Image and movie denoising by nonlocal means

Antoni Buades Bartomeu Coll Jean-Michel Morel

Abstract

Neighborhood filters are image and movie filters which reduce the noise by averaging similar pixels. The object of the paper is to present a unified theory of these filters and reliable criteria to compare them to other classes of filters. First a CCD noise model will be presented justifying this class of algorithm. A classification of neighborhood filters is proposed, including classical image and movie denoising methods and a new one, the nonlocal-means (NL-means). In order to compare denoising methods three principles will be introduced. The first principle, "method noise", specifies that only noise must be removed from an image. Second a "noise to noise" principle states that a denoising method must transform a white noise into a white noise. This principle characterizes artifact-free methods. A third principle, the "statistical optimality", permits to compare the performance of neighborhood filters. The three principles will be applied to compare ten different image and movie denoising methods. It will be first shown that only wavelet thresholding methods and NL-means give an acceptable method noise. Second, that neighborhood filters are the only ones to satisfy the "noise to noise" principle. Third, that among them NL-means is closest to statistical optimality. A particular attention is paid to movie denoising methods. Motion compensated denoising methods turn out to be neighborhood filters where the neighborhood is constrained to stay on a single trajectory. It is demonstrated that this constraint is harmful for denoising purposes and that space-time NL-means preserves more movie details. The paper ends with a careful implementation of NL-means and more examples.

1 Introduction

We shall call *neighborhood filters* all image and movie filters which reduce the noise by averaging similar pixels. General CCD noise models (briefly presented in Section 2) imply that noise in digital images and movies is signal dependent. Fortunately two pixels which received the same energy from the outdoor scene undergo the same kind of perturbations and therefore have the same noise model. Under the fairly general assumption that at each energy level the noise model is additive and white, denoising can be achieved by first finding out the pixels which received the same original energy and then averaging their observed grey levels. This observation has led to the wide class of neighborhood filters classified in [32]. Since the original image value is lost these filters proceed by picking for each pixel i the set of pixels $J(i)$ close i and such that $u(j)$ is close to $u(i)$. Thus the pair $(j, u(j))$ belongs to a space and grey level neighborhood of $(i, u(i))$; hence the general name of neighborhood filter.

Neighborhood filters proceed by replacing $u(i)$ by the average $NFu(i) = \sum_{j \in J(i)} u(j)$. (Depending on the noise model other statistical estimates are of course possible like the median, etc.) Under the assumption that pixels $j \in J(i)$ indeed received the same original energy as i, $NFu(i)$ is a denoised version of $u(i)$. The more famous neighborhood filters are Lee's σ-filter

[14], SUSAN [25] and the bilateral filter [27] where the neighborhoods are gaussian in space and grey level.

In a recent communication [5] (see also [4] for a mathematical analysis) the authors of the present paper extended the above mentioned neighborhood filters to a wider class which they called *non-local means (NL-means).* This algorithm defines the neighborhood $J(i)$ of i by the condition: $j \in J(i)$ if a whole spatial neighborhood and not just a pixel is close to i. The spatial constraint is instead relaxed. It is no more asked that $J(i)$ be contained in a spatial neighborhood of i.

NL-means filters can be given two origins beyond classical neighborhood filters. The same Markovian pixel similarity model was used in the seminal paper [9] for texture synthesis from a texture sample. In that case the neighborhood $J(i)$ is not used for denoising. The aim is to estimate from the texture sample the law of i knowing its neighborhood. This law is used to synthesize similar texture images by an iterative algorithm. Last but not least most state of the art movie denoising methods are neighborhood filters and some of them, in some sense, NL-means filters.

Indeed, motion compensated denoising methods start with the search for a temporal neighborhood $J(i)$, the trajectory, followed by an averaging process. By the Lambertian assumption $u(j)$ must be close to $u(i)$ and the trajectory of i is therefore computed as a neighborhood of i in the sense of neighborhood filters. The comparison of grey levels is no sufficient criterion, a difficulty usually called the *aperture problem.* Thus several motion compensated filters involve *block matching.* They construct $J(i)$ by comparing a whole block around j to a whole block around i. In all of these algorithms the neighborhood $J(i)$ picks a single pixel per frame. One of our objectives is to prove that this restriction is actually counterproductive. In fact the performance of movie denoising filters improves significantly by forgetting about trajectories and using all similar pixels in space-time, no matter how many are picked per frame. This amounts to apply directly a NL-means filter to the movie thought of as a union of images. The time ordering of this union is somewhat irrelevant. The aperture problem results in the existence of more samples for each pixel and therefore increases the denoising performance by nonlocal means.

No new denoising method should be presented without a systematic comparison to the huge variety of denoising methods. On the other hand a comparison between methods based on very different principles cannot be performed without formal comparison criteria. Visual comparison of artificially noisy imges with their denoised version is subjective. Tables comparing distances of the denoised image to the original are useful. They have two drawbacks, though. The added noise is usually not realistic, generally a white uniform noise with too large variance. Such comparison methods depend strongly on the choice of the image and do not permit to address the main issues: the loss of image structure in noise and the creation of artifacts.

Thus we shall propose and apply three principles aiming at more objective benchmarks. The first principle asks that noise and only noise be removed from an image. It has to be tested directly on an image with no artificial noise added. The comparison of methods is performed on the difference between the image and its denoised version which will be called *method noise.* It is much easier to evaluate whether a method noise contains some structure removed from the image or not. The outcome of such experiments is clear cut on a wide class of denoising filters of all origins including all mentioned neighborhood filters. The second principle, *noise to noise*, requires that a denoising algorithm transforms a white noise into white noise. This paradoxical requirement seems to be the best way to characterize artifact-free algorithms. It is affordable to mathematical analysis. Mathematical and experimental arguments will show that bilateral filters and NL-means are the only ones satisfying the noise to noise principle. The third principle, *statistical optimality* is restricted to neighborhood filters. It questions whether a given neighborhood filter is able or not to retrieve faithfully the neighborhood $J(i)$ of any pixel i. NL-means will be shown to best match this requirement.

We shall proceed as follows. Section 2 presents a realistic CCD noise model and the basic hypothesis justifying neighborhood filters. Neighborhood filters in a general sense including NL-means and motion compensated filters are defined in Section 3. Section 4 proposes the three principles to evaluate the performance of any denoising method. All three principles are designed to serve comparative experiments. A special attention will be paid to motion compensated filters. The implementation of NL-means is detailed in Section 5. Section 6 proposes two natural improvements of this algorithm avoiding two slight artifacts due to the fixed position of comparison windows around each pixel (adhesion effect) and to an over smoothing in flat regions.

2 Noise model

Most digital images and movies are obtained by a CCD device. Following [7, 31], CCD's show three kinds of noise. The first one is the *shot noise* proportional to the square root of the number of incoming photons in the captors during the exposure time, namely

$$n_0 = \sqrt{\frac{\Phi}{h\nu} t \cdot A \cdot \eta},$$

where Φ is the light power (W/m^2), $h\nu$ the photon energy (Ws), t the exposure time in seconds (s), A the pixel area (m^2) and η the quantum efficiency. The other constants being fixed we can simply retain $n_0 = c\sqrt{\Phi}$ where Φ is the "true image" and C a constant (see Figure 1.)

Second, a *dark* or *obscurity noise* n_1 is due to spurious photons produced by the captor itself. We can assume the dark noise to be white, additive and with zero mean. The zero mean property is due to the substraction of a *dark frame* from the raw image. The dark frame is obtained by averaging the obscurity noise over a long period of time.

Third, the *read out* noise n_2 is another electronic additive and signal independent noise. This noise can be assumed to have zero mean by the substraction from the raw image of a *bias frame*.

Digital images eventually undergo a "gamma" correction, i.e. a nonlinear increasing contrast change g: "Gamma correction is the name of an internal adjustment made in the rendering of images through photography, television, and computer imaging. The adjustment causes the spacing of steps of shade between the brightest and dimmest part of an image to appear "appropriate" [31]." Summarizing,

$$u(i) = g\left(\Phi(i) + c\sqrt{\Phi(i)} n_0(i) + n_1(i) + n_2(i)\right),$$

where $u(i)$ is the observed intensity at a pixel i, $\Phi(i)$ the "true physical" light intensity average power sent by the scene to pixel i, c a constant, $n_0(i)$, $n_1(i)$ and $n_2(i)$ three independent and signal independent white noises. In practice $g(s) = s^\alpha$ with $0 < \alpha < 1$. When $\Phi(i)$ is large the shot noise $\sqrt{\Phi(i)}$ dominates n_1 and n_2 and is dominated by the signal $\Phi(i)$. Thus we can expand $u(i)$ as

$$u(i) \simeq g(\Phi(i)) + g'(\Phi(i))\left(c\sqrt{\Phi(i)} n_0(i) + n_1(i) + n_2(i)\right) =: g(\Phi(i)) + n(i). \quad (1)$$

Figure 1: Simulated shot noise. Left: original image u. Right: noise image $\sqrt{u}n$ where n is the realization of a zero mean white noise with standard deviation $\sigma = 1$. The noise in bright parts is larger than in dark parts, an effect which is corrected and sometimes reversed by gamma-correction.

If instead $\Phi(i)$ is small with respect to $n_1(i) + n_2(i)$,

$$n(i) \simeq u(i) \simeq g(n_1(i) + n_2(i)). \quad (2)$$

Let us mention a case of particular interest. If $g(s) \simeq s^{\frac{1}{2}}$, the noise $n(i)$ reads

$$n(i) \simeq n_0(i) \quad \text{in the bright parts of the image} \quad (3)$$

$$= \sqrt{n_1(i) + n_2(i)} \quad \text{in the dark parts of the image.} \quad (4)$$

In all cases the noise is signal dependent but independent at different pixels. Figure 2 displays a simulated shot noise associated to the Lena image. This noise is signal dependent and much stronger in bright regions than in dark regions. Figure 11 displays the real noise observed in the same image. The comparison shows that the gamma correction succeeded in restoring a uniform noise in the image. Indeed, in this image the amplitude of the noise is fairly independent of the brightness. This fact is compatible with the relations (3), provided the variances of n_0 and $n_1 + n_2$ have the same magnitude.

In the following we aim at recovering $g(\Phi(i))$, namely the true image up to the unknown gamma correction. Equations (1), (2) and the white noise and independence assumptions on n_0, n_1 and n_2 legitimate the following hypothesis:

Hypothesis 1 *In a digital image, the noise model at each pixel i only depends on the original pixel value $\Phi(i)$ and is additive. Let $J(i)$ be the set of pixels with the same original value as i. Then $n(j)$, $j \in J(i)$ are independent and identically distributed.*

Hypothesis 1 cannot be used directly because $\Phi(i)$ is unknown. The challenge is to find out $J(i)$ for every i. The simplest idea to do so is to assume that all pixels with the same observed value $u(i)$ have the same noise model. A more sophisticated use of Hypothesis 1 is the following : *for a given pixel in an image, detect all pixels which have the same underlying model.* By Hypothesis 1 each j in $J(i)$ obeys a model $u(j) = v(i) + n(j)$ where $n(j)$ are i.i.d. It is then licit to perform a denoising of $u(i)$ by replacing it by $NFu(i) =: \frac{1}{|J(i)|}\sum_{j\in J(i)} u(j)$. By the variance formula for independent variables one then obtains $NFu(i) = v(i) + \tilde{n}(i)$ where

$$\mathrm{Var}(\tilde{n}(i)) = \frac{1}{(\mathrm{Card}(J(i)))^{\frac{1}{2}}}\mathrm{Var(n(i))}. \quad (5)$$

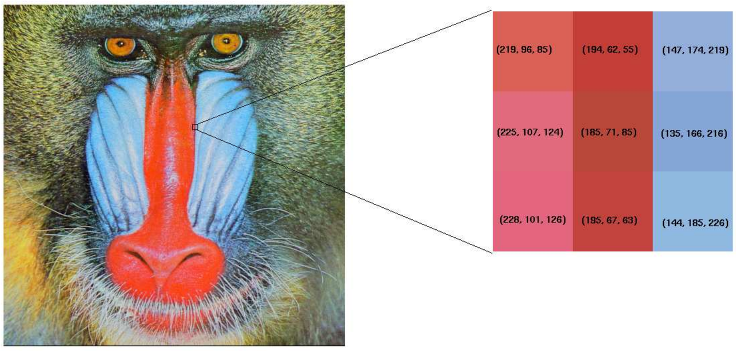

Figure 2: The nine pixels in the baboon image on the left have been enlarged. They present a high red-blue contrast. In the red pixels, the first (red) component is stronger. In the blue pixels, the third component, blue, dominates. Neighborhood filters select pixels with the same color for averaging. In this case the neighborhood of the central pixel should contain the six red pixels or, still better, the pixels of the central column.

By (5) if nine pixels with the same color plus some decorrelated noise are averaged the noise is divided by 3. Algorithms proceeding in this way will be called neighborhood filters. We shall now examine several classical or new instances.

3 General neighborhood filters

3.1 Local neighborhood filters

The more primitive neighborhood filters replace the color of a pixel with an average of the nearby pixels colors. Thus $J(i)$ is a spatial neighborhood. The filtered value can be written as

$$\mathcal{M}_\rho u(\mathbf{x}) = \frac{1}{\pi\rho^2}\int_{\mathbb{R}^2} e^{-\frac{|\mathbf{x}-\mathbf{y}|^2}{\rho^2}} u(\mathbf{y})\, d\mathbf{y}.$$

The parameter ρ is roughly the size of the spatial neighborhood involved in the filtering. Now, the closest pixels to i have not necessarily the same color as i. For instance the red pixel placed in the middle of Figure 2 has five red neighbors and three blue ones. If its color is replaced by the average of the colors of its neighbors, it turns blue. The same process would likewise redden the blue pixels of this figure. Thus, the border between red and blue would be blurred.

In order to denoise the central red pixel, it is better to average the color of this pixel with the nearby red pixels and only them, excluding the blue ones. This is exactly the technique of the sigma-filter. This famous algorithm is generally attributed to J.S. Lee [14] in 1983 but can be traced back to L. Yaroslavsky and the Soviet Union image processing school [32]. The idea is to average neighboring pixels which also have a similar color value. The filtered value by this strategy can be written

$$NF_{h,\rho}\, u(\mathbf{x}) = \frac{1}{C(\mathbf{x})}\int_{B_\rho(\mathbf{x})} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} u(\mathbf{y}) d\mathbf{y}, \tag{6}$$

where $u(\mathbf{x})$ is the color at $\mathbf{x}$ and $NF_{h,\rho}\, u(\mathbf{x})$ its denoised version. Only pixels inside $B_\rho(\mathbf{x})$ are averaged, h controls the color similarity and $C(\mathbf{x})$ is the normalization factor. SUSAN [25] and the bilateral filter [27] make this process more symmetric by involving a "bilateral" gaussian depending on both space and grey level. This leads to

$$SNF_{h,\rho}\, u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int e^{-\frac{|\mathbf{x}-\mathbf{y}|^2}{\rho^2}} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} u(\mathbf{y}) d\mathbf{y}$$

There is another way to avoid the blurring effect of the spatial filtering M_ρ by a statistical correction which we are going to use in the sequel. When the gaussian mean is performed on an edge, the variance of the performed mean can become larger than the variance of the noise. This is a clue that the average is not licit. A statistically optimal correction was proposed by Lee again [13],

$$L\mathcal{M}_\rho u(\mathbf{x}) = \mathcal{M}_\rho u(\mathbf{x}) + \frac{\sigma_{\mathbf{x}}^2}{\sigma_{\mathbf{x}}^2 + \sigma^2}(u(\mathbf{x}) - \mathcal{M}_\rho u(\mathbf{x})),$$

where

$$\sigma_{\mathbf{x}}^2 = \max(0, \frac{1}{\pi\rho^2} \int_{\mathbb{R}^2} e^{-\frac{|\mathbf{x}-\mathbf{y}|^2}{\rho^2}} (u(\mathbf{y}) - \mathcal{M}_\rho u(\mathbf{x}))^2 \, d\mathbf{y} - \sigma^2)$$

and σ is the noise standard deviation. The original noisy values are less altered when the variance of the performed mean dominates the variance of the noise. This happens near the edges or in textured regions. In consequence, the noise is mainly reduced in flat zones as displayed in Figure 4.

The bilateral filters perform a better denoising than Lee's correction. They maintain sharp boundaries, since they average pixels belonging to the same region as the reference pixel. Bilateral filters fail when the standard deviation of the noise exceeds the edge contrast.

Let us finally mention that the mean operation can be replaced by nonlinear operator like the median filter. The median filter [28] chooses the median value, that is, the value which has exactly the same number of grey level values above and below in a fixed neighborhood. The median filter preserves the main boundaries, but it tends to remove the details. This filter is optimal for the removal of impulse noise on images and does'nt blur edges. It is equivalent to an average of the pixels in a direction orthogonal to the gradient, that is to an *anisotropic diffusion* or *mean curvature motion* [16]. In the following, we shall also show experiments based on *anisotropic diffusion*.

Figure 4 compares the performance of the various considered local neighborhood filters on a noisy image. Even if each method provides a reasonable solution (all except the gaussian filtering maintain sharp edges), none of them is fully acceptable. The anisotropic filter removes small details and fine structures. These features are nearly untouched by Lee's statistical filter and therefore completely noisy. The bilateral filters create irregularities on the edges and leave some residual noise on flat zones.

3.2 Non local averaging

The most similar pixels to a given pixel have no reason to be close to it. Think of periodic patterns, or of the elongated edges which appear in most images. In 1999 Efros and Leung [9] used non local self-similarities as the ones illustrated in Figure 3 to synthesize textures and to fill in holes in images. Their algorithm scans a vast portion of the image in search of all the pixels that resemble the pixel in restoration. The resemblance is evaluated by comparing

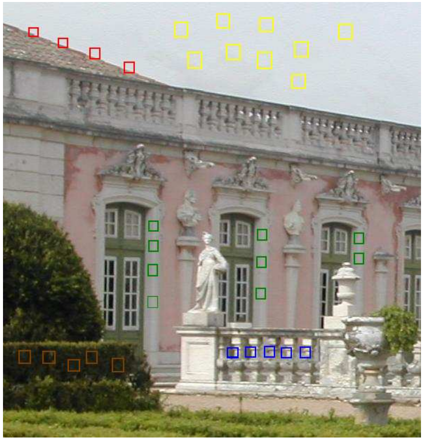

Figure 3: Most image details occur repeatedly. Each color indicates a group of squares in the image which are almost indistinguishable. Image self-similarity can used to eliminate noise. It suffices to average the squares which resemble each other.

a whole window around each pixel, not just the color of the pixel itself. Applying this idea to neighborhood filters leads to a generalized neighborhood filter which we called non-local means (or NL-means) [4, 5]. NL-means has a formula quite similar to the sigma-filter,

$$NLu(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} e^{-\frac{(G_\rho * |u(\mathbf{x}+.)-u(\mathbf{y}+.)|^2)(0)}{h^2}} \; u(\mathbf{y}) \; d\mathbf{y}, \tag{7}$$

where G_ρ is the Gauss kernel with standard deviation ρ, $C(\mathbf{x})$ is the normalizing factor, h acts as a filtering parameter and

$$(G_\rho * |u(\mathbf{x}+.) - u(\mathbf{y}+.)|^2)(0) = \int_{\mathbb{R}^2} G_\rho(\mathbf{t})|u(\mathbf{x}+\mathbf{t}) - u(\mathbf{y}+\mathbf{t})|^2 d\mathbf{t}.$$

The formula (7) means that $u(\mathbf{x})$ is replaced by a weighted average of $u(\mathbf{y})$. The weights are significant only if a gaussian window around $\mathbf{y}$ looks like the corresponding gaussian window around $\mathbf{x}$. Thus the non-local means algorithm uses image self-similarity to reduce the noise as illustrated in Figure 4.

3.3 Movie denoising

Nearly all state of the art movie filters are motion compensated. The underlying idea is the existence of a "ground true" physical motion, which motion estimation algorithms should be able to estimate. Legitimate information should exist only along these physical trajectories. The *motion compensated filters* estimate explicitly the motion of the sequence by a motion estimation algorithm. The motion compensated movie yields a new stationary data on which a static filter can be applied.

One of the major difficulties in motion estimation is the ambiguity of trajectories, the so called *aperture problem.* This problem is illustrated in Figure 6. At most pixels, there are several options for the displacement vector. All of these options have a similar grey level value and a similar block around them. Now, motion estimators have to select one by some additional

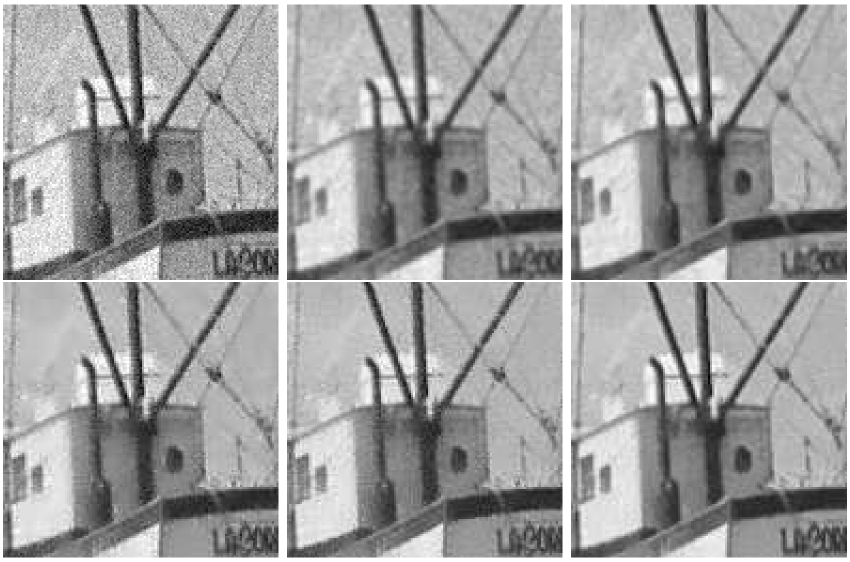

Figure 4: Comparison of neighborhood filters. From top to bottom and left to right: noisy image ($\sigma = 15$), gaussian filtering, anisotropic filtering, Lee's statistical filter, sigma or bilateral filter and the NL-means algorithm. All methods except the gaussian filtering maintain sharp edges. However, the anisotropic filtering removes small details and fine structures. These features are nearly untouched by Lee's statistical filter and therefore completely noisy. The comparison of noisy grey level values by the sigma or bilateral filter is not robust and irregularities are created on the edges. The NL-means better cleans the edges without losing too many fine structures and details.

criterion. The most classical approaches to motion estimation are the *optical flow constraint (OFC)* based methods and *the block matching* algorithms. OFC methods assume that the grey level value of the objects during their trajectory is nearly constant (Lambertian assumption). In order to obtain a unique flow they impose the flow field to vary smoothly in space. There has been a constant progress in this estimation Horn-Schunck [10], Nagel and Weickert [19, 29], Weickert and Schnörr [30]. The second class of motion estimation computes the displacement at each pixel by comparing the grey level values in a whole block around it. The similarity is measured by a l^1 or a l^2 distance. As Figure 6 shows, there can be many blocks with similar configurations in the reference frame.

Once the motion compensation has been performed one can classify the movie denoising methods by the kind of 3D neighborhood filter they apply to the compensated movie. We refer to [3] for a comprehensive review. Samy [23] and Sezan et al. [24] proposed the LMMSE filter which is a *motion compensated Lee's filter*. Ozkan et al [20] proposed the AWA filter, a *motion compensation of the neighborhood filters*. Huang [11] and Martinez [15] implemented *motion compensated median filters*. *Motion compensated Wiener filters* were earlier proposed by Kokaram [12].

Figure 7 illustrates the improvement of motion compensated algorithms compared with their static version. The static filters are obtained by extending the 2D image filter support to the 3D time-space support. We call them static because they do not take into account the dynamic character of image sequences. As this figure illustrates, the details are better preserved and the boundaries less blurred with motion compensation. This explains why most recent papers

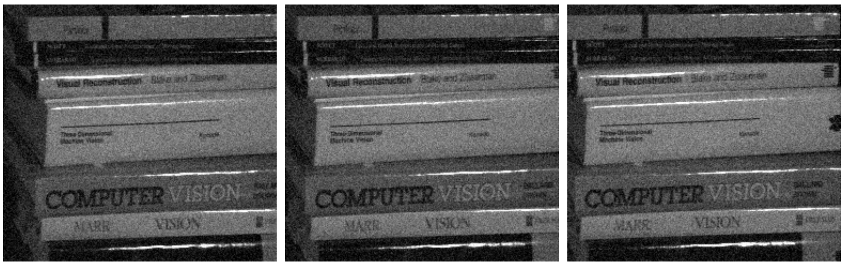

Figure 5: Three consecutive frames of a degraded image sequence. The sparse time sampling in film sequences makes restoration more difficult than in 3D images.

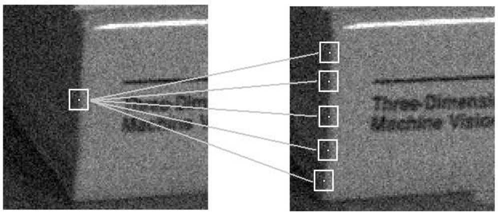

Figure 6: Aperture problem and the ambiguity of trajectories are the most difficult problem in motion estimation: There can be many good matches. The motion estimation algorithms must pick one.

propose motion compensated algorithms.

Non local means for movies

The above description of movie denoising algorithms and its juxtaposition to the NL-means principle shows how the main problem, motion estimation, can be circumvented. In denoising, the more samples we have the happier we are. The *aperture problem* is just a name for the fact that there are many blocks in the next frame similar to a given one in the current frame. Thus, singling out one of them in the next frame to perform the motion compensation is an unnecessary and probably harmful step. A much simpler strategy which takes advantage of the aperture problem is to denoise a movie pixel by involving indiscriminately spatial and temporal similarities: let the best win! In that way, NL-means applied to a movie reads:

$$NLu(\mathbf{x},t) = \frac{1}{C(\mathbf{x},t)} \int_{\Omega} \int_{\mathbb{R}} e^{-\frac{(G_\rho * |u(\mathbf{x}+.,t)-u(\mathbf{y}+.,s)|^2)(0)}{h^2}} \; u(\mathbf{y},s) \; d\mathbf{y}ds. \tag{8}$$

Notice that the Gauss kernel G_ρ is 2D but the integral involves time and space. In practical terms NL-means takes a spatial block around pixel $\mathbf{x}$ in frame t and looks for all similar blocks in all frames *including the current one.* Then a weighted average of all the similar pixels $(\mathbf{y}, s)$ is performed. Figure 7 compares NL-means with motion compensated neighborhood filters.

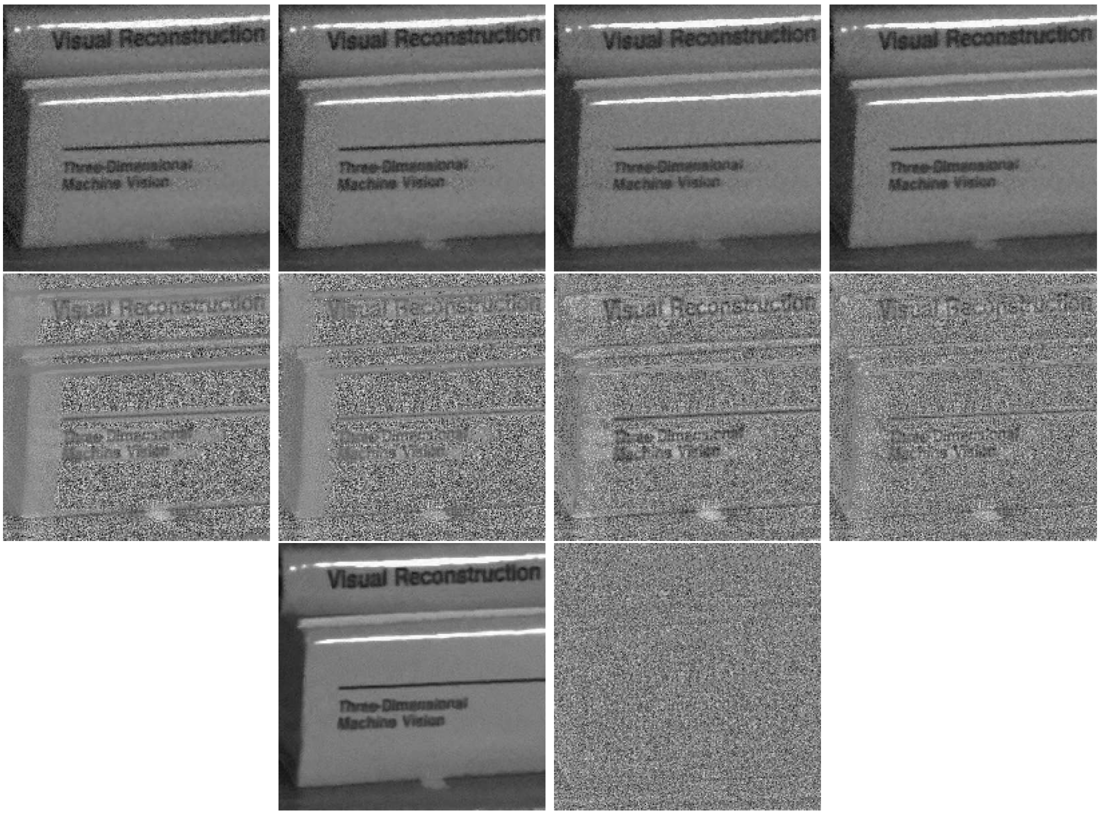

Figure 7: Comparison of static filters, motion compensated filters and NL-means applied to the sequence of Figure 5. Only a piece of the central frame is displayed. From top to bottom and left to right: Lee's correction, Lee's correction with block matching (BM)-(LMMSE), sigma filter and sigma filter with block-matching (AWA). Middle: the noise removed by each method (difference between the noisy and filtered frame). Motion compensation improves the static algorithms by better preserving the details and creating less blur. The LMMSE removed noise is nearly zero on the strong boundaries. Now, these boundaries are kept noisy. We can read the titles of the books in the noise removed by AWA. Therefore, that much information has been removed from the original. Finally, the NL-means algorithm (bottom row) does not have any noticeable structure in its removed noise. As a consequence, the filtered sequence has kept more details and is less blurred.

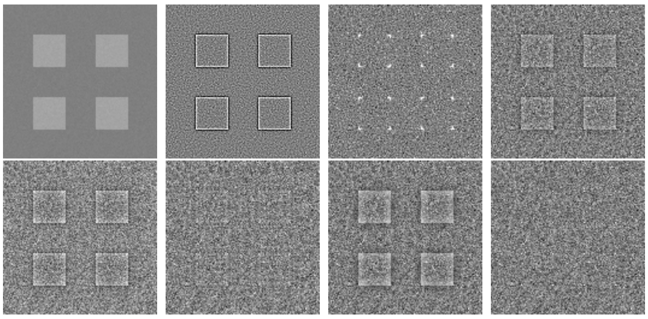

Figure 8: *Method noise* obtained by various denoising methods applied to a slightly noisy image ($\sigma = 2.5$). From top to bottom and left to right: noisy image, gaussian mean, mean curvature motion, total variation minimization, translation invariant soft and hard thresholding, bilateral filter and NL-means. The various method parameters have been adjusted so that the method noise has a per pixel variance equal to σ^2.

4 Three principles for denoising algorithms evaluation

4.1 Method noise

A difference between the original image and its filtered version shows the "noise" removed by the algorithm. We call this residue *method noise*. In principle the method noise should look like a noise. Otherwise, the method noise can be filtered again and its deterministic part turned back to the image. Recent denoising methods adopted this recursive strategy to recover image information lost in method noise [21, 26]. When the standard deviation of the noise is higher than the feature contrast a visual exploration of the method noise is not reliable. Image features can be masked in method noise. Thus the evaluation of a denoising method should not rely on experiments where a white noise with standard deviation larger than 5 has been added to the original. The best way is actually not to add noise at all.

Definition 1 (method noise) *Let u be a (not necessarily noisy) image and D_h a denoising operator depending on h. The* method noise of *u is the image difference*

$$n(D_h, u) = u - D_h(u). \tag{9}$$

Principle 1 *For every denoising algorithm, the method noise must be zero if the image contains no noise and should be in general an image of independent zero-mean random variables.*

Figure 8 and 9 display the method noise of various denoising methods on a simple geometrical image. The algorithms are applied to a slightly noisy version of the image ($\sigma = 2.5$). Method parameters are fixed so that the method noise has exactly σ^2 variance per pixel. The same parameters have been used in the second experiment on a real image. These method noise images should look like a white noise in Figure 8 and like a constant image in Figure 9. The gaussian filter method noise highlights all the boundaries and corners of the image. Averages are performed on a radial neighborhood and therefore do not adapt to the geometrical configuration of the image. The anisotropic (median, mean curvature equation) filter averages pixels in the direction of contours and therefore does not modify straight edges. Straight boundaries are

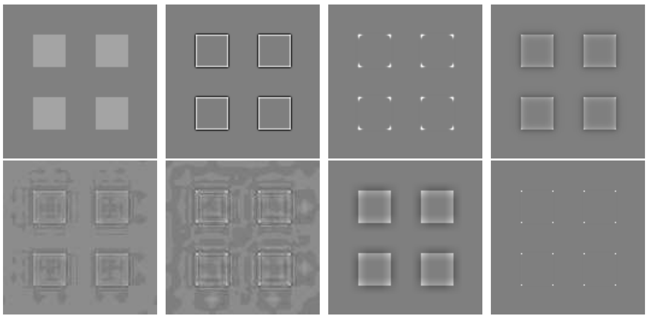

Figure 9: Method noise experiment. Application of various denoising methods to the non noisy image of Figure 8 with the same filtering parameters. From top to bottom and left to right: noisy image, gaussian mean, mean curvature motion, total variation minimization, translation invariant soft and hard thresholding, bilateral filter and the NL-means.

not noticeable in the method noise. However, the corners are not well preserved since they move at the speed of their high curvature. The total variation minimization [22] is praised for maintaining sharp boundaries. However, most structures are modified and even straight edges are not well preserved. This fact has received a mathematical proof in [18]. The wavelet thresholding [8] method noise is concentrated on the edges and corners. These structures lead to coefficients of large enough value but lower than the threshold and which are erroneously canceled. The method noise of the soft thresholding is not only based on the small coefficients but also on an attenuation of the large ones, leading to a general alteration of the original image.

The bilateral filter preserves the flat zones, but the edges with a low contrast have been modified. The NL-means method noise is the one which looks the more like a white noise. When applying the algorithms to the non noisy image, the removed features are more noticeable. The corners of the squares can now be seen in the NL-means method noise. These are the only features with a reduced amount of similar samples, since for every corner there are only three similar corners in the image. The experiment of Figure 9 has been designed to illustrate the usage of the method noise on images without noise at all, usually synthetic images. Such experiments characterize immediately the image features sensitive to a given denoising method.

Finally, Figure 10 displays the method noise on a real image. The algorithms were applied to the original Lena image scanned in 1973, in grey level. It contains a little amount of noise. None of the methods can be claimed to deliver a method noise looking like a noise. The hard thresholding and NL-means give the least structured method noise. The method noise of Lena in color is displayed in Figure 11. Color images are obviously easier to denoise by neighborhood filters.

4.2 Noise to noise principle

The noise to noise principle asks that a white noise be transformed into a white noise. This requirement may look paradoxical since noise is what we wish to get rid of. Now, it is impossible to totally remove noise. The question is how the remnants of noise look like. The transformation of a white noise into any correlated signal creates structure and artifacts. Only white noise is perceptually devoid of structure, as was pointed out by Attneave [1].

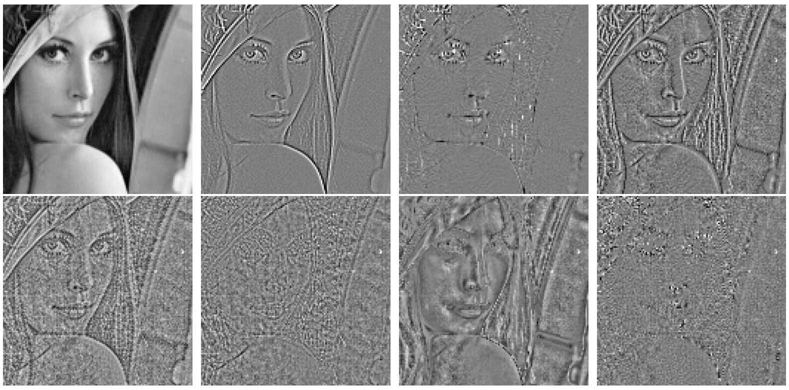

Figure 10: Method noise experiment on Lena (gray levels only). From top to bottom and left to right: original image, gaussian mean, mean curvature motion, total variation minimization, translation invariant soft and hard thresholding, bilateral filter and NL-means.

Figure 11: Lena method noise (in color) by the sigma filter and the NL-means. The results of neighborhood filters improve dramatically on color images because similar pixels are much better identifiable with three components.

Principle 2 *A denoising algorithm must transform a white noise image into a white noise image (with lower variance).*

There are two ways to check this principle for a given denoising method. One of them is to find a mathematical proof that the pixels remain independent (or at least decorrelated) and identically distributed random variables. The experimental device simply is an observation the effect of denoising on the simulated realization of a white noise. Since the Fourier transform of a white gaussian noise is a white gaussian noise, the visualization of the Fourier transform also is an adequate tool. Let us review how well classical algorithms match the *noise to noise* principle. Figures 12 and 13 respectively display the filtered noises and their Fourier transforms.

Gaussian convolution. The convolution with a Gauss kernel G_h is equivalent to the product in the Fourier domain with a Gauss kernel of inverse standard deviation $G_{1/h}$. Therefore, convolving the noise with a kernel reinforces the low frequencies and cancels the high ones. Thus, the filtered noise will no more be a white noise and actually shows big grains due to its prominent low frequencies.

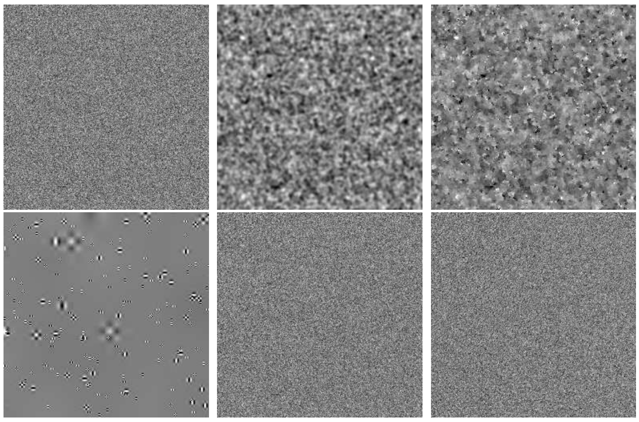

Figure 12: Noise to noise principle: Application of the denoising algorithms to a noise sample. From left to right and top to bottom: noise sample ($\sigma = 15$), filtered noise by the gaussian filtering, total variation minimization, hard wavelet thresholding, bilateral filter and the NL-means algorithm. The parameters have been tuned in order to have a filtered noise of standard deviation 2.5.

Total variation minimization. The Fourier transforms of the total variation minimization and the gaussian filtering are quite similar even if the total variation preserves some high frequency components.

Wavelet Thresholding. Noise filtered by a wavelet thresholding is no more a white noise. The few coefficients with a magnitude larger than the threshold are spread all over the image. The pixels which do not belong to the support of one of these coefficients are set to zero. The visual result is a constant image with superposed wavelets as displayed in Figure 12. It is easy to prove that the denoised noise is spatially highly correlated.

Bilateral filter. For simplicity consider the case where the grey level neighborhood is an interval. Given a noise realization, the filtered value by the bilateral filter at a pixel i only depends on its value $n(i)$ and the parameters h and ρ. The bilateral filter averages noise values at a distance from $n(i)$ less or equal than h. Thus as the size ρ of the neighborhood increases by the law of large numbers the filtered value tends to the expectation of the Gauss distribution restricted to the interval $(n(i) - h, n(i) + h)$. The filtered value is therefore a deterministic function of $n(i)$ and h. Independent random variables are mapped by a deterministic function on independent variables. Thus the *noise to noise* requirement is asymptotically satisfied by the bilateral filter. A visual check in Figure 13 fully confirms this theoretical result.

NL-means algorithm Figure 13 indicates that NL-means satisfies the *noise to noise* principle in the same extent as a classical bilateral filter. However, a mathematical statement and proof of this property are intricate and we shall skip them.

4.3 Statistical optimality

The statistical optimality means the ability of a generalized neighborhood filter to find the right set of pixels $J(i)$ for performing the average yielding the new estimate for $u(i)$. This principle for the comparison of denoising algorithms applies for all algorithms performing an

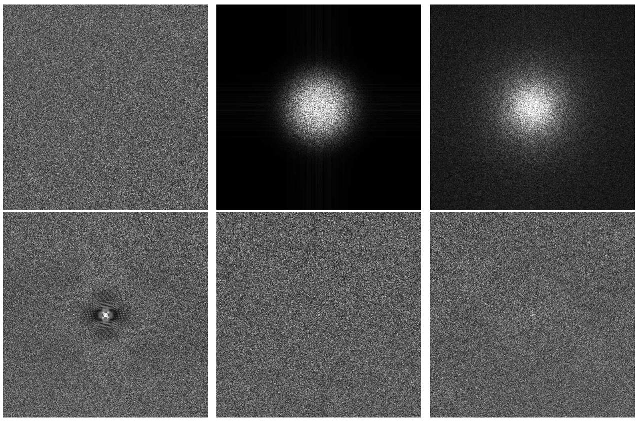

Figure 13: Noise to Noise principle: Fourier transforms of the filtered noises displayed in Figure 12. The Fourier transform of a gaussian white noise is a gaussian white noise.

average over a set of selected pixels, namely, all neighborhood filters including NL-means and all motion compensated movie denoising algorithms (AWA, LMMSE,...).

Principle 3 *A generalized neighborhood filter is optimal if it finds for each pixel i all and only the pixels j having the same model as i.*

Returning to the signal dependent noise model given by Hypothesis 1, we notice that the ideal denoising algorithm from that point of view would give for $J(i)$ the set of all pixels j with the same original, noiseless value $\Phi(j) = \Phi(i)$ as i. This aim is not attainable and can be replaced by a search for pixels j which are *likely* to have the same value as i. In movies, by the Lambertian assumption, it can be assumed that a non occluded pixel keeps the same grey level in several frames. Thus in motion compensated movie filters $J(i)$ is the trajectory of i. In the case of NL-means, it is assumed that pixels having similar neighborhoods for some distance also have close colors. Principle 3 cannot be checked in theory but can be in practice explored by displaying the probability distribution of $w(j)$, $j \in J(i)$ for various algorithms and images. In that way it can be checked whether $J(i)$ corresponds to the pixels j perceptually equivalent or similar to i. This visualization is actually quite informative as illustrated in Figures 14 and 16.

As displayed in Figure 14, the orientation computed by the anisotropic filter is not exactly the one should expect. This fact is due to the noise interference on the gradient computation. The noise also degrades the probability distribution of the bilateral filter. The window comparison of NL-means is more robust to noise and yields a more adapted weight configuration.

Statistical optimality and the aperture problem.

Let us now address the problem of statistical optimality for movies. We shall sustain the position that, in fact, motion estimation is not only unnecessary, but probably counterproductive. The aperture problem, viewed as a general phenomenon in movies, can be positively interpreted in the following way: There are many pixels in the next or previous frames which can match the current pixel. Thus, it seems sound to use not just one trajectory, but rather *all similar pixels* to the current pixel across time and space.

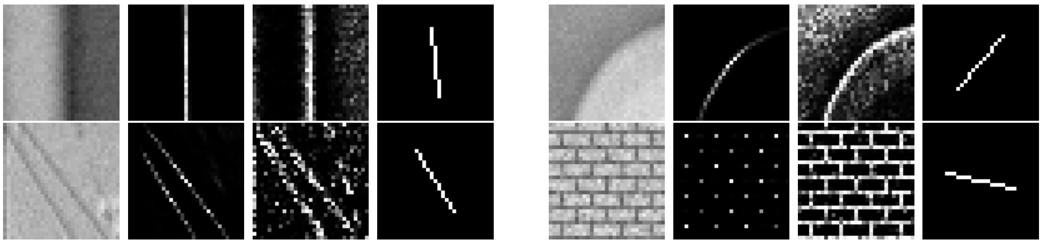

Figure 14: Weight distribution of NL-means, the bilateral filter and the anisotropic filter used to estimate the central pixel in four detail images.

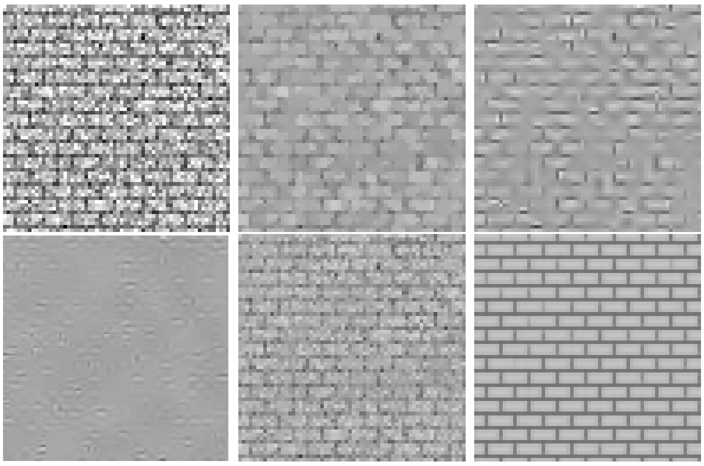

Figure 15: Denoising experiment on a noisy periodic texture. From top to bottom and left to right: noisy image (standard deviation 35), total variation minimization, translation invariant hard thresholding (threshold 3σ), translation invariant hard thresholding (optimal threshold $\sqrt{2\log N}\sigma$), bilateral filter and NL-means.

Motion estimation algorithms try to solve the aperture problem. The block matching algorithm chooses the pixel with the more similar configuration, thus loosing many other interesting possibilities, as displayed in Figure 6. Algorithms based on the optical flow constraint must impose a regularity condition of the flow field in order to choose a single trajectory. Thus, the motion estimation algorithms are forced to choose a candidate among all possible equally good choices. However, when dealing with sequence restoration, the redundancy is not a problem but an advantage. Figure 6 shows all possible and equally good candidates for the averaging. Why not take them all.

Figure 16 displays the probability distribution of the weights computed by NL-means for three different cases. The algorithm favors pixels with a similar local configuration even if they are far away from the reference pixel. As the similar configurations move, so do the weights. Thus, the algorithm is able to follow the similar configurations when they move without any explicit motion computation. No need to solve any aperture problem; it turns out to be advantageous.

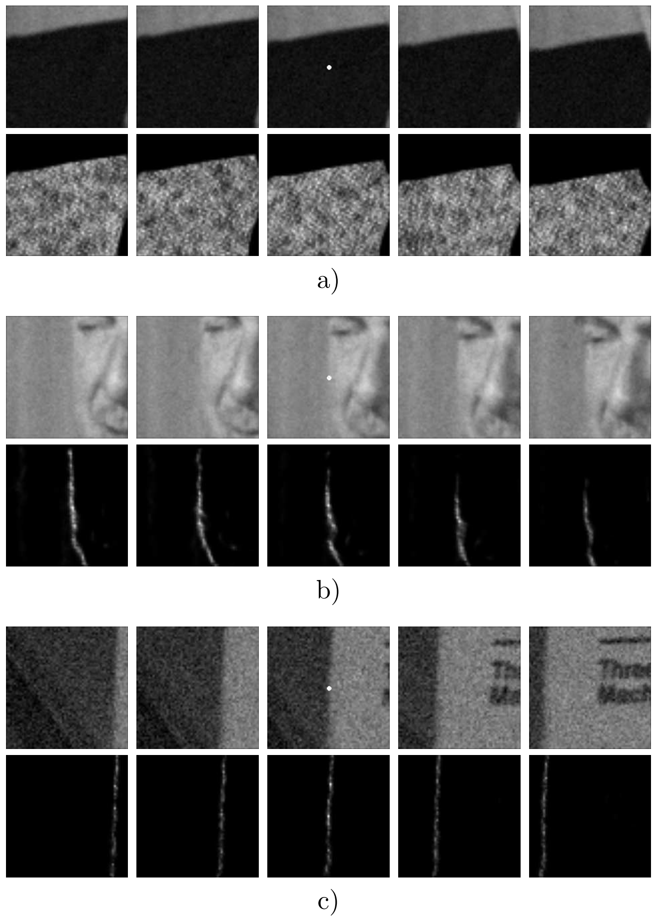

Figure 16: Weight distribution of NL-means applied to a movie. In a), b) and c) the first row shows a five frames image sequence. In the second row, the weight distribution used to estimate the central pixel (in white) of the middle frame is shown. The weights are equally distributed over the successive frames, including the current one. They actually involve all the candidates for the motion estimation instead of picking just one per frame. The aperture problem can be taken advantage of for a better denoising performance by involving more pixels in the average.

5 Implementation

The basic idea of bilateral or sigma filters is to average pixels at a grey level distance less or equal than a certain threshold. If an additive white noise model is assumed, then this parameter is usually set to 2.5σ or 3σ where σ is the noise standard deviation. Thus these filters fail at image details whose local variance is below 3σ. (Of course the same remark applies to all thresholding methods.) NL-means not only compares the grey level values but a whole fixed size window $\mathcal{N}_i \setminus \{i\}$ around the reference pixel i and the corresponding window $\mathcal{N}_j$ around every other pixel j. The comparison of both windows is made by an Euclidean norm of their difference. Indeed, if the noise samples are locally i.i.d. with zero mean and σ variance,

$$E||u(\mathcal{N}_i) - u(\mathcal{N}_j)||^2 = ||u_0(\mathcal{N}_i) - u_0(\mathcal{N}_j)||^2 + 2\sigma^2,$$

where u_0 denotes the original (unknown) image, u the noisy one obtained by the addition of a white noise. Thus using a threshold function and setting this hard threshold to $2\sigma^2$ leads to take an average of pixels which originally had an almost identical window around them. A first version of NL-means can be written as

$$NL(i) = \frac{1}{|J(i)|} \sum_{j \in J(i)} u(j),$$

where $J(i) = \{j \mid d(\mathcal{N}_i, \mathcal{N}_j) < 2\sigma^2\}$ and d denotes the Euclidean distance of windows normalized by their area.

It must be explained why the central pixel is excluded in the window when computing the Euclidean distance of two windows. Otherwise the search for similar windows would be influenced by the value of the central pixel and the average would be closer to this original and noisy value. Thus the denoised value would still contain some of the pixel's noise, a drawback observed in classical bilateral or sigma filters. The next point is : why using a soft gaussian threshold instead of a hard one? We may find pixels for which there is no identical or nearly identical window in the image. In that case, the threshold strategy should leave exactly the noise value at such points. The result would visually be identified as an impulse noise and the *noise to noise* principle would be violated. An exponential function is used instead of the threshold and makes a more adaptive weighting distribution. Thus, given an image $u = \{u(i) \mid i \in I\}$, the filtered value $NLu(i)$, for a pixel i, can be computed as a weighted average of all the pixels in the image,

$$NLu(i) = \sum_{j \in I} w(i,j)u(j),$$

where the family of weights $\{w(i,j)\}_j$ is

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||u(\mathcal{N}_i) - u(\mathcal{N}_j)||^2}{h^2}},$$

$Z(i)$ is the normalizing constant and the parameter h acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

The Euclidean distance between two windows is also weighted by a gaussian-like kernel decaying from the center of the widow to its boundary. Notice that the window sizes need not exceed 5×5, except maybe in very noisy images. Indeed in digital images, closer pixels are more dependent and therefore closer pixels to the reference one should have more importance in the window comparison. In order to involve the current pixel in its own average, the distance between the window centered at the reference pixel and itself is set equal to the minimum of the other distances. Otherwise, the probability distribution should be excessively picked on the pixel itself.

6 Drawbacks and improvements of NL-means

In general bilateral or sigma filters share a shock effect due to their enhancing character. Because of this enhancement character spurious boundaries can be created inside regular zones as displayed in Figure 17. This shock effect has been mathematically studied by computing the subjacent PDE of the filter and avoided by replacing the average by a linear regression approximation [6].

A common effect of algorithms using correlation and window comparison is the adhesion artifact. If a pixel inside a flat zone is near an edge, then the window distances are dominated by it. Then the weight configuration provided by the NL-means at this pixel is concentrated in the same direction of the edge. This effect is visible in Figure 17 as a shadow surrounding the boundary of the hat.

Figure 17 also displays an example where NL-means have excessively filtered a textured zone. This section is devoted to the reduction of this blurring and adhesion effects.

6.1 Sliding window correction of adhesion effect

The adhesion effect is due to the dominance of edges in window comparison. In order to avoid this effect, one must use a window containing the reference pixel but avoiding nearby edges. This is easily done by involving all the windows with fixed size containing the reference pixel instead of just one. Thus for any two pixels i and j and a fixed window size n^2, we dispose of n^2 distances, $d_1(i,j), \ldots, d_{n^2}(i,j)$, each one coming from a different pair of windows. The variant of NL-means with sliding windows is

$$SNL(i) = \frac{1}{|J|} \sum_{k=1}^{n^2} \sum_{j \in J(k)} u(j),$$

where $|J| = |J_1| + \ldots + |J_{n^2}|$ and $J(k) = \{j \in I \mid d_k(i,j) < h\}$. A threshold is used now instead of the exponential function. Since many windows containing the reference point are being compared, there is always a sufficient number of similar ones. Figure 17 shows how the adhesion effect is avoided by this modified NL-means.

6.2 Best average correction of excessively filtered details

In practice, commercial algorithms reduce this excessive filtering effect by averaging the filtered image with the noisy one. This average recovers details but also adds back noise. The same strategy is used by Lee's statistical filter. In that case, the intensity of the correction depends on the variance of the performed mean and noise is only added on edges and details. However, the visual quality of the restored image is downgraded by the added noise. Another correction strategy adapted to neighborhood filters can be defined as follows. Let us first compute the weights of an ideal average filter in the presence of uniform white noise with standard deviation

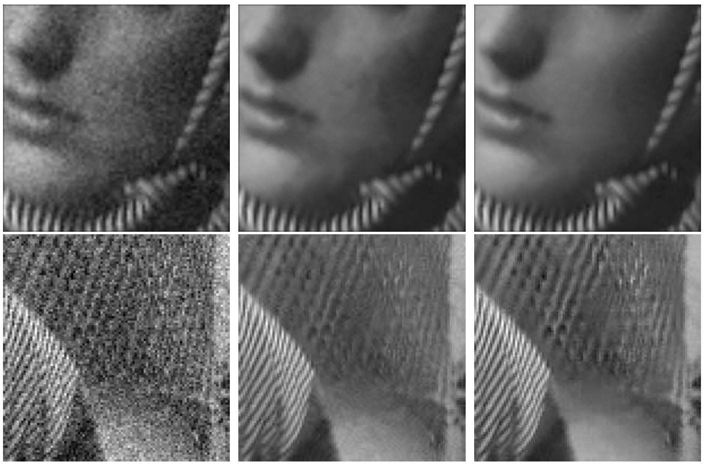

Figure 17: NL-means drawbacks. From left to right: noisy image, the same filtered by NL-means and the correction of the drawback. Top: Correction of the shock effect by taking a linear regression instead of an average (see [6]). Bottom: over-filtering avoided by using the best average correction and sliding windows

σ. The filtered value of a pixel i is a weighted average of a certain spatial neighborhood $B(i)$ which can be the whole image,

$$\hat{u}(i) = \sum_{j \in B(i)} w(i,j)u(j)$$

where the family of weights are such that $\sum w(i,j) = 1$. An easy computation shows the following result.

Theorem 1 *(Optimal linear correction)*
The family of weights minimizing the mean square error between $u_0(i)$ and $\hat{u}(i)$ under an additive white noise model is given by

$$w(i,j) = \frac{1}{C(i,j)}(\sigma^2 + \sum_k d_k^2 - d_j \sum_k d_k), \tag{10}$$

where $C(i,j)$ is the normalization factor and $d_k = u_0(i) - u_0(k)$ is the true grey level difference.

The optimal weights depend on the original grey level distances. If the whole image is used as the spatial neighborhood, the pixels originally having the same grey level value will then have the same filtered one. Thus, the level lines of the original image are maintained by this averaging process. Of course, the ideal weights cannot be computed since the original differences d_k are not available. Now, they can be estimated on a previously filtered image by any denoising method. In Figure 18 the above optimal average uses in turn the d_k given by the filtered image by the bilateral or sigma filter, the translation invariant wavelet thresholding and NL-means. The application of the optimal linear correction recovers details and enhances edges when applied to the three filtered images. However, the application of this correction to the previously filtered image by the sigma filter also recovers and enhances the noise inside the flat zones. We recall that the filtered value by the bilateral filter inside a flat zone is a deterministic function of the noise value and of the threshold (see Section 4.2.) Thus, a residual noise is always present inside the flat zones but not visible because of its low variance. Similarly, when applied to the wavelet thresholding, the correction enhances the wavelet artifacts. For these reasons, the optimal linear correction cannot be applied successfully to both filters. The application of the optimal linear correction to NL-means recovers details and enhances edges without creating any artifact. Figure 18 illustrates the obtained improvement by the optimal linear correction in presence of both edges and texture.

Finally, Figure 19 compares the performance of the different denoising algorithms. The NL-means experiments have been performed with the sliding window modification and optimal linear correction.

References

[1] F. Attneave. Some informational aspects of visual perception. *Psych. Rev.*, 61:183–193, 1954.

[2] J.-F. Aujol, G. Aubert, L. Blanc-Féraud and A. Chambolle, "Image decomposition into a bounded variation component and an oscillating component", Journal of Mathematical Imaging and Vision, vol.22(1), pp. 71-88, 2005.

[3] J. C. Brailean, R. P. Kleihorst, S. Efsratiadis, A. K. Katsaggelos, and R. L. Lagendijk, "Noise reduction filters for dynamic image sequences: a review", Proceedings of the IEEE, vol. 83, pp. 1272-1292, 1995.

[4] A. Buades, B. Coll and J.M. Morel, "A review of image denoising methods, with a new one", Multiscale Modeling and Simulation, vol 4 (2), pp 490-530, 2005.

[5] A. Buades, B. Coll and J.M. Morel, "A non-local algorithm for image denoising", IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2005.

[6] A. Buades, B. Coll and J.M Morel, "The staircasing effect in neighborhood filters and its solution", IEEE Transactions on Image Processing, in Press 2006.

[7] M. Colleen Gino, "Noise, Noise, Noise", $http : //www.astrophys - assist.com/educate/noise/noise.htm$.

[8] D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", Biometrika, 81 pp. 425-455, 1994.

[9] A. Efros and T. Leung, "Texture synthesis by non parametric sampling," Proc. Int. Conf. Computer Vision (ICCV 99), vol. 2, pp. 1033-1038, 1999.

[10] B. Horn and B. Schunck, "Determining optical flow," Artif. Intell., Vol. 17, pp. 185-203, 1981.

[11] T. Huang, "Image Sequence Analysis", Springer-Verlag, 1981.

[12] A. C. Kokaram, "Motion Picture Restoration", PhD thesis, Cambridge University, 1993.

[13] J.S. Lee, "Digital image enhancement and noise filtering by use of local statistics", IEEE Trans. Patt. Anal. Machine Intell., vol. 2, pp. 165-168, 1980.

[14] J.S. Lee, "Digital image smoothing and the sigma filter", Computer Vision, Graphics and Image Processing, vol. 24, pp. 255-269, 1983.

[15] D. M. Martinez, "Model-based motion estimation and its application to restoration and interpolation of motion pictures", PhD thesis, Massachusetts Institute of Technology, 1986.

[16] B. Merriman, J. Bence, and S. Osher, "Diffusion generated motion by mean curvature", In Proc. of the Geometry Center Workshop, 1992.

[17] T. Le and L. Vese, "Image decomposition using total variation and div(BMO)", Multiscale Modeling and Simulation, vol. 4, pp. 390-423, 2005.

[18] Y. Meyer, "Oscillating Patterns in Image Processing and Nonlinear Evolution Equations", AMS University Lecture Series, vol. 22, 2002.

[19] H.H. Nagel, "Constraints for the estimation of displacement vector fields from image sequences," Proc. Eighth Int. Joint Conf. on Artificial Intelligence (IJCAI '83), pp. 945-951, 1983.

[20] M.K. Ozkan, M.I. Sezan, and A.M. Tekalp, "Adaptive motion compensated filtering of noisy image sequences", IEEE Trans. Circuits and Systems for Video Technology, vol. 3, pp. 277-290, 1993.

[21] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, "An iterative regularization method for total variation based image restoration", Multiscale Modelling and Simulation, vol. 4, 460-489, 2005.

[22] L. Rudin, S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms", Physica D, 60, pp. 259-268, 1992.

[23] R. Samy, "An adaptive image sequence filtering scheme based on motion detection", SPIE, Vol. 596, pp. 135-144, 1985.

[24] M.I. Sezan, M.K. Ozkan and S.V. Fogel, "Temporally adaptive filtering of noisy sequences using a robust motion estimation algorithm", Proceedings of the Int. Conf. Acoustics, Speech, Signal Processing 91, pp. 2429-2432, 1991.

[25] S.M. Smith and J.M. Brady, "Susan - a new approach to low level image processing", International Journal of Computer Vision, Volume 23 (1), pp. 45-78, 1997.

[26] E. Tadmor, S. Nezzar, and L. Vese, "A multiscale image representation using hierarchical (BV, L^2) decompositions", Multiscale Modeling and Simulation, vol 2, pp.554–579, 2004.

[27] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Sixth International Conference on Computer Vision, pp. 839-46. 1998.

[28] J. Tukey, "Exploratory Data Analysis", Addison-Wesley, 1977.

[29] J. Weickert, "On discontinuity-preserving optic flow," Proc. Computer Vision and Mobile Robotics Workshop, pp. 115-122, 1998.

[30] J. Weickert and C. Schnörr, "Variational optic flow computation with a spatio-temporal smoothness constraint", Journal of Mathematical Imaging and Vision, vol. 14, pp. 245-255, 2001.

[31] Wikipedia, $http://en.wikipedia.org/wiki/Gamma_correction$, article on Gamma correction, Wikipedia, The Free Encyclopedia.

[32] L.P. Yaroslavsky, "Digital Picture Processing - An Introduction", Springer Verlag, 1985.

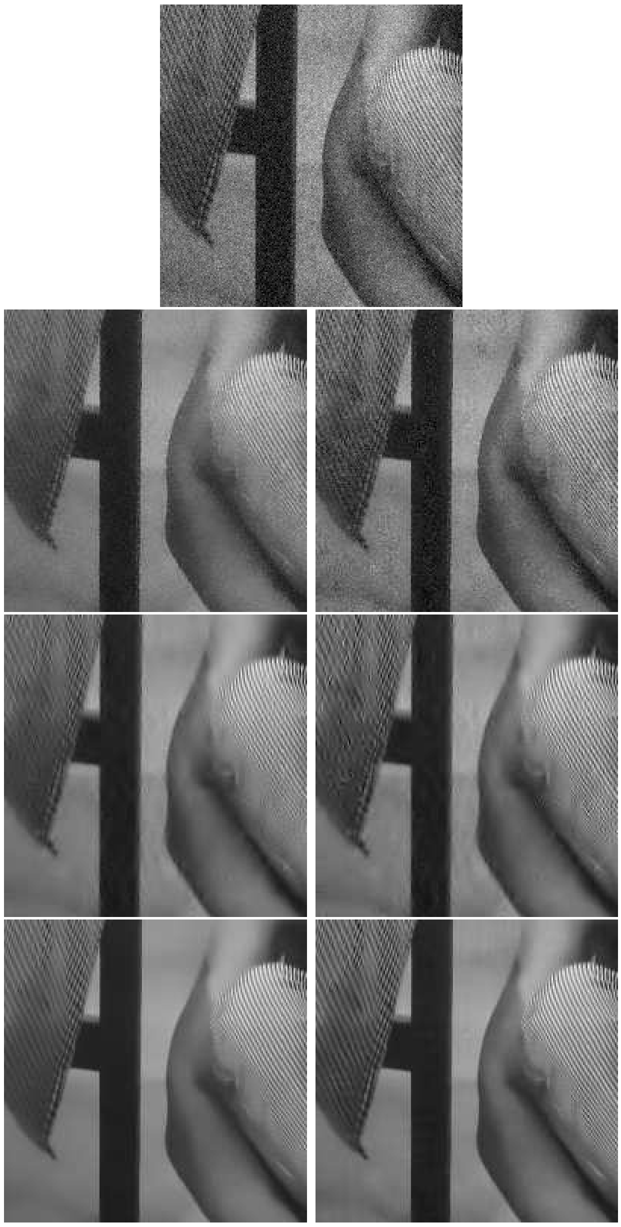

Figure 18: Application of the optimal linear correction (right images) computing the grey level differences on a previously filtered image (left images). From top to bottom: noisy image, bilateral filter, translation invariant wavelet thresholding and the NL-means algorithm. The application of the correction recovers details and enhances edges when applied to the three filtered images. However, noise is recovered inside flat zones by the bilateral filter and wavelet artifacts are enhanced. The application of the correction to the NL-means recovers details and enhances edges without creating any artifact.

Figure 19: Comparison experiment. From top to bottom and left to right: noisy image, sigma or bilateral filter, translation invariant wavelet thresholding and the sliding window NL-means with the optimal linear correction. More image details are kept or even restored in the image denoised image by NL-means. However, a fully denoised image looks unnatural as all natural images have everywhere a microtexture similar to white noise, even if not due to captors. This is why industrial software add some white noise to restored images and movies. It is not the object of the present paper to discuss this esthetic aspect of image restoration.