A TWO-STAGE DENOISING FILTER: THE PREPROCESSED YAROSLAVSKY FILTER

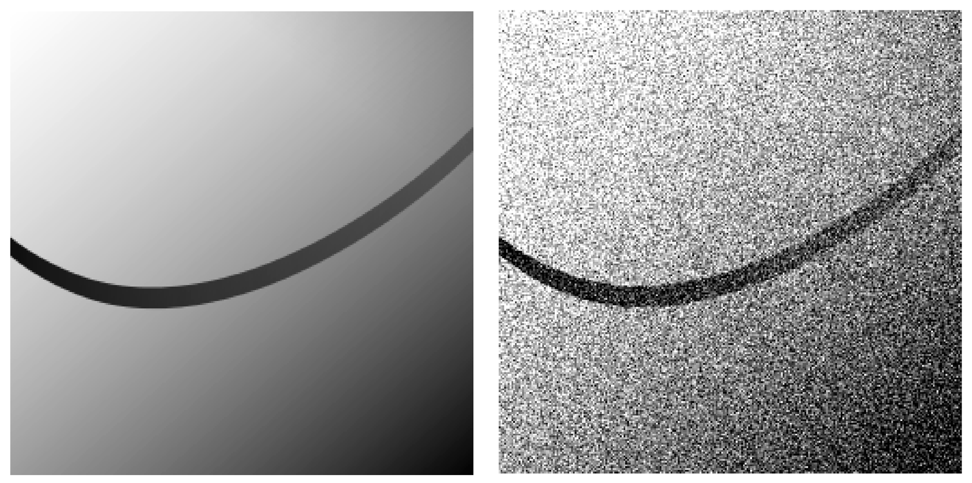
J. Salmon * R. Willett * E. Arias-Castro†

* Duke University, ECE Department, Durham, NC, USA — †Department of Mathematics, University of California, San Diego, CA, USA.

PROBLEM FORMULATION

We observe noisy samples $\{y_i \in \mathbb{R} : i \in I_n^d\}$ (with $I_n := \{1, \ldots, n\}$) of the target function $f : [0,1]^d \to [0,1]$ from a **cartoon class** (*i.e.,* with Holder α-smooth surfaces and Lipschitz boundary) at n^d design points $\{x_i \in \mathbb{R}^d : i \in I_n^d\}$ corrupted by AWGN ($\sigma^2 > 0$), $\{\varepsilon_i \in \mathbb{R} : i \in I_n^d\}$, as follows

$$y_i = f(x_i) + \varepsilon_i, \quad i \in I_n^d. \tag{1}$$



Original, MSE=0 — Noisy, MSE=2.51e+03

NEIGHBORHOOD FILTERS

We consider neighborhood filters of the form

$$\widehat{f}_i = \frac{\Sigma_{j \in I_n^d}\, \omega(i,j)\, y_j}{\Sigma_{k \in I_n^d}\, \omega(i,k)}. \tag{2}$$

where the weights $\omega(i,j)$ (may) depend on the observation $\mathbf{y}$. For $\alpha > 1$ we incorporate local polynomial regression to adapt to higher orders of smoothness.

Linear filtering (LF): Only spatial proximity is used here, so for a kernel K and a bandwidth $h > 0$, the weights can be written

$$\omega(i,j) = K_h(x_i, x_j)\,, \tag{3}$$

where $K_h(x_i, x_j) = K(\frac{x_i}{h}, \frac{x_j}{h})$ for any sample points x_i and x_j.

Weight oracle (WO): The weights are based on the true image f:

$$w_{i,j} := K_h(x_i, x_j) \mathbb{I}_{\{|f_i - f_j| < h_y\}}. \tag{4}$$

Again, K and h control the spatial proximity; the photometric bandwidth h_y controls the photometric proximity.

Yaroslavsky's filter (YF): The similarity between two pixels is based on spatial distance and on the relative proximity of image intensity:

$$\omega(i,j) = K_h(x_j, x_j)\; \mathbb{I}_{\{|y_i - y_j| < h_y\}}. \tag{5}$$

PATCH BASED FILTERS

Non-Local Means (NLM): For the *nonlocal means (NLM)*, one estimates the photometric distance between pixels using patches of noisy pixels. For $h_{\mathrm{P}} > 0$, let $\mathbf{y}_{\mathrm{P}_i}$ be the vector of pixel values over the patch centered at x_i. The weights for NLM are:

$$\omega(i,j) = K_h(x_i, x_j)\; \mathbb{I}_{\{\|\mathbf{y}_{\mathrm{P}_i} - \mathbf{y}_{\mathrm{P}_j}\| < h_y\}}. \tag{6}$$

Non-Local Means Average (NLM-Av.): A fast approximation to NLM is effective on cartoon images: compute the average of pixels within each patch, and use the differences of averages (here $\overline{y}_{\mathrm{P}_i}$ is the pixel average on patch i) to estimate photometric distances:

$$\omega(i,j) = K_h(x_i, x_j)\; \mathbb{I}_{\{|\overline{y}_{\mathrm{P}_i} - \overline{y}_{\mathrm{P}_j}| < h_y\}}, \tag{7}$$

THE PREPROCESSED YAROSLAVSKY FILTER

- Compute an initial estimate of $\mathbf{f}$, denoted $\tilde{\mathbf{f}} := \mathrm{denoise}(\mathbf{y})$.
- Use $\tilde{\mathbf{f}}$ to compute the weights in a Yaroslavsky-type filter

$$w_{i,j}^{\mathrm{PY}} := K_h(x_i, x_j) \mathbb{I}_{\{|\tilde{f}_i - \tilde{f}_j| < h_y^{\mathrm{PY}}\}}. \tag{8}$$

Possible first step considered for $\tilde{\mathbf{f}}$ are: wavelet with cycle spinning, Curvelet and Linear Filtering, leading to YFWav, YFCurvelet and NLM-Av.

THEORETICAL RESULTS

The Weight oracle $\hat{\mathbf{f}}_h^{\mathrm{WO}}$ achieves minimax rate on the cartoon class:

$$\inf_h \sup_{f \in \mathcal{F}} \mathrm{MSE}_f(\mathbf{f}, \mathbf{f}_h^{\mathrm{WO}}) \asymp \mathcal{R}^{\mathrm{WO}} := (\sigma^2/n^d)^{2\alpha/(d+2\alpha)},$$

For small noise (*i.e.,* $\sigma^2 = O(1/\sqrt{\log n})$), then $|y_i - y_j|$ is a close approximation to $|f_i - f_j|$, the YF performs nearly as well as WO.

Theorem:

Suppose an estimator $\tilde{\mathbf{f}}$ satisfies for any $f \in \mathcal{F}$ the following deviation bound, with probability at least $1 - \delta$:

$$|\tilde{f}_i - f_i|^2 \leq M, \qquad \forall i \in I_n^d \text{ such that } B(x_i, \tilde{h}) \cap \partial\Omega = \emptyset.$$

where $\partial\Omega$ is the boundary between the smooth surfaces, then if $M = o(1)$, for $\hat{\mathbf{f}}_h^{\mathrm{PY}}$ defined with weights as in (8), one has

$$\inf_h \sup_{f \in \mathcal{F}} \mathrm{MSE}_f(\mathbf{f}, \hat{\mathbf{f}}_h^{\mathrm{PY}}) \asymp \tilde{h} + \delta + (\sigma^2/n^d)^{2\alpha/(d+2\alpha)},$$

and the optimal choice of bandwidths are $h \asymp h^{\mathrm{WO}}$ and $h_y \asymp 1$.

MSE PERFORMANCE

	Blob	Swoosh	Ridges	Cameraman
	$\sigma = 5$			
LF	35.33	40.29	48.80	437.79
WavCS	1.40	1.78	1.65	14.74
Curvelet	5.12	4.88	1.58	28.96
YF	1.27	2.10	16.95	14.57
NLM-Av.	0.86	2.39	4.19	315.96
YFWavCS	**0.78**	1.21	1.49	14.12
YFCurvelet	1.16	2.02	1.81	24.42
NLM	0.96	**1.14**	2.11	13.40
BM3D	1.22	1.20	**0.88**	**9.95**
WO	1.61	2.15	36.36	32.59
	$\sigma = 20$			
LF	43.19	48.17	56.62	445.65
WavCS	15.31	20.15	14.98	102.07
Curvelet	19.20	34.32	10.66	148.92
YF	17.00	22.00	189.05	120.11
NLM-Av.	5.06	7.39	18.95	345.69
YFWavCS	4.19	6.41	13.57	88.76
YFCurvelet	**3.22**	**4.67**	13.88	114.92
NLM	4.03	4.74	25.98	91.72
BM3D	5.72	7.04	**8.94**	**59.36**
WO	2.66	3.19	38.00	34.24

Results averaged over 100 Gaussian noise replicas on common images.

TIME PERFORMANCE

Size	LF	Wav	Curv.	YF	YFWav	YFCurv.	NLM-Av	NLM	BM3D
256^2	0.03 .s	0.08 .s	0.33 .s	0.16 .s	0.26 .s	0.48 .s	0.15 .s	14.75 .s	1.18 .s
512^2	0.08 .s	0.53 .s	1.13 .s	0.72 .s	1.28 .s	1.75 .s	0.63 .s	60.00 .s	4.99 .s
1024^2	0.18 .s	2.97 .s	3.90 .s	2.89 .s	5.94 .s	6.47 .s	2.48 .s	241.87 .s	21.42 .s

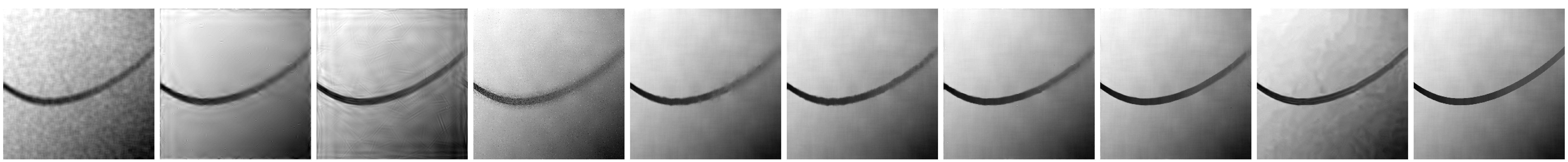
Computing times for Matlab mex/C implementations (except Curvelet is pure Matlab) on an Intel Core i7 CPU 2.67GHz.

ACKNOWLEDGEMENTS

This work was supported by NSF CAREER Award No. CCF-06-43947 and ARO Grant No. W911NF-09-1-0262.

Online code : http://josephsalmon.eu/

VISUAL RESULTS



LF, MSE=9.13e+01 — WavCS, MSE=7.89e+01 — Curvelet, MSE=7.52e+01 — YF, MSE=1.29e+02 — NLM, MSE=3.73e+01 — NLM-Av., MSE=2.69e+01 — YFWavCS, MSE=2.52e+01 — YFCurvelet, MSE=1.59e+01 — BM3D, MSE=2.29e+01 — WO, MSE=9.16e+00