Some Denoising Techniques

1. Gaussian (Gabor) filter
2. Anisotropic PDE
3. Rudin-Osher-Fatemi TV
4. SUSAN filter
5. Wiener
6. wavelets
7. DUDE
8. bilateral
9. nonlocal means

Bilateral Filter

Use some weighted average

$$\widehat{x}_k = \frac{\sum_n w_{kn} g_{k-n}}{\sum_n w_{kn}}$$

1. $w^s_{kn} = e^{-\frac{d^2_{k,k-n}}{2\sigma_s^2}}$ Gaussian, physical distance
2. $w^r_{kn} = e^{-\frac{(g_k - g_{k-n})^2}{2\sigma_r^2}}$ Gaussian, gray value distance

$$w_{kn} = w^s_{kn} w^r_{kn}$$

σ_s , σ_r large implies uniform non-adaptive , degrades signal

σ_s , σ_r small implies no smoothing

Nonlinear Means-Global Filter

Buades-Coll&Morel

$$NL_h u(x) = \frac{1}{D(x)} \int K\left(\frac{G_a * [u(x+.) - u(y+.)]^2}{h}\right) u(y) dy$$

for example, $$NL_h u(x) = \frac{\int e^{-\frac{G_a * [u(x+.)-u(y+.)]^2}{h^2}} u(y)dy}{\int e^{-\frac{G_a * [u(x+.)-u(y+.)]^2}{h^2}} dy}$$

where $$G_a * [u(x+.) - u(y+.)]^2 = \int G_a(t) [u(x+t) - u(y+t)]^2 \, dt$$

where h is a filter parameter (usually dependence on variance of the noise), and D is a normalizing factor. So $G_a * [u(x_+.) - u(y+.)]^2$ is the convolution of the squared difference of the shifted signals with a Gaussian Kernel.

Thus, $u(y)$ is used to denoise $u(x)$ if the local pattern near $u(y)$ is similar to the local pattern near $u(x)$.

Method Noise: Decompose the image v

$$v = D_h v + n(D_h, v)$$

where

1. $D_h v$ is smoother than v
2. $n(D_h, v)$ is the estimated noise.

So

$$n(D_h, v) = v - D_h v$$

we want $n(D_h, v)$ to look like white noise and D_h should be close to the identity for smooth enough images.

Fig. 1. A digital image with standard deviation 55, the same with noise added (standard deviation 3), the SNR therefore being equal to 18, and the same with SNR slightly larger than 2. In this second image, no alteration is visible. In the third, a conspicuous noise with standard deviation 25 has been added, but, surprisingly enough, all details of the original image still are visible.

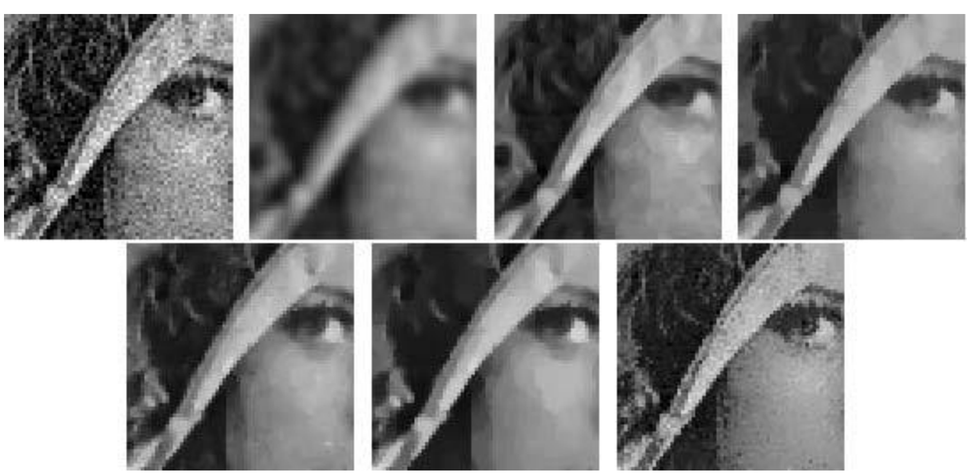


Fig. 3. Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation 20), Gaussian convolution, anisotropic filter, total variation minimization, Tadmor–Nezzar–Vese iterated total variation, Osher et al. iterated total variation, and the Yaroslavsky

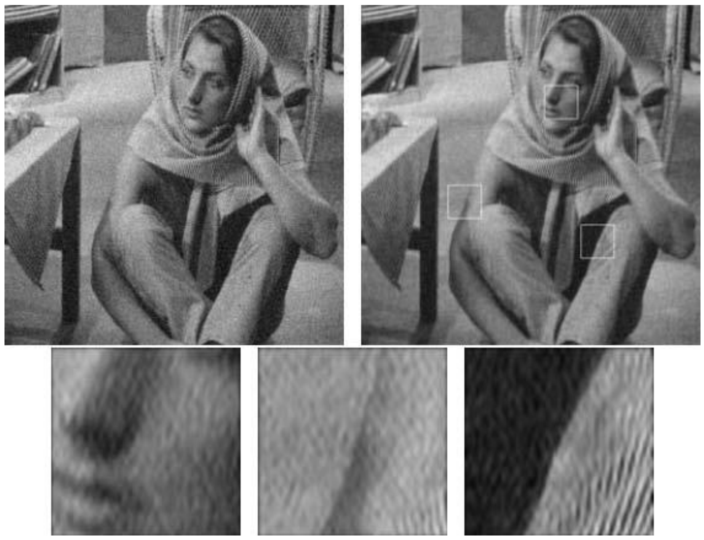


Fig. 4. Fourier–Wiener filter experiment. Top left: Degraded image by an additive white noise of σ =15. Top right: Fourier–Wiener filter solution. Bottom: Zoom on three different zones of the solution. The image is filtered as a whole, and therefore a uniform texture is spread all over the image.

Fig. 5. Denoising experiment on a natural image. From left to right and from top to bottom: noisy image (standard deviation 20), Fourier–Wiener filter (ideal filter), the DCT empirical Wiener filter, the wavelet hard thresholding

•

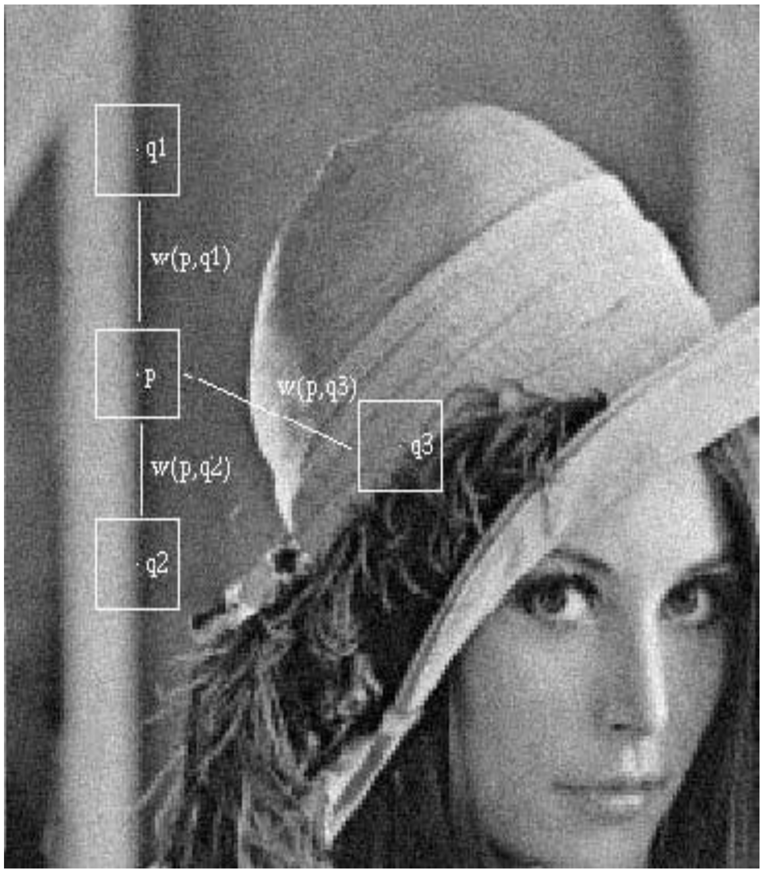


Fig. 6. q1 and q2 have a large weight because their similarity windows are similar to that of p. On the other side the weight w(p, q3) is much smaller because the intensity grey values in the similarity windows are very different.

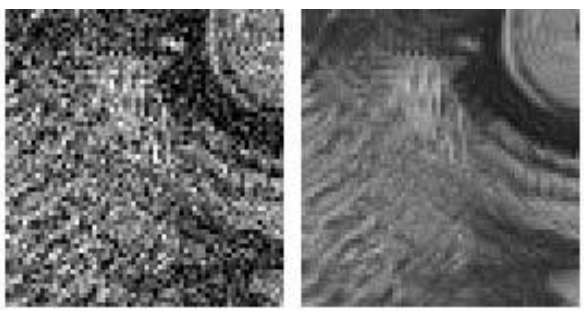


FIG. 10. *NL-means denoising experiment with a natural image. Left: Noisy image with standard deviation 35. Right: Restored image.*

Fig. 9. NL-means denoising experiment with a natural image. Left: Noisy image with standard deviation 20 Right: Restored image.

FIG. 11. *NL-means denoising experiment with a color image. Left: Noisy image with standard deviation 15 in every color component. Right: Restored image.*

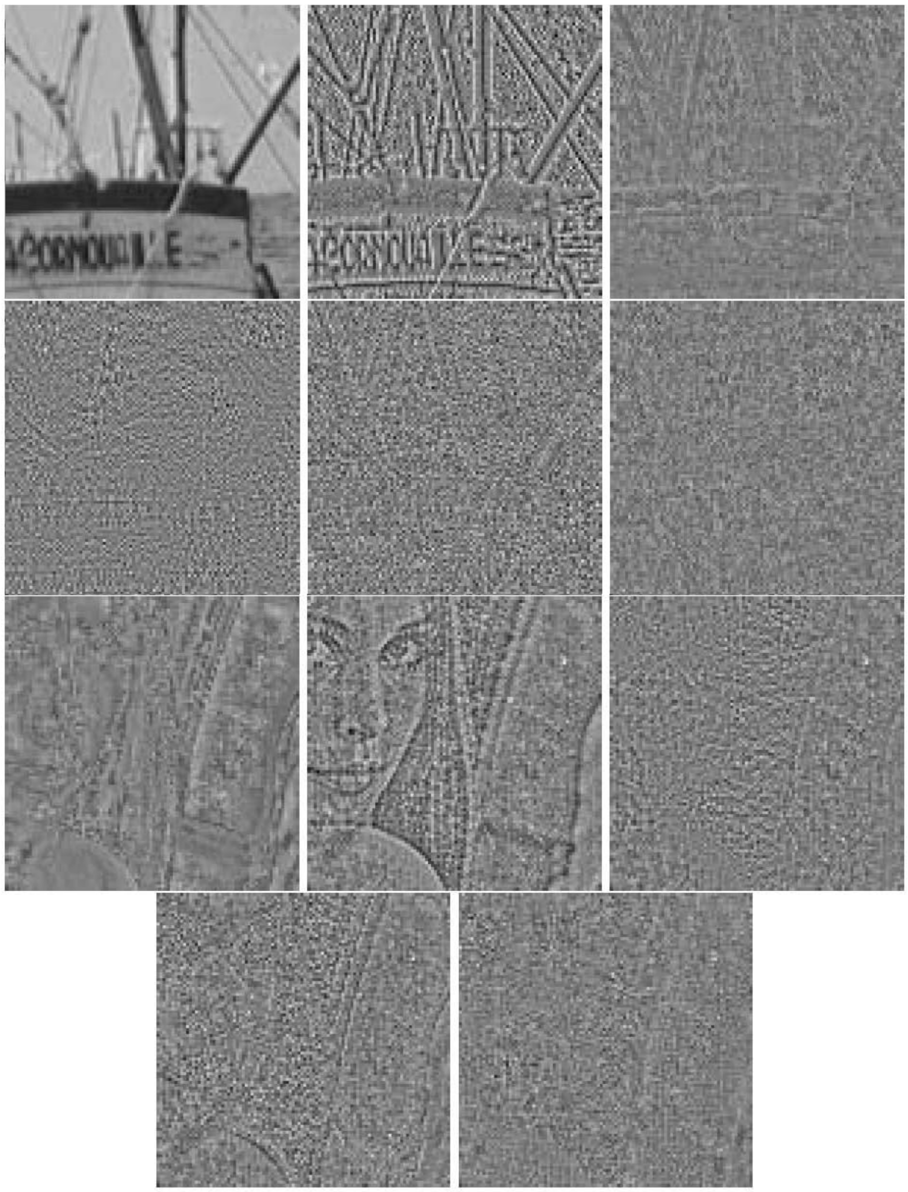


FIG. 15. *Image method noise. From left to right and from top to bottom: original image, Gaussian convolution, mean curvature motion, total variation, Tadmor–Nezzar–Vese iterated total variation, Osher et al. total variation, neighborhood filter, soft TIWT, hard TIWT, DCT empirical Wiener filter, and the NL-means algorithm.*

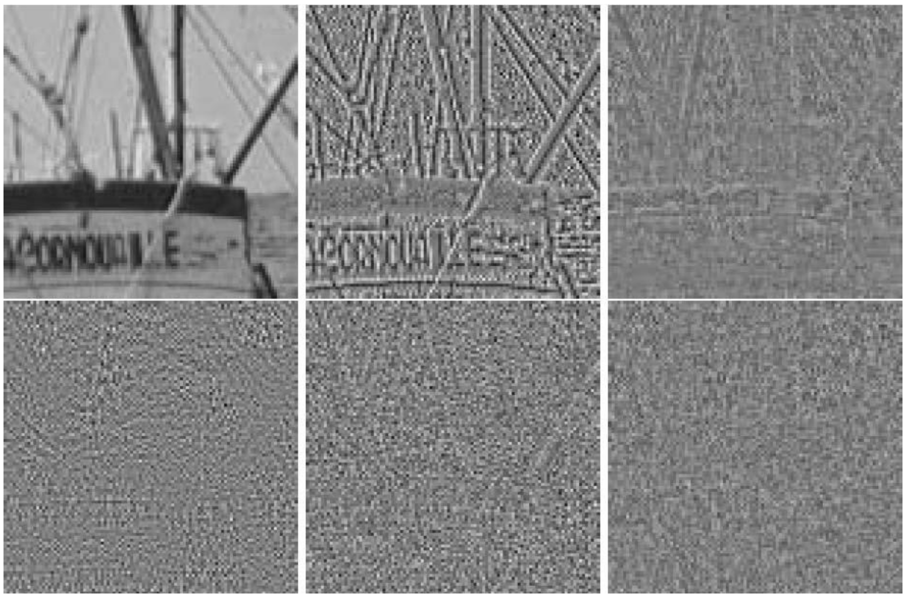
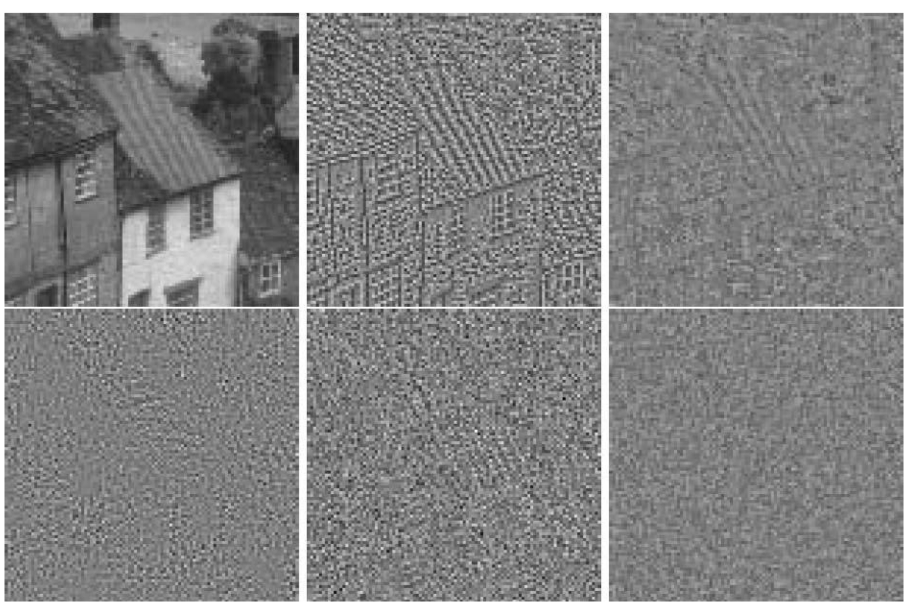


Fig. 16. Image method noise. From left to right and from top to bottom: original image, total variation,



neighborhood filter, hard TIWT, DCT empirical Wiener filter, and the NL-means algorithm.

Fig. 17. Image method noise. From left to right and from top to bottom: original image, total variation, neighborhood filter, hard TIWT, DCT empirical Wiener filter, and the NL-means algorithm.

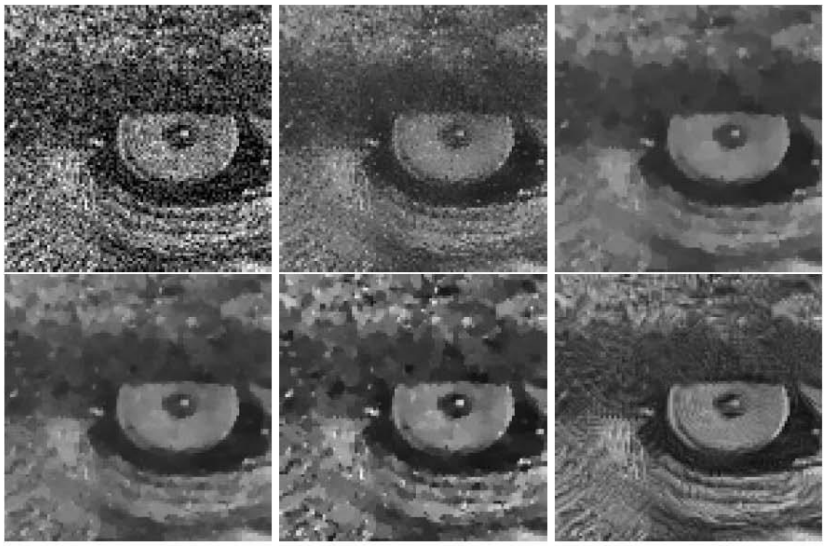


FIG. 19. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation 35), neighborhood filter, total variation, Tadmor–Nezzar–Vese iterated total variation, Osher et al. iterated total variation, and the NL-means algorithm.*

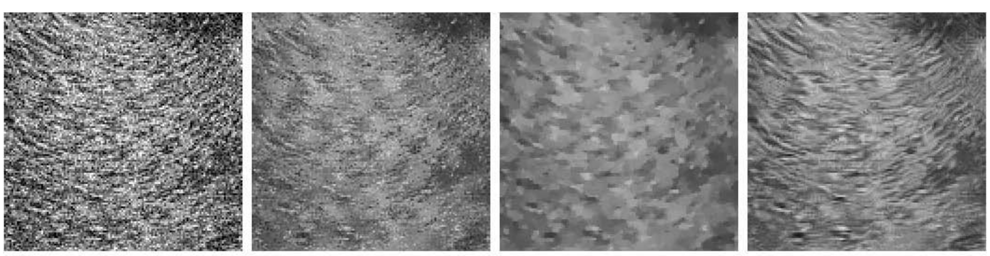


FIG. 20. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation 35), neighborhood filter, total variation, and the NL-means algorithm.*