Oracle inequalities and minimax rates for non-local means

Rebecca Willett, Duke

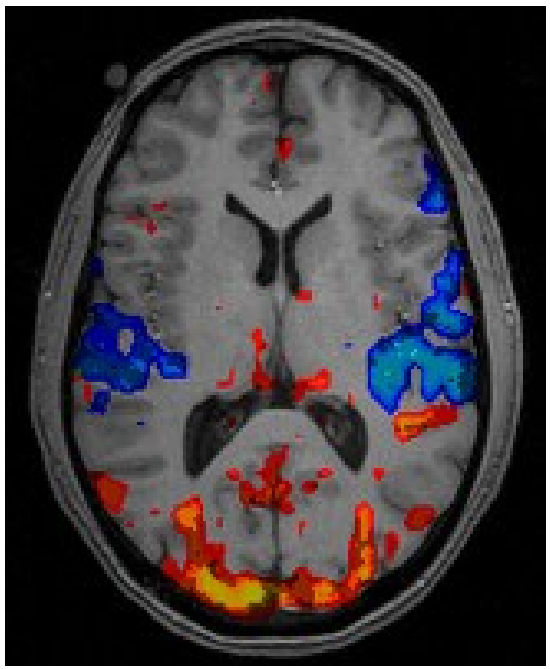
Ery Arias-Castro, UCSD

Joseph Salmon, Duke

Image filtering in neuroimaging

Standard processing chain for fMRI data analysis:

1. Collect data, reconstruct time-series of images
2. Filter to reduce noise
3. Map data to standard anatomical template
4. Look for active regions



Filtering is most often done with linear Gaussian filter.

Image filtering in neuroimaging

Linear Gaussian filtering is problematic:

- it introduces bias
- it blurs BOLD signals across different functional regions
- it blurs BOLD signals across sulci

More recent fMRI literature considers replacing the linear Gaussian filter with a (nonlinear) bilateral filter[1], but this has not been widely adopted.

These challenges arise in many neuroimaging contexts beyond fMRI.

When and how can we do better?

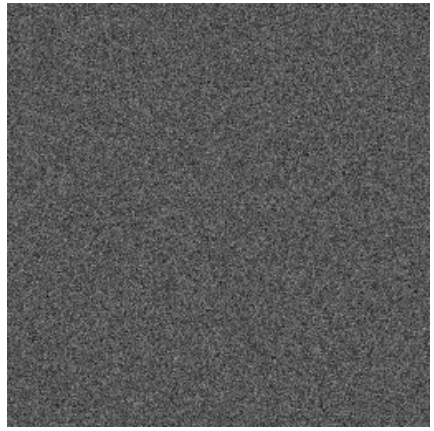
[1] cf. Walker, Miller, & Tanabe 2006; Rydell, Knutsson, & Borga 2008


Observed image y

$=$


Underlying scene f

$+$


Noise ε

$$y = f + \varepsilon; \qquad \varepsilon \text{ uncorrelated, mean}=0, \text{ var}=\sigma^2$$

Estimate f_i as a weighted average of the noisy pixels:

$$\widehat{f_i} = \sum_j w_{i,j} y_j$$

How should we choose the weights?

Kernel-based denoising: $\widehat{f_i} = \sum_j w_{i,j} y_j$

Usual kernel method [a]

$$w_{i,j} = K_h(x_i, x_j)$$

- w has no dependency on y
- K: kernel and h: bandwidth (smoothing parameter)
- Gaussian kernel example : $K_h(x_i, x_j) = e^{-\|x_i - x_j\|_2^2 / 2h^2}$

[a] Nadaraya '64, Watson '64

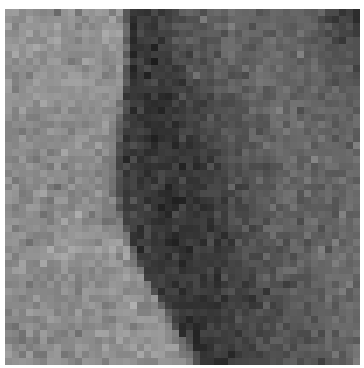
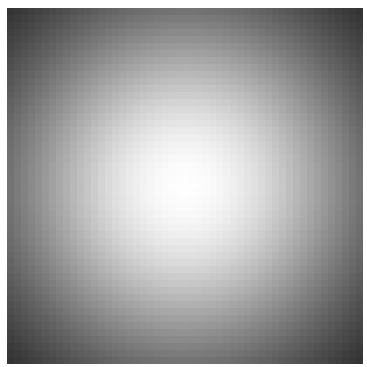


Image Search Zone



Spatial

Kernel-based denoising: $\widehat{f_i} = \sum_j w_{i,j} y_j$

Yaroslavsky/Bilateral Filter [a]

$$w_{i,j} = K_h(x_i, x_j) L_{h_y}(y_i, y_j)$$

- Use spatial *and* photometric proximity
- K, L: kernels; h, h_y: bandwidths (smoothing parameters)

[a] Yaroslavsky '85, Lee '83, Tomasi and Manduchi '98

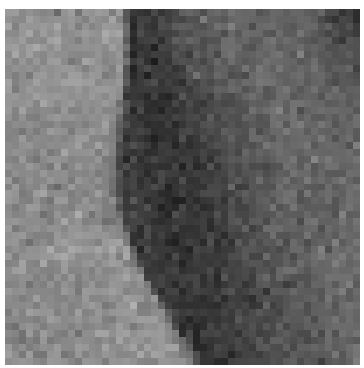
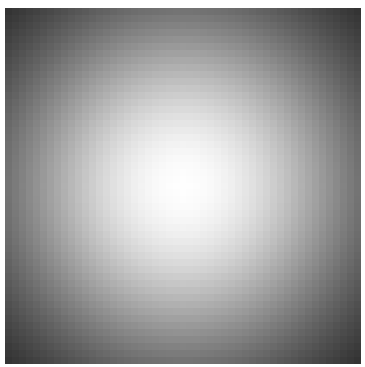
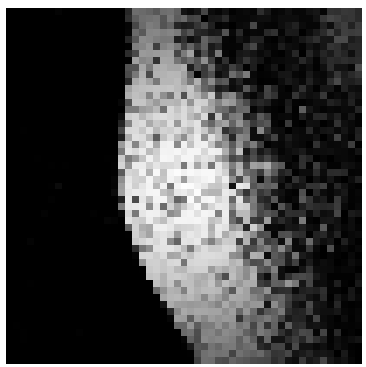


Image Search Zone



Spatial



Yaroslavsky / Bilateral

Kernel-based denoising: $\widehat{f_i} = \sum_j w_{i,j} y_j$

Non-local Means [a]

$$w_{i,j} = K_h(x_i, x_j) L_{h_y}(y_{P_i}, y_{P_j})$$

- Use spatial *and* photometric proximity
- K, L: kernels; h, h_y: bandwidths (smoothing parameters)
- P_i is a small patch of pixels centered around pixel i

[a] Buades, Coll & Morel '05

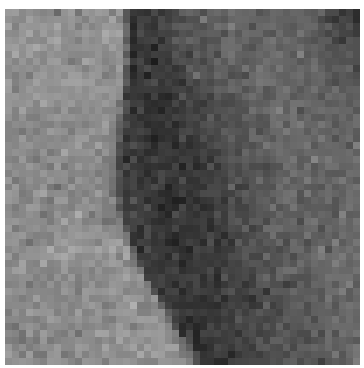
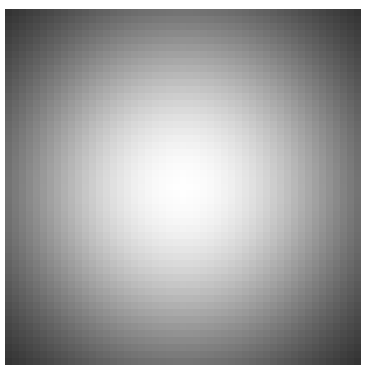
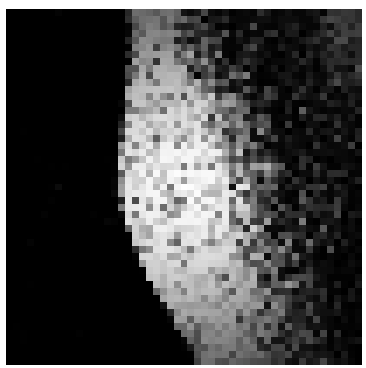


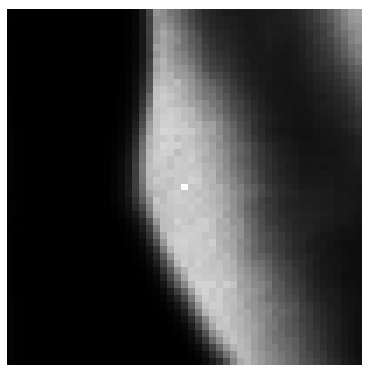
Image Search Zone



Spatial



Yaroslavsky / Bilateral



Non-local means

Problem formulation

We will bound the risk

$$\mathcal{R}_n(\widehat{f}, \mathcal{F}) := \sup_{f \in \mathcal{F}} \mathrm{MSE}_f(\widehat{f}) = \sup_{f \in \mathcal{F}} \frac{\mathbb{E}\|\widehat{f} - f\|_2^2}{n^d}.$$

- How do errors scale with
  - n (number of pixels),
  - d (dimension), and
  - σ^2 (noise variance)?
- How do errors compare with those of wavelets, which can be minimax optimal?
- How should parameters [h (search zone or bandwidth), h_y (photometric bandwidth), h_{P} (patch size)] be chosen?
- How can these insights be leveraged to develop better methods?

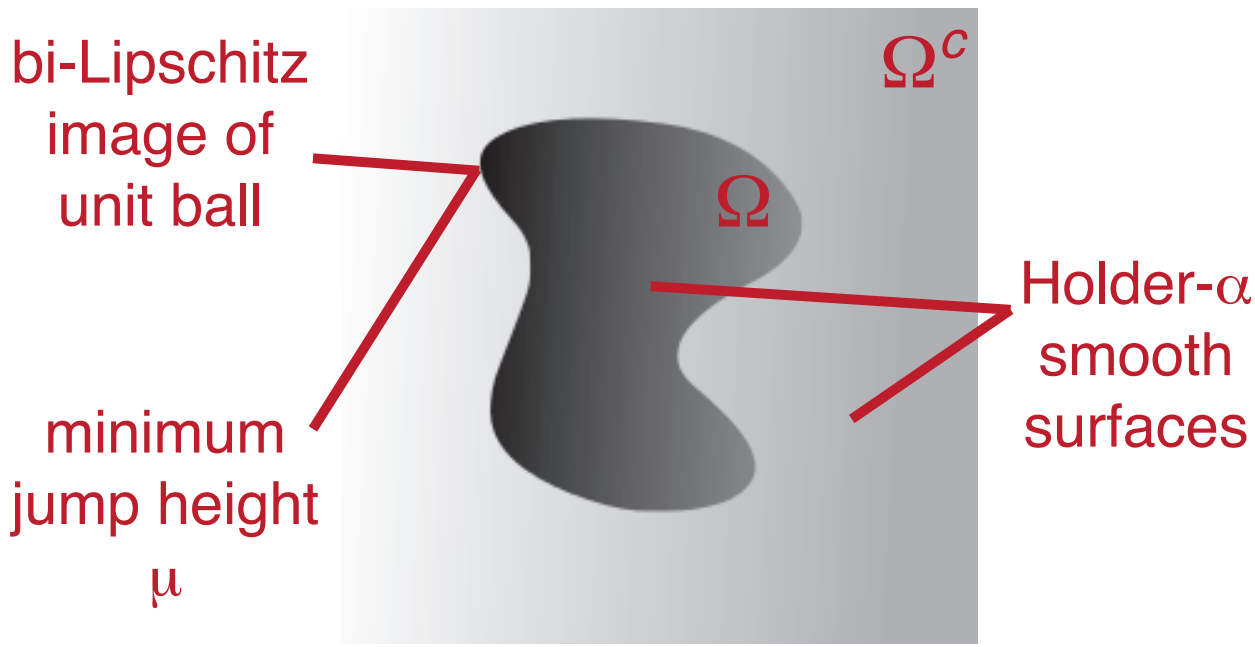
Related theoretical investigations

- Information-theoretic interpretation: Weissman *et al.* '05
- Consistency: Buades *et al.* '05
- Graph diffusion interpretation: Singer *et al.* '09, Taylor & Meyer '11
- Rare patch effect: Duval *et al.* '11
- SURE estimate of parameters: Van De Ville & Kocher '09,'11, Duval *et al.* '11, Deledalle *et al.* '11
- Cramer-Rao bounds: Levin & Nadler '11, Chatterjee & Milanfar '11
- Minimax rates for piecewise constant images: Maleki, Narayan & Baraniuk '11

Cartoon images

$f \in \mathcal{F}^{\text{cartoon}}$ is a "cartoon image" if it is a piecewise smooth (Hölder-α, $\alpha \geq 1$) image with discontinuities along smooth hypersurfaces.[2].

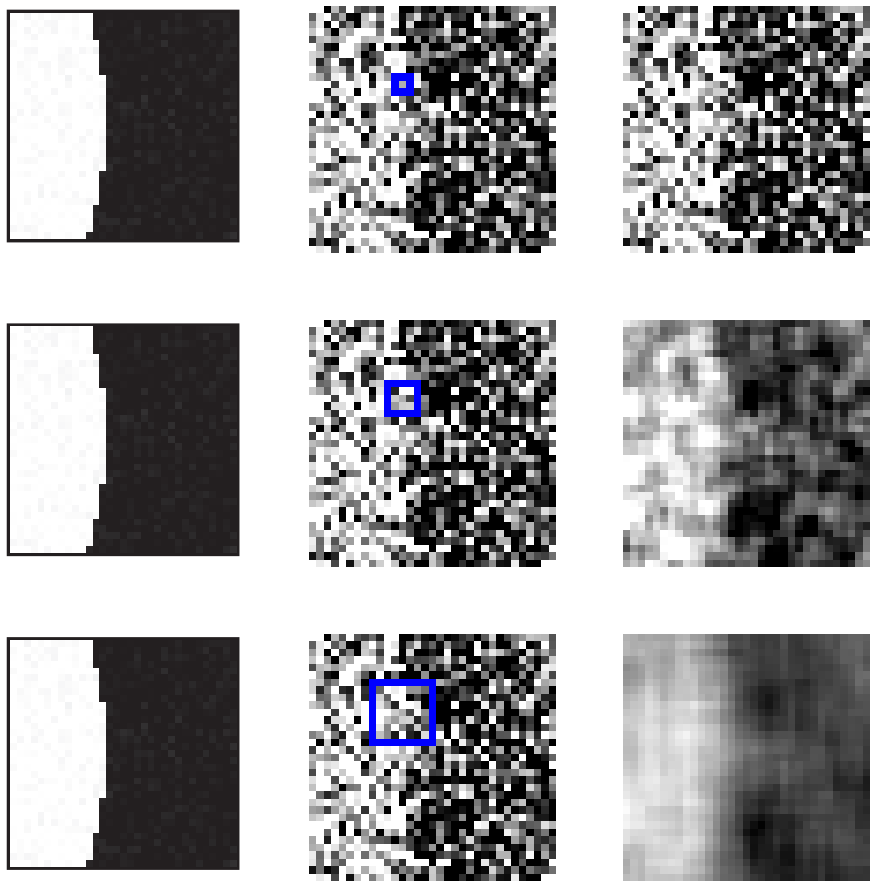
$$f(x) = \mathbb{1}_{\{x \in \Omega\}} f_{\Omega}(x) + \mathbb{1}_{\{x \in \Omega^c\}} f_{\Omega^c}(x),$$



[2] Korostelev and Tsybakov '93

Linear filtering

Larger kernels are more robust to noise but blur edges and boundaries.



True image portion | Searching zone and kernel support | Linear filter output

Linear filtering bounds

- If the kernel intersects the boundary, boundary is blurred

$$\mathbb{E}\left((\widehat{f}_i - f_i)^2\right) \asymp 1.$$

  $O(n^d h)$ pixels have kernels which intersect the boundary

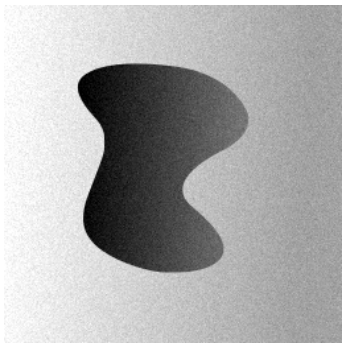
- If the kernel doesn't intersect the boundary,

$$\mathbb{E}\left((\widehat{f}_i - f_i)^2\right) \asymp h^{2\alpha} + \sigma^2(nh)^{-d}$$

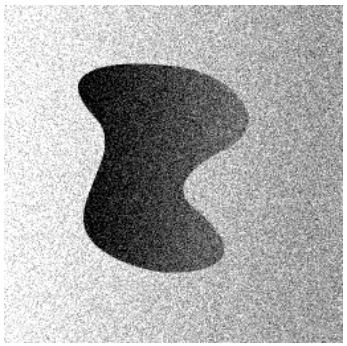
$$\mathcal{R}^{\mathrm{LF}} \asymp (\sigma^2/n^d)^{1/(d+1)}$$

This bound is independent of surface smoothness α!

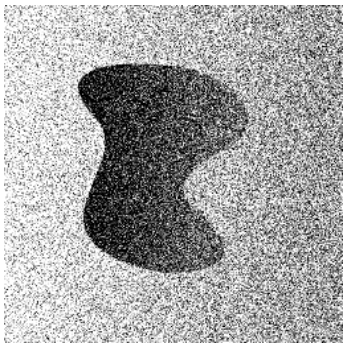
Linear filtering results



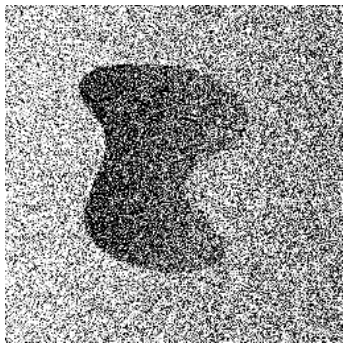
Noisy, MSE = 2.50e+01



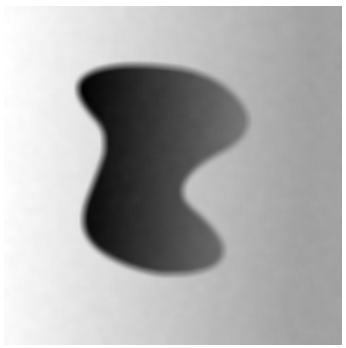
Noisy, MSE = 3.99e+02



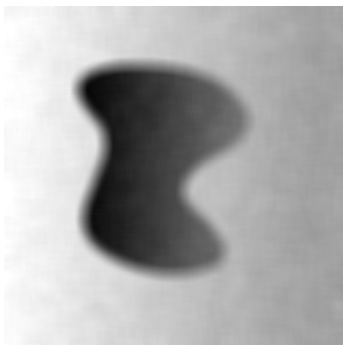
Noisy, MSE = 2.50e+03



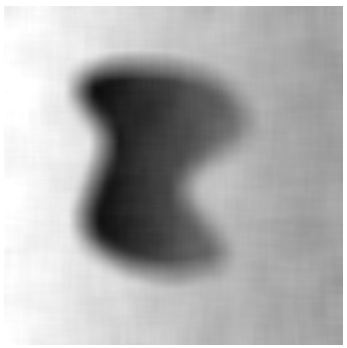
Noisy, MSE = 9.98e+03



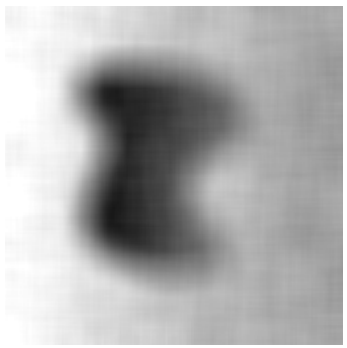
LF0, MSE = 3.52e+01



LF0, MSE = 7.78e+01

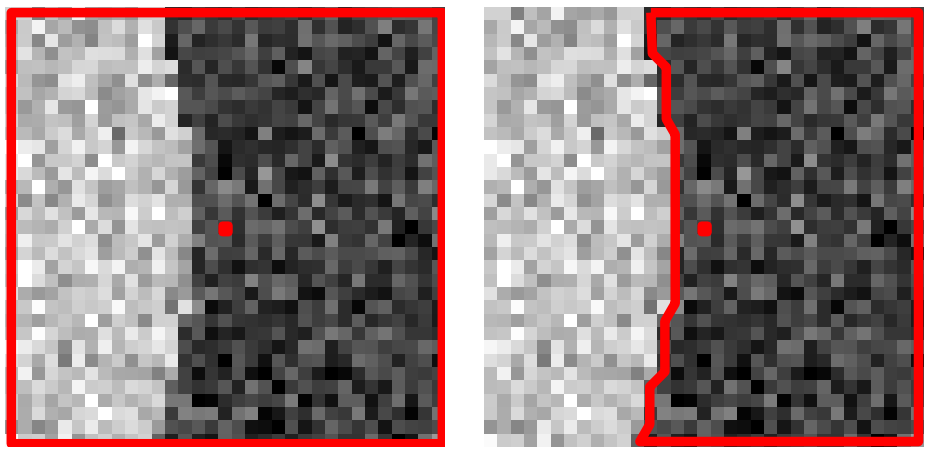


LF0, MSE = 1.51e+02



LF0, MSE = 2.43e+02

Membership oracle (the gold standard)



Kernel smoothing Membership oracle

We use local polynomial regression[3] of order $r \geq \lfloor \alpha \rfloor$ over the kernel domain.

[3] Fan & Gijbels '96, Hastie, Tibshirani & Friedman '09

Membership oracle bounds

The analysis is very similar to linear filters – only now the kernel never intersects the boundary. This gives

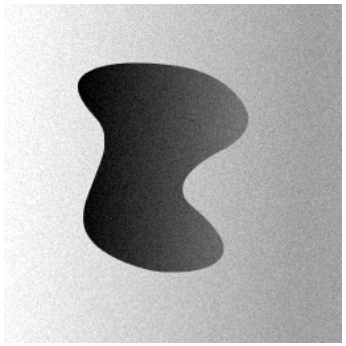
$$\mathcal{R}^{\mathrm{MO}} \asymp (\sigma^2/n^d)^{2\alpha/(d+2\alpha)}$$

Compare with linear filter, which had

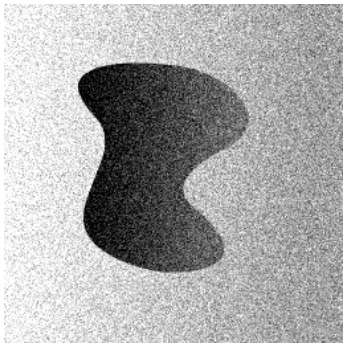
$$\mathrm{MSE} \asymp (\sigma^2/n^d)^{1/(d+1)}$$

for all α.

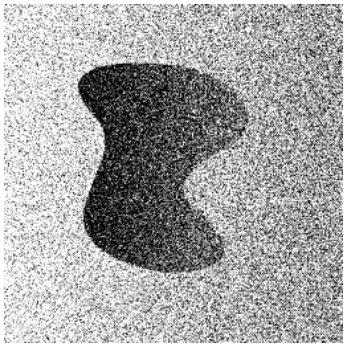
Membership oracle results



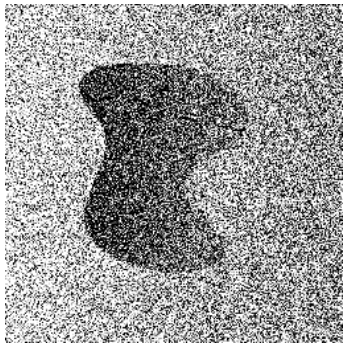
Noisy, MSE = 2.50e+01



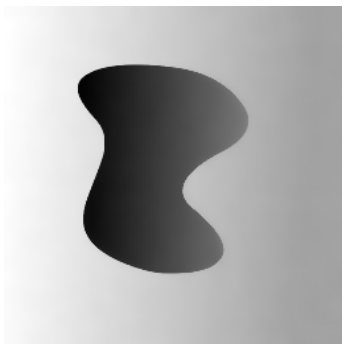
Noisy, MSE = 3.99e+02



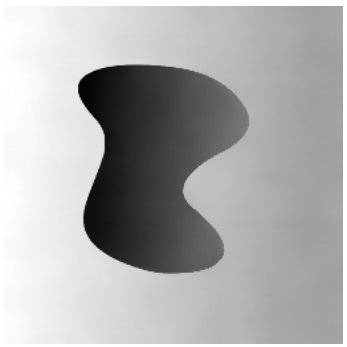
Noisy, MSE = 2.50e+03



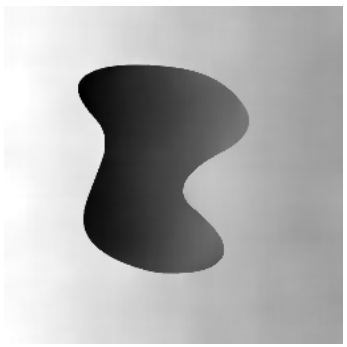
Noisy, MSE = 9.98e+03



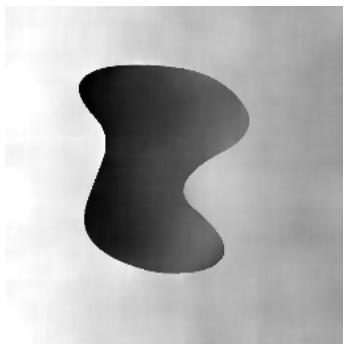
MO2, MSE = 9.57e-01



MO2, MSE = 2.37e+00



MO2, MSE = 6.09e+00

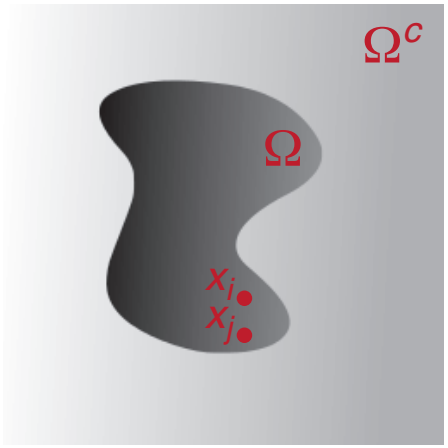


MO2, MSE = 1.96e+01

Yaroslavsky's filter bounds

Basic idea: if noise is small, then Yaroslavsky approximates the Membership Oracle.

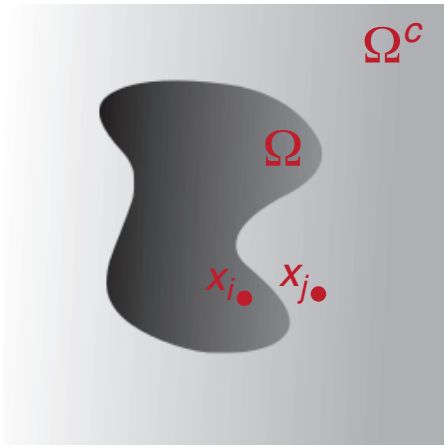
- f varies smoothly within Ω, so if $x_j \in \Omega$, we have an upper bound on $f_i - f_j$ and concentration bounds on $y_i - y_j$.



Yaroslavsky's filter bounds

Basic idea: if noise is small, then Yaroslavsky approximates the Membership Oracle.

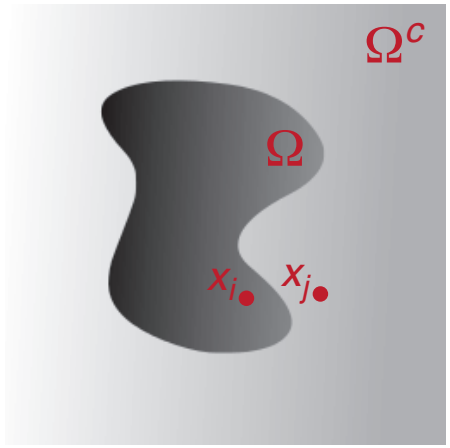
- f varies smoothly within Ω, so if $x_j \in \Omega$, we have an upper bound on $f_i - f_j$ and concentration bounds on $y_i - y_j$.
- We have a jump of height at least μ between Ω and Ω^c, so if $x_j \in \Omega^c$, we have a lower bound on $f_i - f_j$ and concentration bounds on $y_i - y_j$.



Yaroslavsky's filter bounds

Basic idea: if noise is small, then Yaroslavsky approximates the Membership Oracle.

- f varies smoothly within Ω, so if $x_j \in \Omega$, we have an upper bound on $f_i - f_j$ and concentration bounds on $y_i - y_j$.
- We have a jump of height at least μ between Ω and Ω^c, so if $x_j \in \Omega^c$, we have a lower bound on $f_i - f_j$ and concentration bounds on $y_i - y_j$.

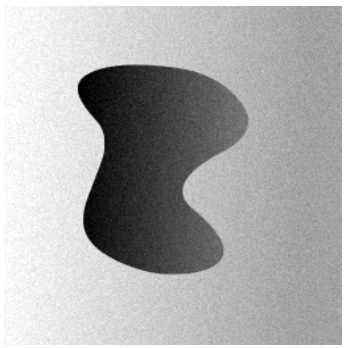


Thus if we choose h_y between these two bounds, we ensure that the y_j we select are in Ω with very high probability (for sufficiently small σ).

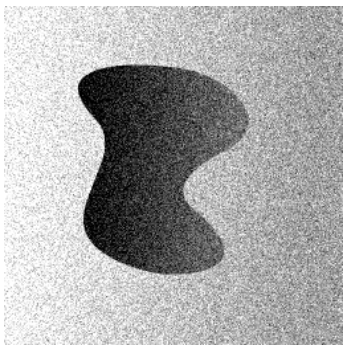
$$\mathcal{R}^{\mathrm{YF}} \leq (1 + o(1))\mathcal{R}^{\mathrm{MO}} \quad \text{for} \quad \sigma = O(1/\sqrt{\log n})$$

Yaroslavsky's filter results

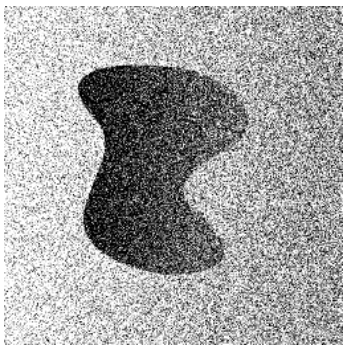
As predicted by theory, performance is very strong for low noise.



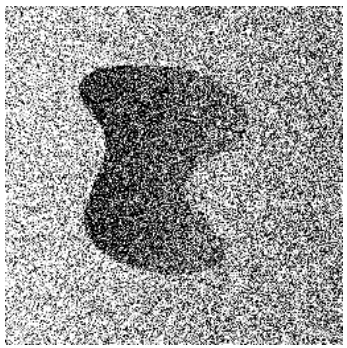
Noisy, MSE = 2.50e+01



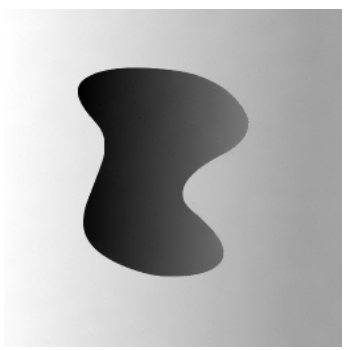
Noisy, MSE = 3.99e+02



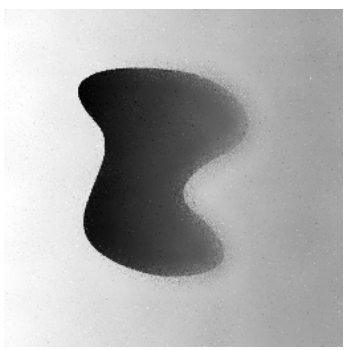
Noisy, MSE = 2.50e+03



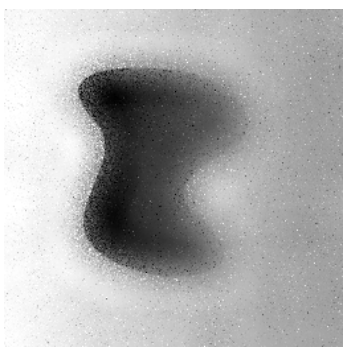
Noisy, MSE = 9.98e+03



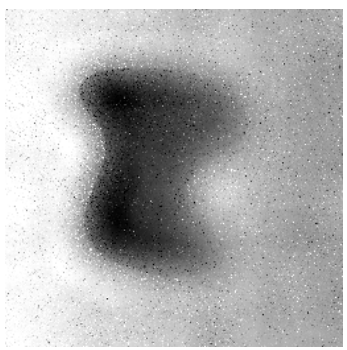
YF2, MSE = 9.37e-01



YF2, MSE = 1.90e+01



YF2, MSE = 1.46e+02

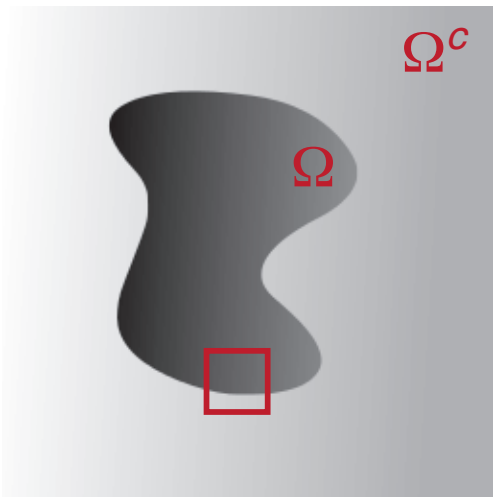


YF2, MSE = 2.98e+02

NLM bounds

Basic idea: patch distance approximates pixel distance, so NLM approximates membership oracle

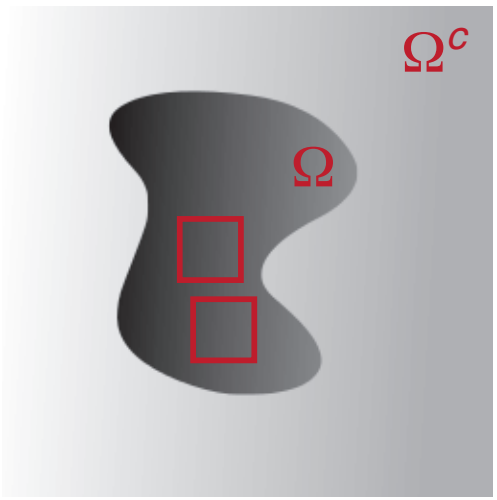
- If x_i is near the boundary, then the error can be $O(1)$, and there are $O(h_P n^d)$ such pixels, where h_P is the patch sidelength.



NLM bounds

Basic idea: patch distance approximates pixel distance, so NLM approximates membership oracle

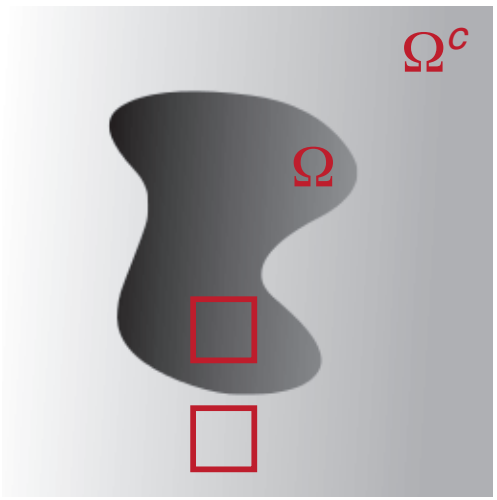
- If x_i is near the boundary, then the error can be $O(1)$, and there are $O(h_{\mathrm{P}} n^d)$ such pixels, where h_{P} is the patch sidelength.
- f varies smoothly within Ω, so if $\mathrm{P}_i, \mathrm{P}_j \subseteq \Omega$, we have an upper bound on $f_i - f_j$ and concentration bounds on $\|y_{\mathrm{P}_i} - y_{\mathrm{P}_j}\|_2$.



NLM bounds

Basic idea: patch distance approximates pixel distance, so NLM approximates membership oracle

- If x_i is near the boundary, then the error can be $O(1)$, and there are $O(h_P n^d)$ such pixels, where h_P is the patch sidelength.
- f varies smoothly within Ω, so if $P_i, P_j \subseteq \Omega$, we have an upper bound on $f_i - f_j$ and concentration bounds on $\|y_{P_i} - y_{P_j}\|_2$.
- We have a jump of height at least μ between Ω and Ω^c, so if $x_j \in \Omega^c$, we have a lower bound on $f_i - f_j$ and concentration bounds on $\|y_{P_i} - y_{P_j}\|_2$.



NLM bounds

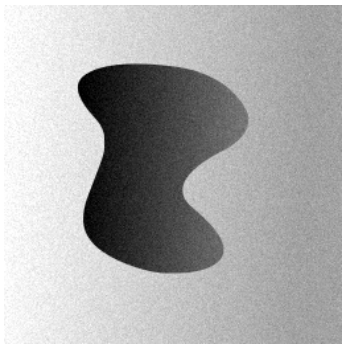
With high probability, NLM

- behaves like the membership oracle away from the boundary and
- behaves like the linear filter for a very small volume near the boundary.

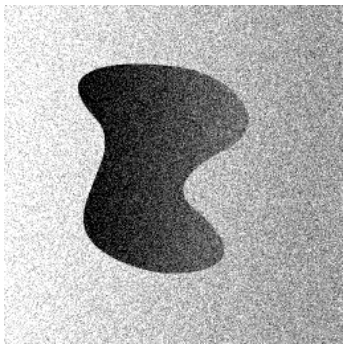
$$\mathcal{R}^{\mathrm{NLM}} \preceq \max\left(\frac{(\sigma^4 \log n)^{1/d}}{n}, (\sigma^2/n^d)^{2\alpha/(d+2\alpha)}\right)$$

NLM results

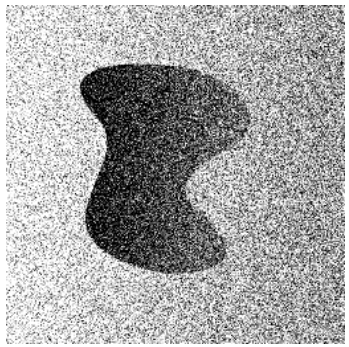
Because NLM uses entire patch to measure similarity between pixels, kernel weights are more robust to noise.



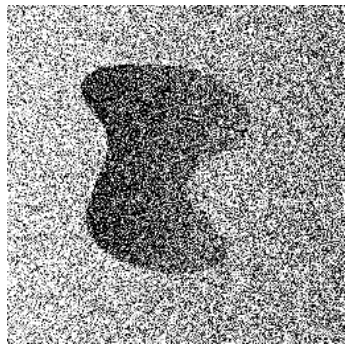
Noisy, MSE = 2.50e+01



Noisy, MSE = 3.99e+02

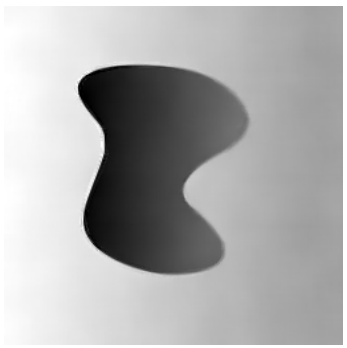


Noisy, MSE = 2.50e+03

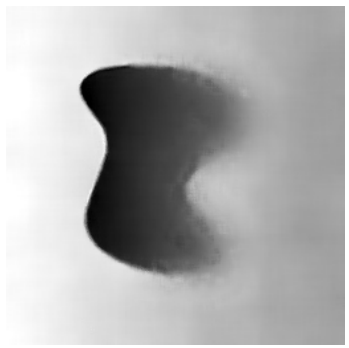


Noisy, MSE = 9.98e+03

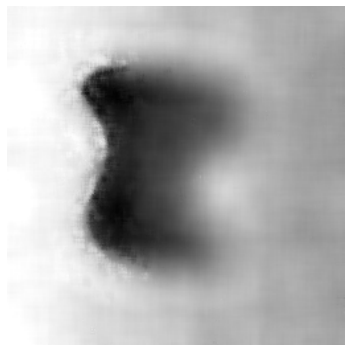
NLM2, MSE = 1.30e+00



NLM2, MSE = 4.92e+00



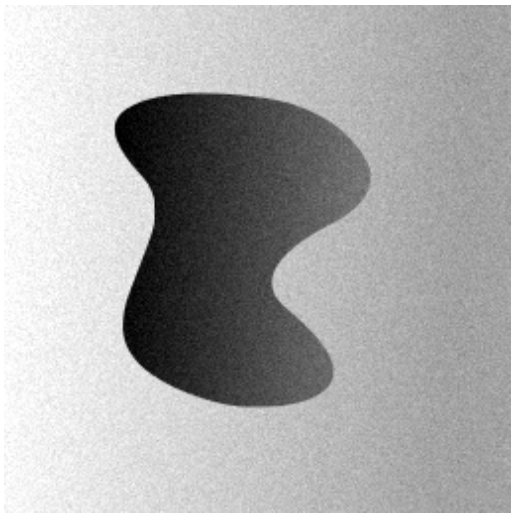
NLM2, MSE = 3.74e+01



NLM2, MSE = 1.37e+02

Examples, $\sigma = 5$

With low noise, all methods perform well.



Noisy, MSE = 2.50e+01

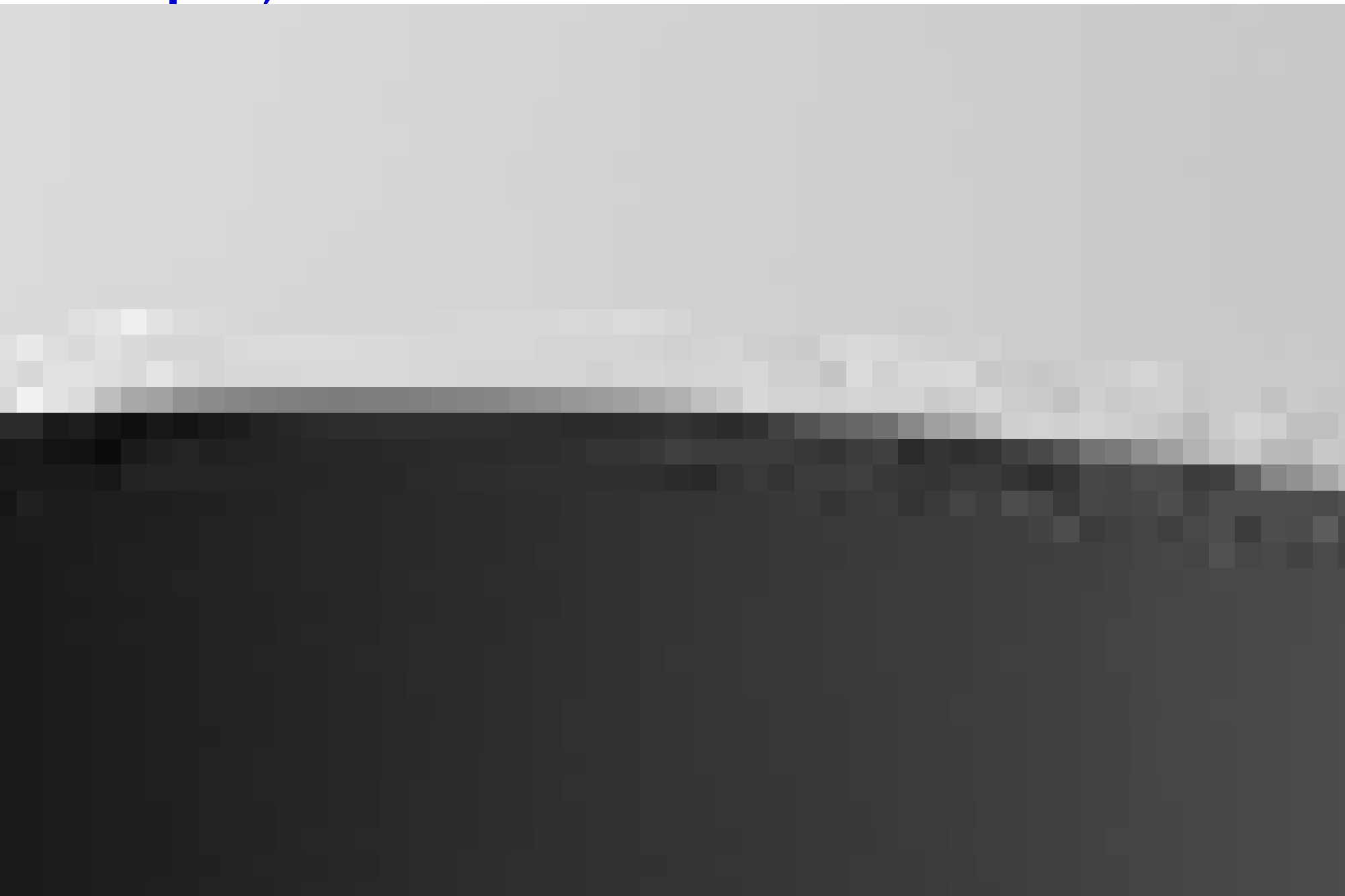
LF2, MSE = 7.21e+01

YF2, MSE = 9.37e-01

NLM2, MSE = 1.30e+00

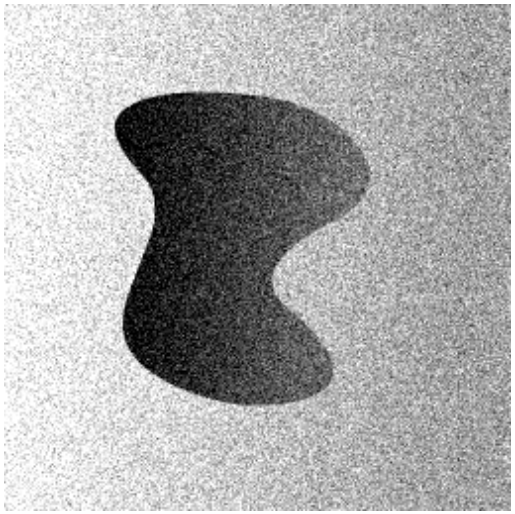
MO2, MSE = 9.57e-01

Examples, $\sigma = 5$

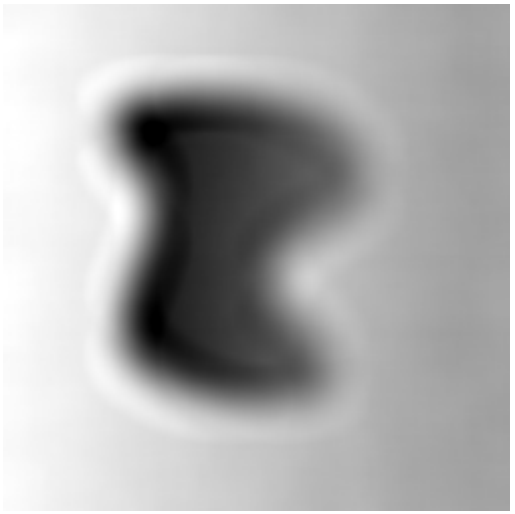


Examples, $\sigma = 20$

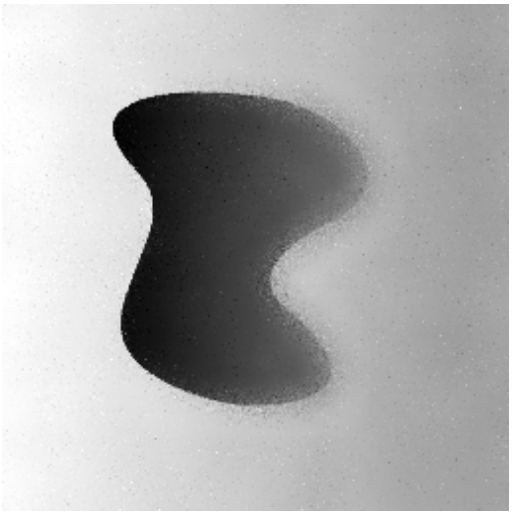
As noise increases, we first see the linear filter start to break down.



Noisy, MSE = 3.99e+02



LF2, MSE = 1.40e+02



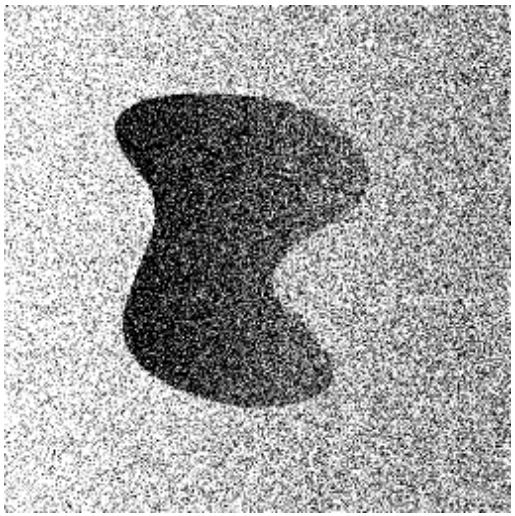
YF2, MSE = 1.90e+01

NLM2, MSE = 4.92e+00

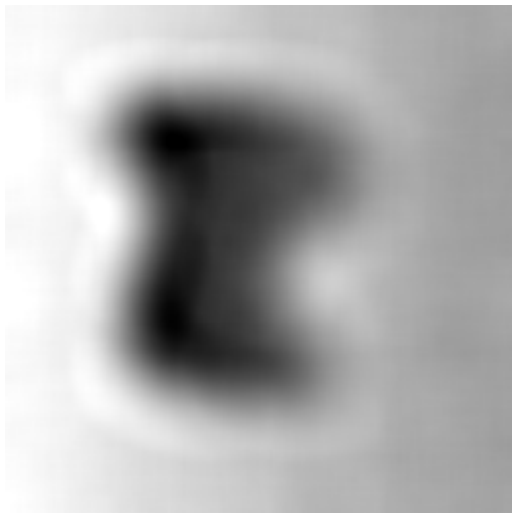
MO2, MSE = 2.37e+00

Examples, $\sigma = 50$

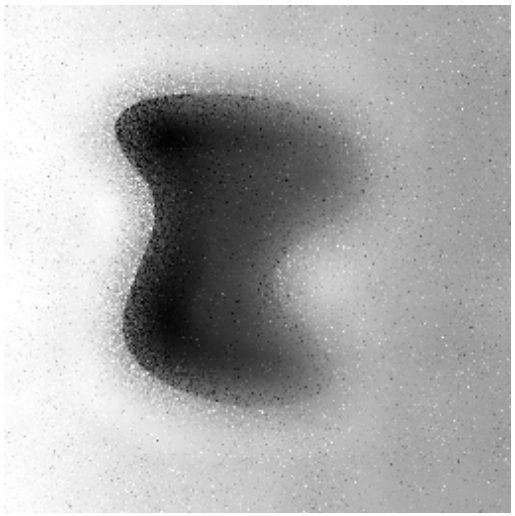
With even more noise, Yaroslavsky's filter starts to perform poorly.



Noisy, MSE = 2.50e+03



LF2, MSE = 2.11e+02



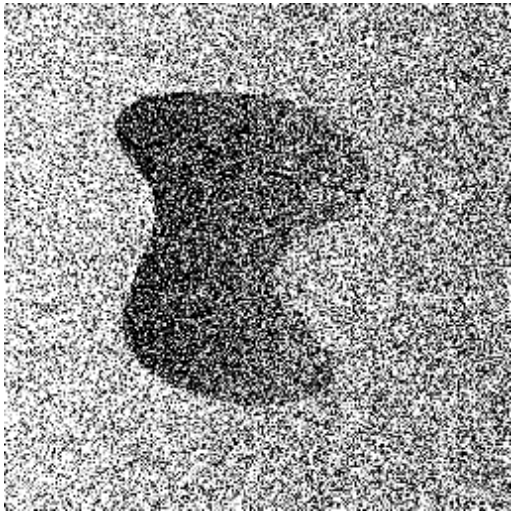
YF2, MSE = 1.46e+02

NLM2, MSE = 3.74e+01

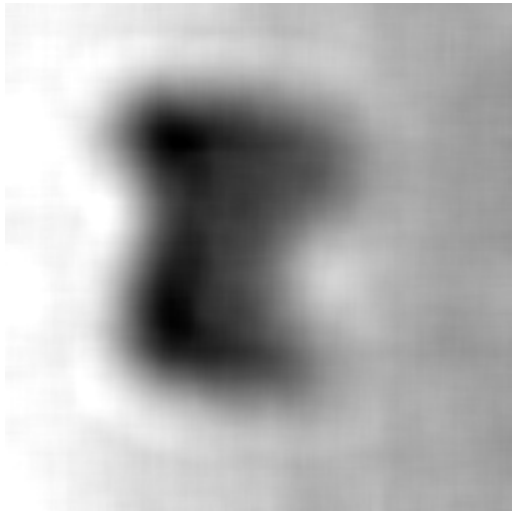
MO2, MSE = 6.09e+00

Examples, $\sigma = 100$

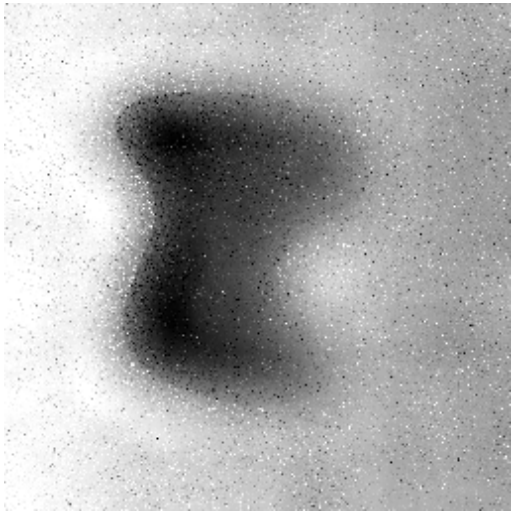
We also see how performance varies with the size of the "jump".



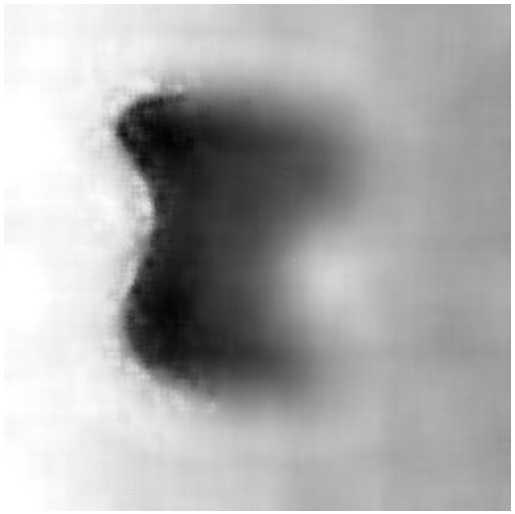
Noisy, MSE = 9.98e+03



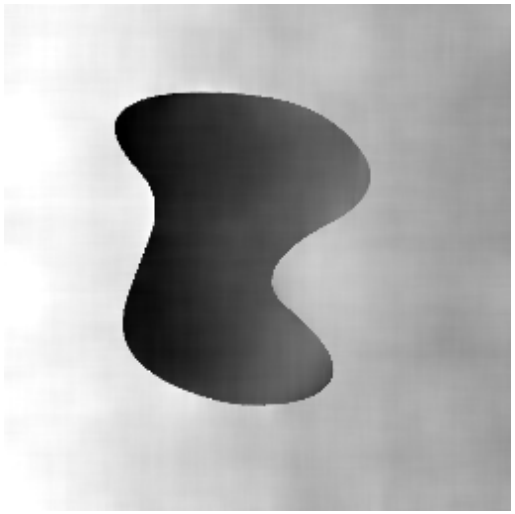
LF2, MSE = 2.29e+02



YF2, MSE = 2.98e+02



NLM2, MSE = 1.37e+02



MO2, MSE = 1.96e+01

Repeating patterns

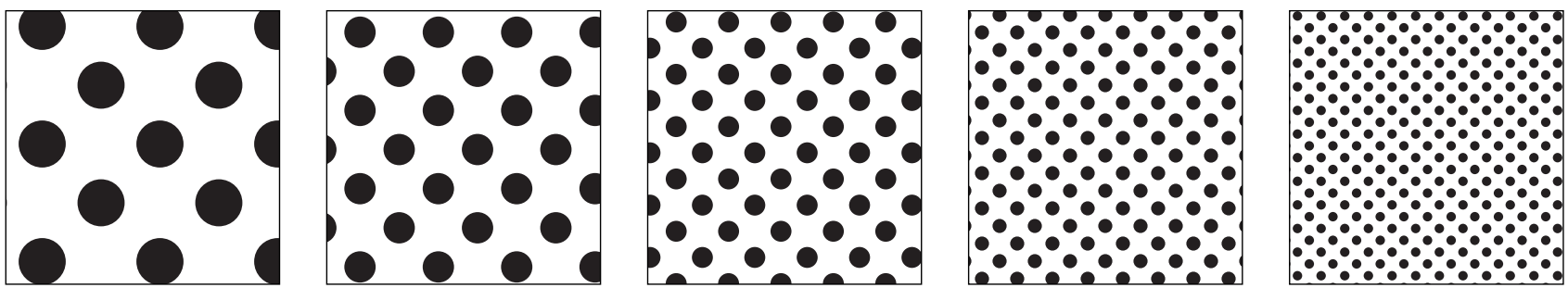
As before,

$$f(x) = \mathbb{1}_{\{x \in \Omega\}} f_\Omega(x) + \mathbb{1}_{\{x \in \Omega^c\}} f_{\Omega^c}(x),$$

but now

$$\Omega = (0,1)^d \cap \bigcup_{v \in a\mathbb{Z}^d} (\Xi + v)$$

where a is the pattern period and $a \to 0$ as $n \to \infty$. This function class is like the cartoon class, but the underlying scene (especially the frequency of repetition) scales with n.



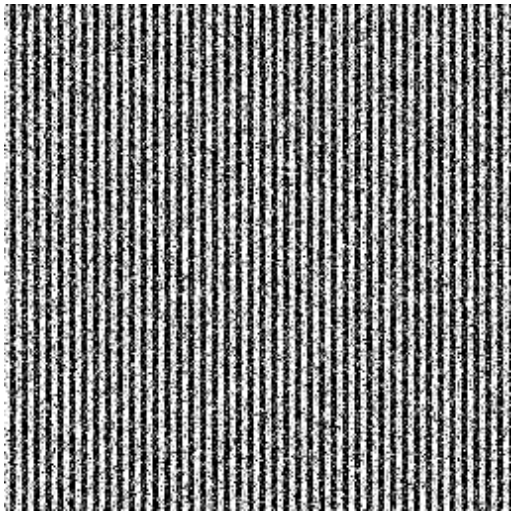
n small $\longrightarrow$ n medium $\longrightarrow$ n large

Performance bounds for patterns

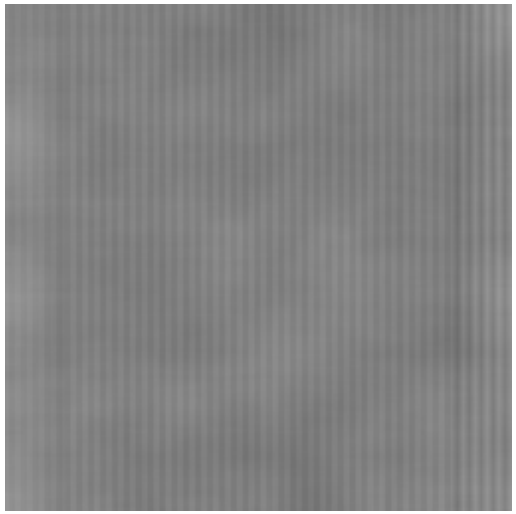
Consider $f \in \mathcal{F}^{\text{pattern}}$. Assume the volumes of Ω and Ω^c are comparable.

- MO with $h = h^{\text{MO}}$ achieves an MSE of order $\mathcal{R}^{\text{MO}}$
- YF with $h^{\text{MO}}, h_y \asymp 1$ achieves an MSE of order $\mathcal{R}^{\text{MO}}$ if the noise is low
- NLM with bandwidths $h = h^{\text{MO}}, h_y = h_y^{\text{NLM}}$ and patch size $h_{\text{P}}^{\text{NLM}}$ achieves an MSE of order $(na)^d \mathcal{R}^{\text{MO}}$ if the pattern is sufficiently "strong" (foreground-centered patches must be distinct from background-centered patches).

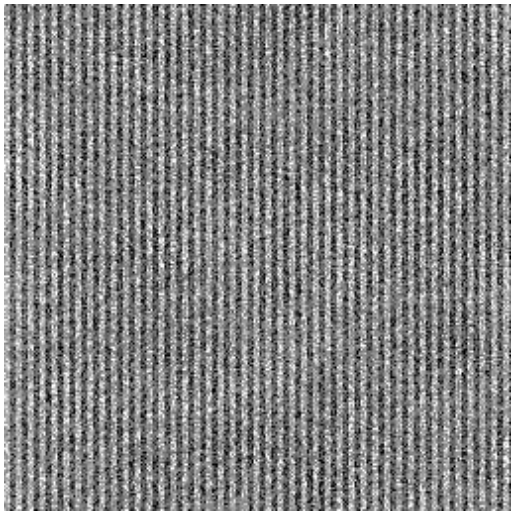
Examples, $\sigma = 100$



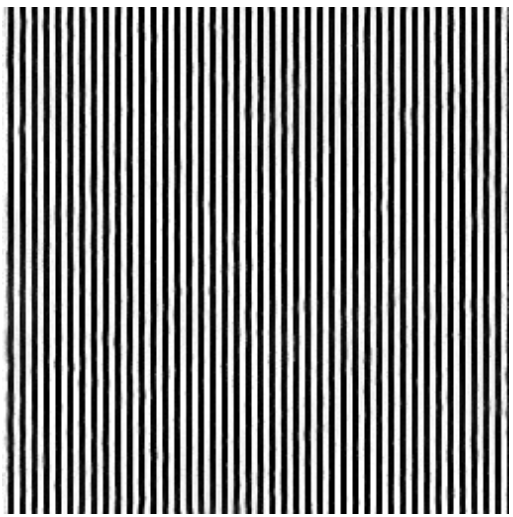
Noisy, MSE = 9.98e+03



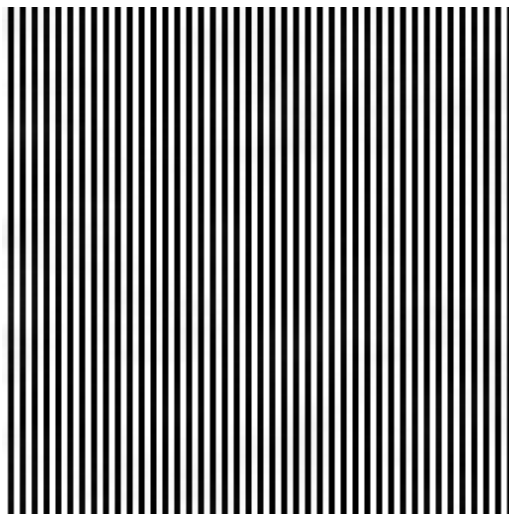
LF2, MSE = 1.71e+04



YF2, MSE = 8.87e+03

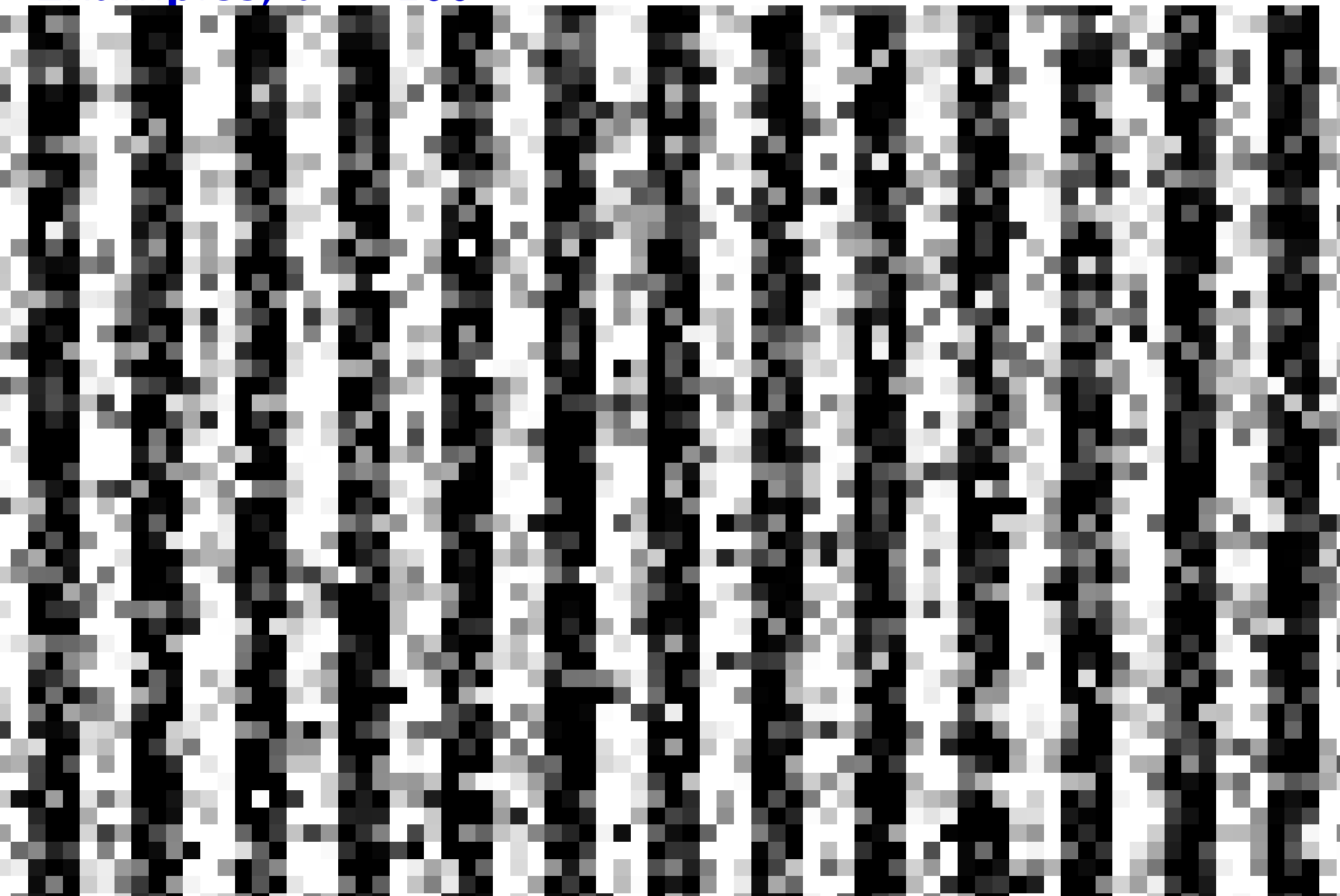


NLM2, MSE = 2.33e+02

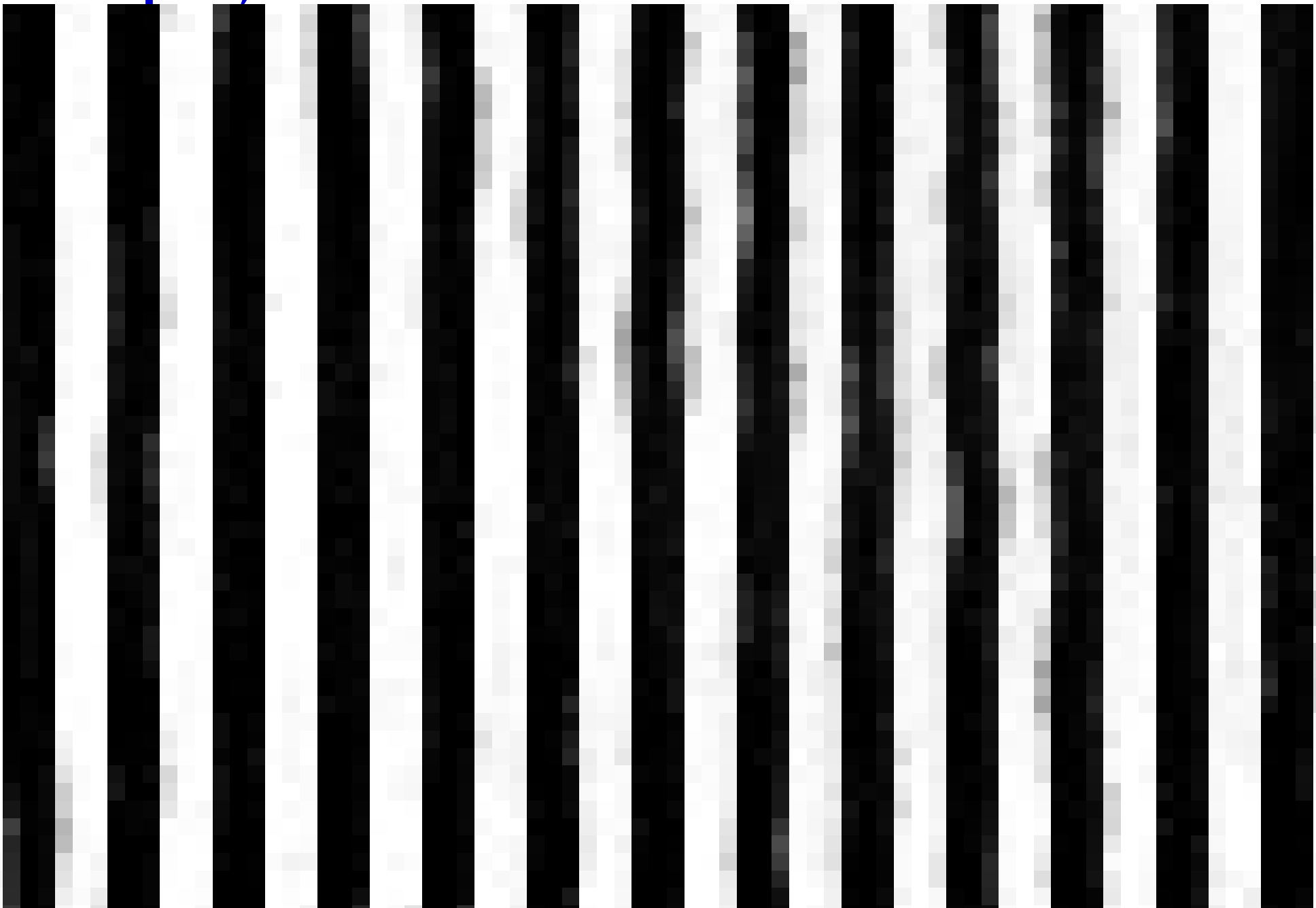


MO2, MSE = 2.28e+01

Examples, $\sigma = 100$



Examples, $\sigma = 100$



Examples, $\sigma = 20$

Noisy, MSE = 3.99e+02

LF2, MSE = 9.01e+02

YF2, MSE = 2.44e+02

NLM2, MSE = 1.31e+02

MO2, MSE = 4.65e+01

Insights lead to better methods

Recall

$$\widehat{f}_i = \sum_j w_{i,j} y_j$$

Consider a weight oracle (WO), where $w_{i,j} = K_h(x_i, x_j) L_{h_y}(f_i, f_j)$.

- **This estimator is minimax optimal.**
- Every method we've considered tries to estimate $L_{h_y}(f_i, f_j)$
  - Yaroslavsky's filter uses y_i, y_j in place of f_i, f_j.
  - NLM uses distance between noisy patches
- What if we compute a coarse estimate of f, and use that to estimate $L_{h_y}(f_i, f_j)$?

A preprocessed Yaroslavsky filter

What if we compute a coarse estimate of f, and use that to estimate $L_{h_y}(f_i, f_j)$?

- A **faster** variant of NLM (NLM-Av.) looks at the difference between the **mean** of two patches to estimate $L_{h_y}(f_i, f_j)$. This does as well as NLM for cartoon images both theoretically and empirically.
- This is equivalent to applying a linear filter to noisy image y to get $\widehat{f}^{\mathrm{LF}}$, and then applying Yaroslavsky's filter with the weights set using $\widehat{f}^{\mathrm{LF}}$.
- We could also use Yaroslavsky's filter with weights set using a wavelet-denoised image.
- We could also use Yaroslavsky's filter with weights set using a curvelet-denoised image.

Example, $\sigma = 50$

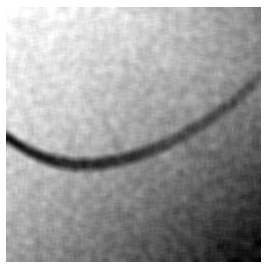
With even more noise, Yaroslavsky's filter starts to perform poorly.

Original

Noisy,
MSE=2.51e+03

YF,
MSE=1.29e+02

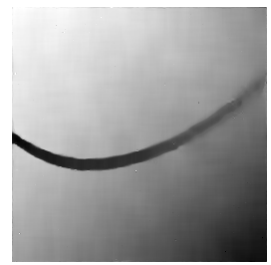
NLM,
MSE=3.73e+01



LF,
MSE=9.13e+01

NLM-Av.,
MSE=2.69e+01

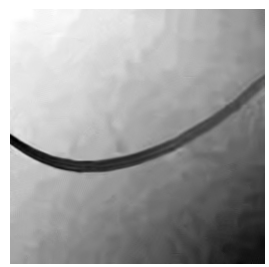
WavCS,
MSE=7.89e+01



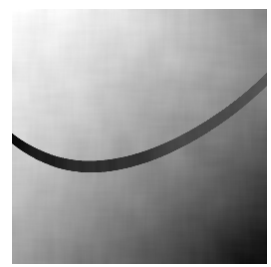
YFWavCS,
MSE=2.52e+01

Curvelet,
MSE=7.52e+01

YFCurvelet,
MSE=1.59e+01



BM3D,
MSE=2.29e+01



WO,
MSE=9.16e+00

Example, $\sigma = 50$

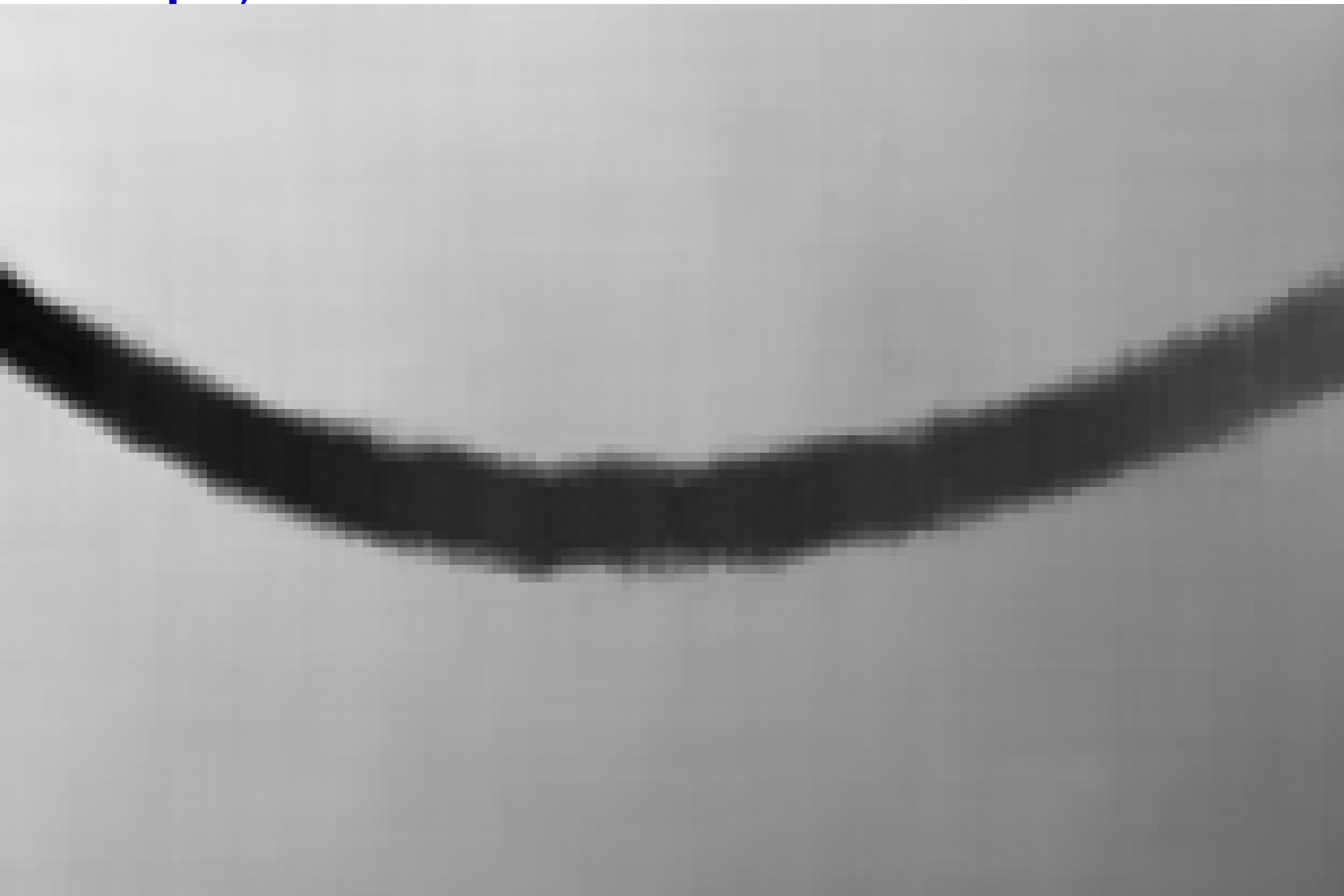
Example, $\sigma = 50$

Example, $\sigma = 50$

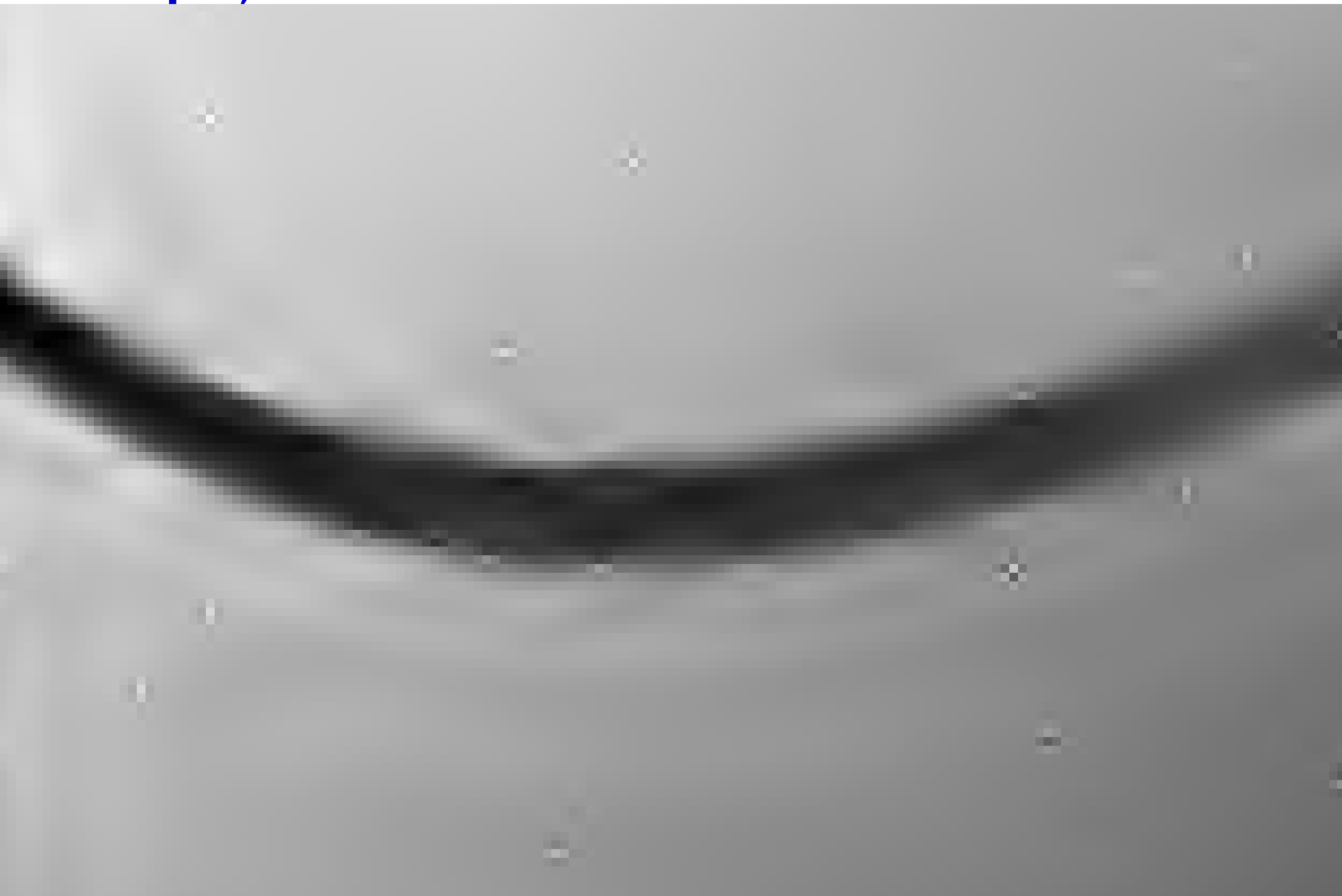
Example, $\sigma = 50$

Example, $\sigma = 50$

Example, $\sigma = 50$



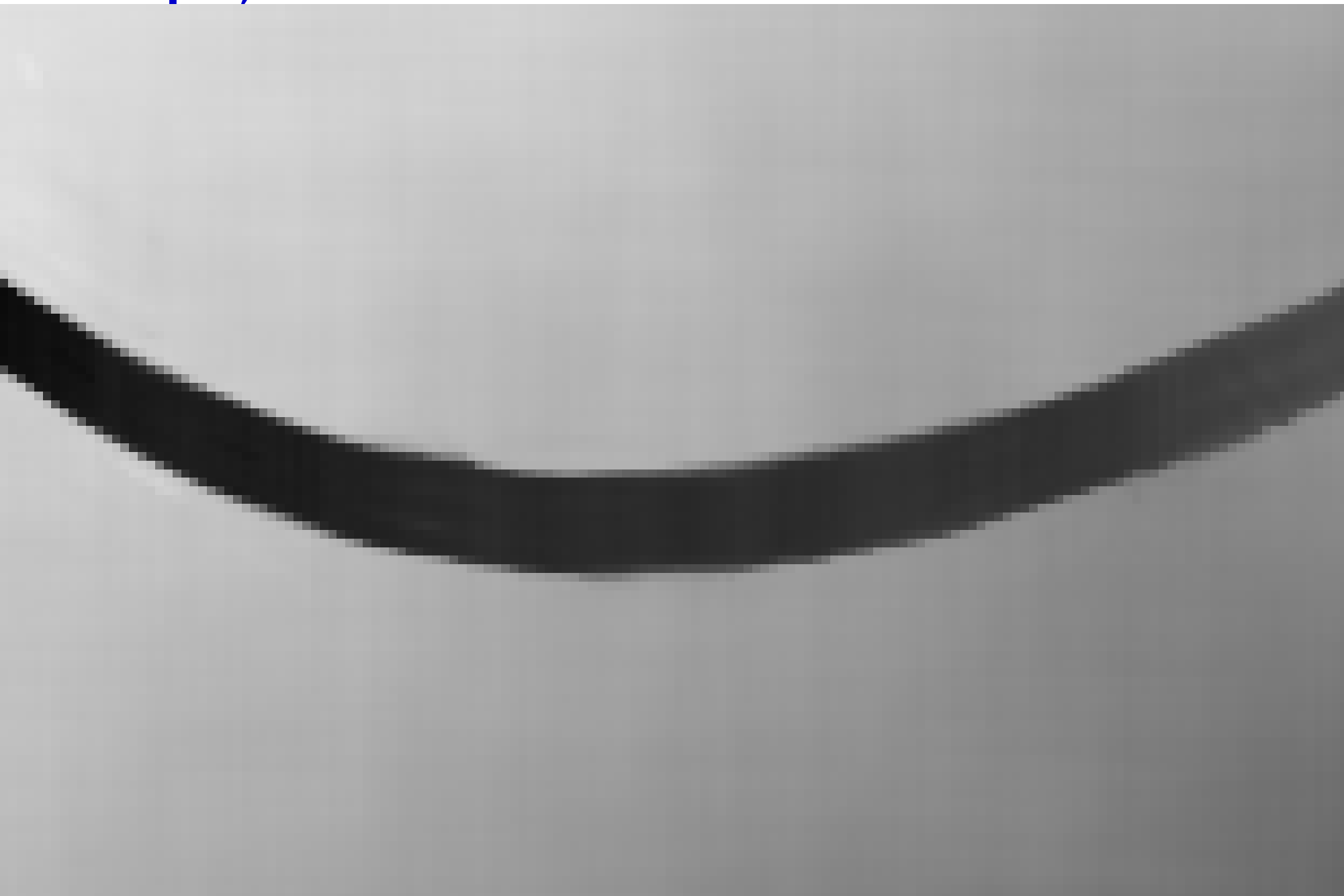
Example, $\sigma = 50$



Example, $\sigma = 50$

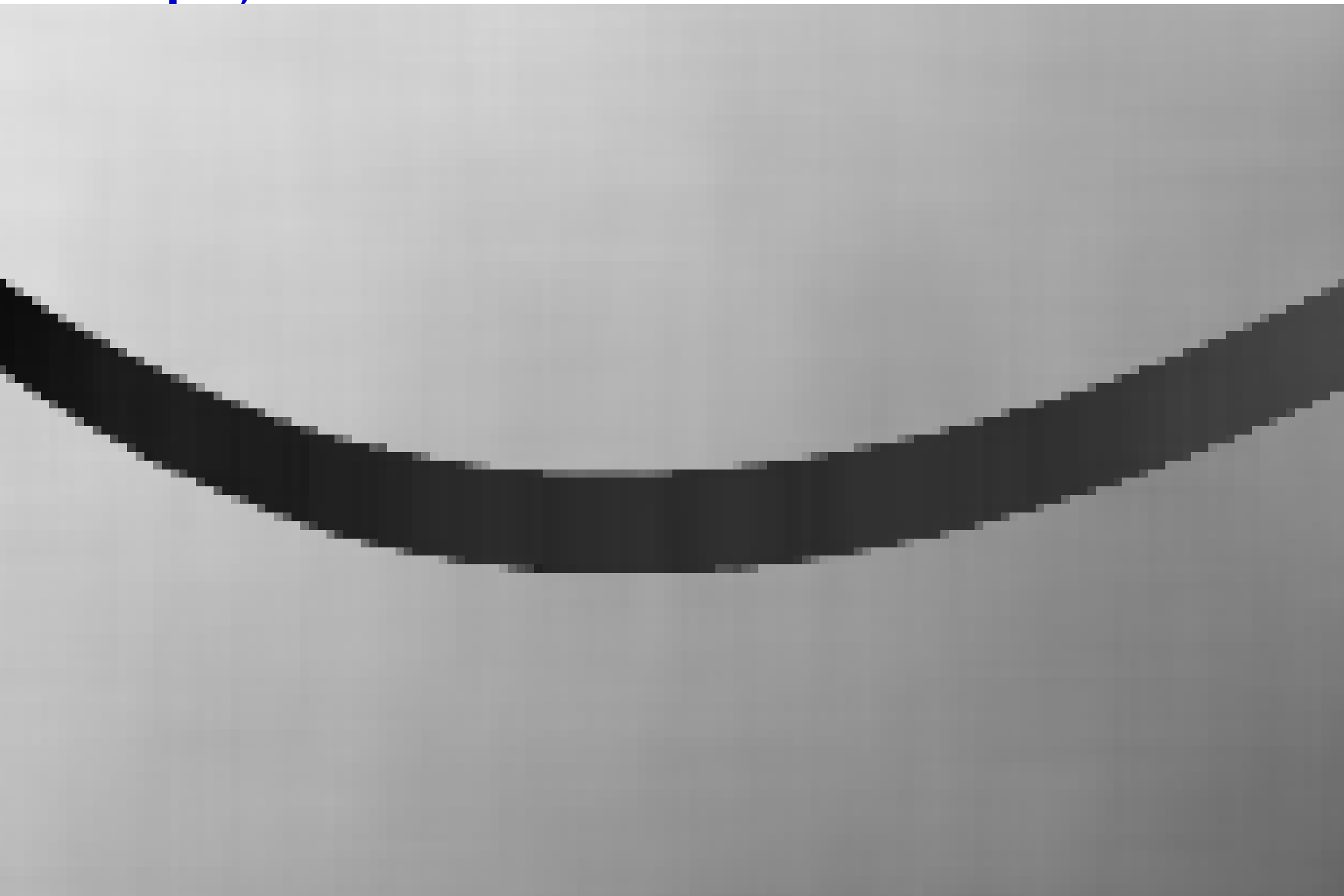
Example, $\sigma = 50$

Example, $\sigma = 50$



Example, $\sigma = 50$

Example, $\sigma = 50$



Conclusions

- Novel membership oracle gives new insight into key limitations of adaptive filtering methods.
- The classical Yaroslavsky's method behaves optimally at low noise levels.
- NLM mimics Yaroslavsky's filter, but uses patches to robustly determine pixel similarity.
- Novel image class describes repeating patterns and redundancy not present in classical image models and not well-suited to methods like wavelet thresholding – we show how NLM performs well in this setting.
- Theoretical insights lead to new methods which can reduce artifacts in other noise removal techniques used in neuroimaging

Thank you.

http://arxiv.org/abs/1112.4434