gamma(r) = \min_{\omega > 0} \frac{\omega}{2} r^2 + \psi(\omega), \qquad \omega^\star(r) = \frac{2}{\gamma^2 + r^2}$$

   $\rightarrow$ Iteratively reweighted least-squares $\rightarrow$ solved by **Conjugate Gradient**

2. d-subproblem: weighted shrinkage (closed form)

$$\mathbf{d}_{i,j}^{k+1} = \mathrm{shrink}\left(\nabla_C^\alpha u_{i,j}^{k+1} - \mathbf{b}_{i,j}^k, \frac{w_{i,j}}{\beta}\right)$$

3. Bregman update: $\mathbf{b}^{k+1} = \mathbf{b}^k + \nabla_C^\alpha u^{k+1} - \mathbf{d}^{k+1}$

Efficient Split Bregman Solver with Half-Quadratic Reformulation

Introduce auxiliary field $\mathbf{d} \approx \nabla_C^\alpha u$
Constrained problem $\rightarrow$ Augmented Lagrangian:

$$\mathcal{L}(u, \mathbf{d}, \mathbf{b}) = \lambda \sum_{i,j} \log\left(1 + \frac{(f_{i,j} - u_{i,j})^2}{\gamma^2}\right) + \sum_{i,j} w_{i,j} |\mathbf{d}_{i,j}| + \frac{\beta}{2} \sum_{i,j} \|\mathbf{d}_{i,j} - \nabla_C^\alpha u_{i,j} + \mathbf{b}_{i,j}\|_2^2$$

Split Bregman iteration (k):

1. u-subproblem (nonconvex Cauchy term)

$$u^{k+1} = \operatorname*{argmin}_u \lambda \sum \phi_\gamma(f - u) + \frac{\beta}{2} \|\mathbf{d}^k - \nabla_C^\alpha u + \mathbf{b}^k\|_2^2$$

$\rightarrow$ Half-quadratic reformulation:

$$\phi_\gamma(r) = \min_{\omega > 0} \frac{\omega}{2} r^2 + \psi(\omega), \qquad \omega^\star(r) = \frac{2}{\gamma^2 + r^2}$$

$\rightarrow$ Iteratively reweighted least-squares $\rightarrow$ solved by **Conjugate Gradient**

2. d-subproblem: weighted shrinkage (closed form)

$$\mathbf{d}_{i,j}^{k+1} = \mathrm{shrink}\left(\nabla_C^\alpha u_{i,j}^{k+1} - \mathbf{b}_{i,j}^k, \frac{w_{i,j}}{\beta}\right)$$

3. Bregman update: $\mathbf{b}^{k+1} = \mathbf{b}^k + \nabla_C^\alpha u^{k+1} - \mathbf{d}^{k+1}$

Convergence: typically < 130 outer iterations on 1024x1024 images

Automatic hyperparameter tuning via Bayesian Optimization

Performance heavily depends on the choice of the hyperparameters $\Theta = (\lambda,\ \alpha,\ \beta)$
Manual tuning is time-consuming, image-dependent, and suboptimal

Bayesian Optimization (BO) framework

1. **Objective function** (noiseless when ground-truth available)

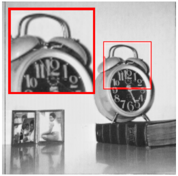
$$J(\Theta) \;=\; \mathrm{PSNR}\big(u^\star(\Theta;\, f),\ u_{\text{clean}}\big)$$

2. **Search space:** $\lambda \in [10^{-5}, 5], \quad \alpha \in (1.1, 2.0], \quad \beta \in [10^{-2}, 5]$
3. **Surrogate model**: Gaussian Process (GP) with Matérn-5/2 kernel
4. **Acquisition function**: Expected Improvement (EI)

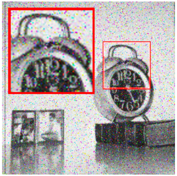
$$\mathrm{EI}(\Theta) = \mathbb{E}\big[\max(0,\ J(\Theta) - J(\Theta^+))\big]$$

5. **Initialization**: 15 random points (Latin Hypercube Sampling)
6. **Optimization loop**
   - For $t = 1$ to 20:
     $\Theta_t = \arg\max_\Theta \mathrm{EI}(\Theta)$
     Run full denoising with $\Theta_t \to$ observe $J(\Theta_t)$
     Update GP posterior with $(\Theta_t, J(\Theta_t))$
7. **Final selection**: $\Theta^\star = \arg\max_t J(\Theta_t)$

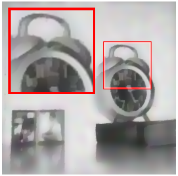
$\longrightarrow$ **Fully automatic**, robust across images and noise levels $\longrightarrow$ No user intervention required



(a) True image



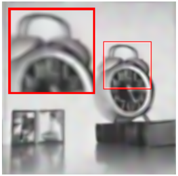
(b) Noisy image



(c) TV model

(d) pen-TV

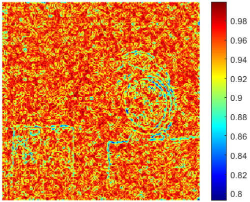
(e) TGV



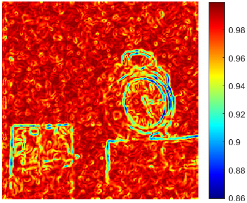
(f) FOTV

(g) Weighted FOTV

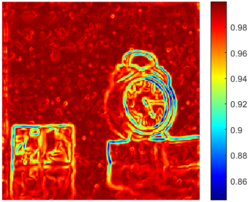
Figure : From left to right: The original image (clock), noisy image degraded by Cauchy noise with $\gamma = 0.05$ and the restored images by TV, FOTV model and weighted FOTV



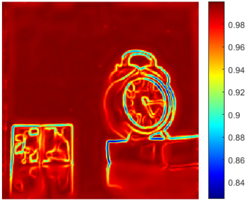
(a) iteration 1



(b) Iteration 30



(c) Iteration 70



(d) Iteration 120

Figure : From left to right: The weighted function w at different iterations

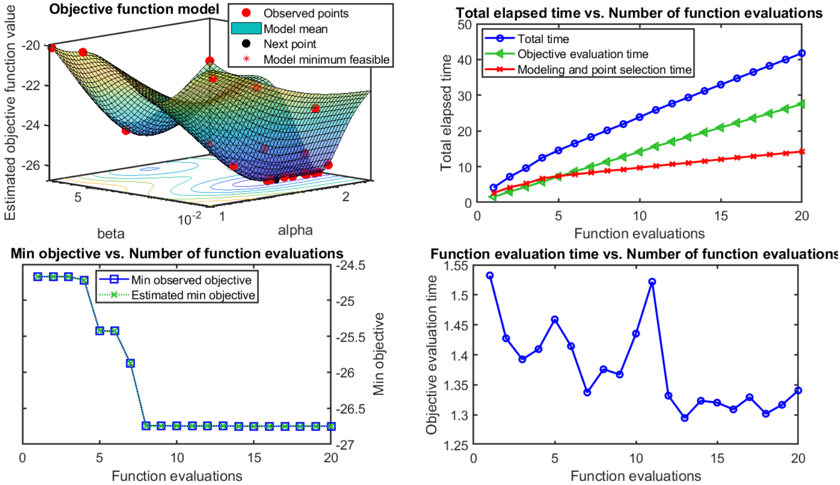
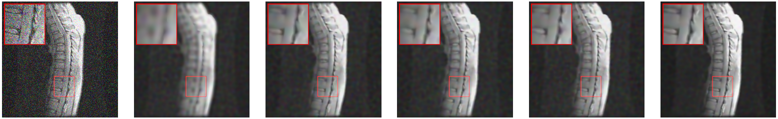
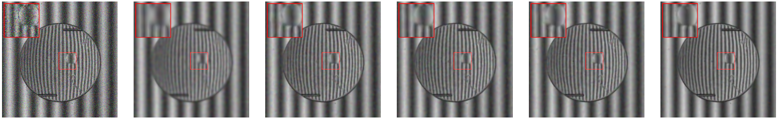


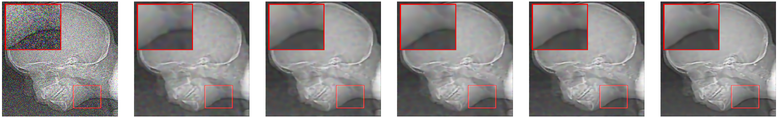
Figure : From left to right: The tuned parameters using BO for the image in Figure 1(g).



(a) Noisy image (b) **TV** (c) **pen-TV** (d) **TGV** (e) **FOTV** (f) **Weighted FOTV**



(g) Noisy image (h) **TV** (i) **pen-TV** (j) **TGV** (k) **FOTV** (l) **Weighted FOTV**



(m) Noisy image (n) **TV** (o) **pen-TV** (p) **TGV** (q) **FOTV** (r) **Weighted FOTV**

Figure : From left to right: The noisy image degraded by Cauchy noise and the restored images by TV, pen-TV, TGV, FOTV and Weighted FOTV models, respectively.

Table : Table for comparison of results of gray-scale images.

Image	Metrics	Noisy	Models				
			TV	**pen-TV**	**TGV**	**FOTV**$^{\alpha}$	**FOTV**$_{w}^{\alpha}$
Clock (256×256) $\gamma = 0.05$	SNR	12.887	22.372	24.039	24.00	24.367	25.129
	PSNR	15.2841	24.926	26.550	26.532	26.878	27.595
	SSIM	0.170	0.782	0.822	0.812	0.824	0.843
	Iterations	-	180	160	140	140	110
MRI (512×512) $\gamma = 0.07$	SNR	6.176	15.899	16.736	16.786	16.817	20.710
	PSNR	13.944	24.879	25.68	25.738	25.762	29.701
	SSIM	0.070	0.741	0.763	0.768	0.764	0.840
	Iterations	-	200	180	140	150	130
Skull (374×452) $\gamma = 0.05$	SNR	9.159	27.331	25.661	22.908	23.761	26.490
	PSNR	13.998	31.819	31.053	28.294	29.131	32.820
	SSIM	0.077	0.726	0.824	0.759	0.784	0.874
	Iterations	-	160	140	160	120	110
Lens (590×614) $\gamma = 0.05$	SNR	9.367	21.350	22.697	24.110	24.138	27.123
	PSNR	15.316	27.893	29.226	30.621	30.644	33.622
	SSIM	0.12943	0.865	0.891	0.900	0.902	0.936
	Iterations	-	240	220	200	150	130

Conclusion

Summary

- First weighted Caputo fractional-order model for Cauchy denoising
- Theoretically grounded (existence in weighted BV_w^α)
- Highly efficient solver (SB + half-quadratic + CG)
- Fully automatic thanks to Bayesian optimization

Thank you for your attention!

Questions?