Chapter 1

Local smoothing neighborhood filters

1.1 Introduction

The *neighborhood filter or sigma filter* is attributed to J.S. Lee [48] in 1983 but goes back to L. Yaroslavsky and the Sovietic image processing theory [76]. This filter is introduced in a denoising framework for the removal of additive white noise

$$v(\mathbf{x}) = u(\mathbf{x}) + n(\mathbf{x}),$$

where $\mathbf{x}$ indicates a pixel site, $v(\mathbf{x})$ is the noisy value, $u(\mathbf{x})$ is the "true" value at pixel $\mathbf{x}$, and $n(\mathbf{x})$ is the noise perturbation. The noise values $n(\mathbf{x})$ and $n(\mathbf{y})$ at different pixels are assumed to be independent random variables and independent of the image value $u(x)$, one talks about "white noise." Generally, $n(\mathbf{x})$ is supposed to follow a Gaussian distribution of zero mean and standard deviation σ.

Lee and Yaroslavsky proposed to smooth the noisy image by averaging only those neighboring pixels having a similar intensity. Averaging is the principle of most denoising methods. The variance law in probability theory ensures that if N noise values are averaged, the noise standard deviation is divided by $\sqrt{N}$. Thus, one should for example find for each pixel nine other pixels in the image with the same color (up to the fluctuations due to noise) in order to reduce the noise by a factor 3. A first idea might be to chose the closest ones. Now, the closest pixels have not necessarily the same color as illustrated in Fig. 1.1. Look at the red pixel placed in the middle of Fig. 1.1. This pixel has five red neighbors and three blue ones. If the color of this pixel 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 red and blue border would be blurred. It is clear that 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 proposed by neighborhood filters.

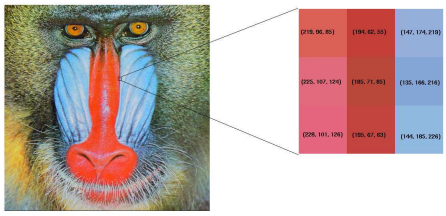

Figure 1.1: The nine pixels in the baboon image on the right have been enlarged. They present a high red-blue contrast. In the red pixels, the first (red) component is stronger. In the blue pixels, the third component, blue, dominates.

The original sigma and neighborhood filter were proposed as an average of the spatially close pixels with a grey level difference lower than a certain threshold h. Thus, for a certain pixel $\mathbf{x}$, the denoised value is the average of pixels in the spatial and intensity neighborhood

$$\{\mathbf{y} \in \Omega \mid \|\mathbf{x}-\mathbf{y}\| < \rho \text{ and } |u(\mathbf{x})-u(\mathbf{y})| < h\}.$$

However, in order to make it coherent with further extensions and facilitate the mathematical development of this chapter, we will write the filter in a continuous framework under a weighted average form. We will denote the neighborhood or sigma filter by *NF* and define it for a pixel $\mathbf{x}$ as

$$NF_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})}\int_{B_\rho(\mathbf{x})} u(\mathbf{y})e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}}d\mathbf{y}, \tag{1.1}$$

where $B_\rho(\mathbf{x})$ is a ball of center $\mathbf{x}$ and radius $\rho > 0$, $h > 0$ is the filtering parameter and $C(\mathbf{x}) = \int_{B_\rho(\mathbf{x})} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}}d\mathbf{y}$ is the normalization factor. The parameter h controls the degree of color similarity needed to be taken into account in the average. This value depends on the noise standard deviation σ, and it was set to 2.5σ in [48] and [76]. We will justify the choice of this value for the h parameter in Section 1.2.

The Yaroslavsky and Lee's filter (1.1) is less known than more recent versions, namely the *SUSAN filter* [69] and the *Bilateral filter* [71]. Both algorithms, instead of considering a fixed spatial neighborhood $B_\rho(\mathbf{x})$, weigh the distance to the reference pixel $\mathbf{x}$,

$$SF_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})}\int_{\Omega} u(\mathbf{y})e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\rho^2}}e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}}d\mathbf{y}, \tag{1.2}$$

where $C(\mathbf{x}) = \int_\Omega e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\rho^2}}e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}}d\mathbf{y}$ is the normalization factor and $\rho > 0$ is now a spatial filtering parameter. Even if the SUSAN algorithm was previously introduced, the whole literature refers to it as the Bilateral filter. Therefore, we shall call this filter by the latter name in subsequent sections.

Figure 1.2: From left to right: noise image, Gaussian convolution, neighborhood filter and Bilateral filter. The neighborhood and Bilateral filters avoid the excessive blurring caused by a Gaussian convolution and preserve all contrasted edges in the image.

The only difference between the neighborhood filter and the Bilateral or SUSAN filter is the way the spatial component is treated. While for the neighborhood filter all pixels within a certain spatial distance are treated uniformly, for the Bilateral or SUSAN filter pixels closer to the reference one are considered as more important. We display in Fig. 1.2 a denoising experience where a Gaussian white noise of standard deviation 10 has been added to a non noisy image. We display the denoised image by both the neighborhood and Bilateral filters. We observe that both filters avoid the excessive blurring caused by a Gaussian convolution and preserve all contrasted edges in the image.

The above denoising experience was applied to color images. In order to clarify how the neighborhood filters are implemented in this case, we remind that each pixel $\mathbf{x}$ is a triplet of values $u(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}), u_3(\mathbf{x}))$, denoting the red, green and blue components. Then, the filter rewrites

$$NF_{h,\rho}u_i(\mathbf{x}) = \frac{1}{C(\mathbf{x})}\int_{B_\rho(\mathbf{x})} u_i(\mathbf{y})e^{-\frac{||u(\mathbf{y})-u(\mathbf{x})||^2}{h^2}}d\mathbf{y}, \tag{1.3}$$

being $||u(\mathbf{y}) - u(\mathbf{x})||^2$ the average of the distances of the three channels,

$$||u(\mathbf{y}) - u(\mathbf{x})||^2 = \frac{1}{3}\sum_{i=1}^{3}|u_i(\mathbf{y}) - u_i(\mathbf{x})|.$$

The same definition applies for the SUSAN and Bilateral filter by incorporating the spatial weighting term. The above definition naturally extends to multispectral images with an arbitrary number of channels. Bennett et al. [7] applied it to multispectral data with an infrared channel and Peng et [56] al for general multispectral data.

The evaluation of the denoising performance of neighborhood filters and comparison with state of the art algorithms is postponed to Section 1.2. In the same section we will present a natural extension of the neighborhood filter, the

NLmeans algorithm, proposed in [12]. This algorithm evaluates the similarity between two pixels $\mathbf{x}$ and $\mathbf{y}$ not only by the intensity or color difference of $\mathbf{x}$ and $\mathbf{y}$ but the difference of intensities in a whole spatial neighborhood.

Figure 1.3: Several applications of the Bilateral filter for increasing values of parameters ρ and h. The parameter ρ increases from top to bottom taking values $\{2, 5, 10\}$ and h increases from left to right taking value $\{5, 10, 25, 100\}$.

The Bilateral filter was also proposed as a filtering algorithm with a filtering scale depending on both parameters h and ρ. Thus, taking several values for these parameters we obtain different filtered images and corresponding residuals in a multi-scale framework. In Fig. 1.3 we display several applications of the Bilateral filter for different values of the parameters h and ρ. We also display the differences between the original and filtered images in Fig. 1.4. For moderated values of h this residual contains details and texture but it doesn't contain contrasted edges. These contrasted information is removed by the bilateral filter only for large values of h. In that case, all pixels are judged as having a similar intensity level and the weight is set taking into account only the spatial component. It is well known that the residual by such an average is proportional to the laplacian of the image. In section 1.3 we will mathematically analyze the asymptotical expansion of the neighborhood residual image.

This detail removal of the Bilateral while conserving very contrasted edges

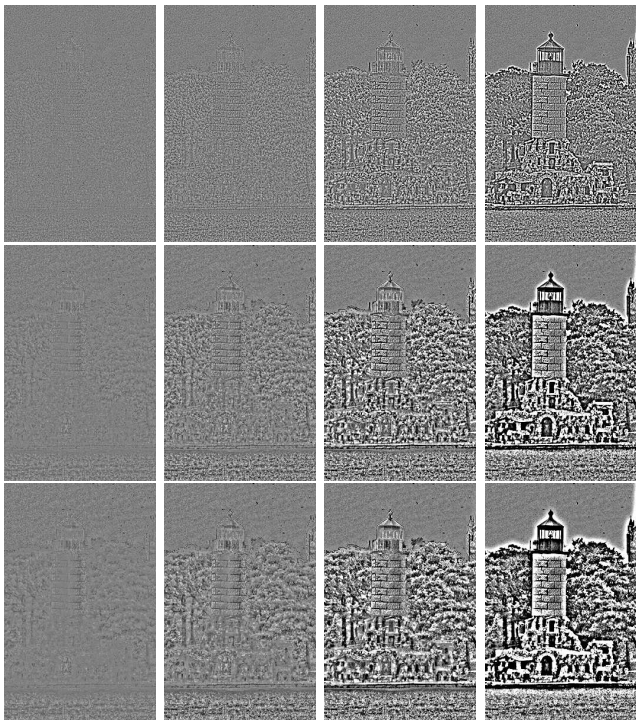

Figure 1.4: Residual differences between original and filtered images in Fig. 1.3. For moderated values of h this residual contains details and texture but it doesn't contain contrasted edges. These contrasted information is removed by the bilateral filter only for large values of h.

is the key in many image and video processing algorithms. Durand and Dorsey [28] use this property in the context of tone mapping whose goal is to compress the intensity values of a high-dynamic range image. The authors isolate the details before compressing the range of the image. Filtered details and texture are added back at the final stage. Similar approaches for image editing are presented by Bae et al. [5], which transfers the visual look of an artist picture onto a casual photograph. Eisemann and Durand [32] and Petschnigg et al. [58] combine the filtered and residual image of a flash and non flash image of the same scene. These two last algorithms, in addition computes the weight configuration in one image of the pair and averages the intensity values of the other image. As we will see in Section 1.4 this is a common feature with iterative versions of neighborhood filters. However, for these applications, both images of the pair must be correctly and precisely registered.

The iteration of the neighborhood filter was not originally considered by the pioneering works of Lee and Yaroslavsky. However, recent applications have shown its interest. The iteration of the filter as a local smoothing operator

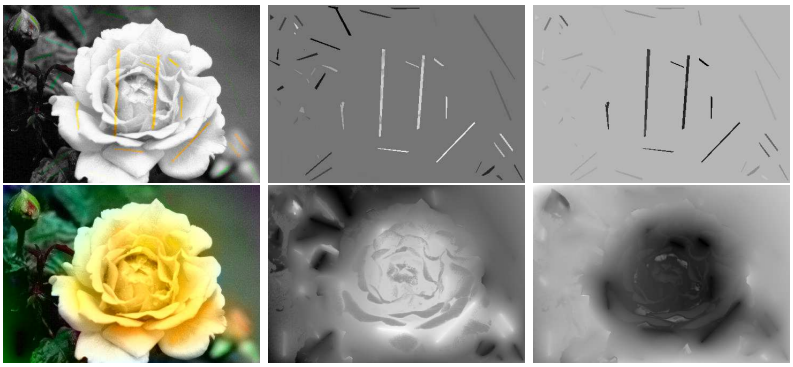

Figure 1.5: Colorisation experiment using the linear iteration of the neighborhood filter. Top Left: Input image with original luminance and initial data on the chromatic components. Bottom Right: Result image by applying the linear neighborhood scheme to the chromatic components using the initial chromatic data as boundary conditions. Top middle and right: initial data on the two chromatic components. Bottom middle and bottom right: final interpolated chromatic components.

tends to piecewise constant images by creating artificial discontinuities in regular zones. Barash et al. [6] showed that an iteration of the neighborhood filter was equivalent to a step of a certain numerical scheme of the classical Perona-Malik equation [57]. A complete proof of the equivalence between the neighborhood filter and the Perona-Malik equation was presented in [13] including a modification of the filter to avoid the creation of shocks inside regular parts of the image. Another theoretical explanation of the shock effect of the neighborhood filters can be found in Van de Weijer and van den Boomgaard [72] and Comaniciu [20]. Both papers show that the iteration of the neighborhood filter process makes points tend to the local modes of the histogram but in a different framework. The first for images and the second for any dimensional clouds of points. This discontinuity or shock creation in regular zones of the image is not desirable for filtering or denoising applications. However, it can be used for image or video editing as proposed by Winnemoller et al. [74] in order to simplify video content and achieve a cartoon look.

Even if it may seem paradoxical, linear schemes have showed to be more useful than non linear ones for iterating the neighborhood filter. That is, the weight distribution for each pixel is computed once and is maintained during the whole iteration process. We will show in section 1.4, that by computing the weights on an image and keeping them constant during the iteration process, an histogram concentration phenomenon makes the filter a powerful segmentation algorithm. The same iteration is useful for linearly diffuse or filter any initial data or seeds as proposed by Grady et al [38] for medical image segmentation or [11] for colorization (see Fig. 1.5 for an example). The main hypothesis for this seed diffusion algorithms is that pixels having a similar grey level value should

be related and are likely to belong to the same object. Thus, pixels of different sites are related as in a graph with a weight depending on the grey level distance. The iteration of the neighborhood filter on the graph is equivalent to the solution of the heat equation on the graph by taking the graph Laplacian. Eigenvalues and eigenvectors of such a graph Laplacian can be computed allowing the design of Wiener and thresholding filters on the graph, see [70] and [59], [60] for more details.

Both the neighborhood filter and the NLmeans have been adapted and extended for other types of data and other image processing tasks: for 3D data set points [43], [35], [17][80], [26], and [42], *demosaicking*, the operation which transforms the "R or G or B" raw image in each camera into an "R and G and B" image [63], [15], [51], *movie colorization*, [34] and [49]; *image inpainting* by proposing a non local image inpainting variational framework with a unified treatment of geometry and texture [2] (see also [75]) ; *Zooming* by a fractal like technique where examples are taken from the image itself at different scales [29]; *movie flicker stabilization* [24], compensating spurious oscillations in the colors of successive frames; *super-resolution* an image zooming method by which several frames from a video, or several low resolution photographs, can be fused into a larger image [62]. The main point of this super-resolution technique is that it gives up an explicit estimate of the motion, allowing actually for a multiple motion, since a block can look like several other blocks in the same frame. The very same observation is made in [30] for devising a super-resolution algorithm and in [33], [22].

1.2 Denoising

1.2.1 Analysis of neighborhood filter as a denoising algorithm

In this section we will further investigate the neighborhood filter behavior as a denoising algorithm. We will consider the simplest neighborhood filter version which averages spatially close pixels with a intensity difference lower than a certain threshold h. By classical probability theory, the average of N random and i.i.d values has a variance N times smaller than the variance of the original values. However, this theoretical reduction is not observed when applying neighborhood filters.

In order to evaluate the noise reduction capability of the neighborhood filter, we apply it to a noise sample and evaluate the variance of the filtered sample. Let us suppose that we observe the realization of a white noise at a pixel $\mathbf{x}$, $n(\mathbf{x}) = a$. The nearby pixels with a intensity difference lower than h, will be independent and identically distributed with probability distribution function the restriction of the Gaussian to the interval $(a-h, a+h)$. If the research zone is large enough, then the average value will tend to the expectation of such a variable. Thus, the increase of the research zone and therefore of the number of pixels being averaged does not increase the noise reduction capability of the

filter. Such a noise reduction factor is computed in next result.

Theorem 1.2.1 *Assume that the $n(i)$ are i.i.d. with zero mean and variance σ^2. Then, the filtered noise by the neighborhood filter NF_h satisfies,*

(i) The noise reduction depends only on the value of h,

$$Var\ NF_h\, n(\boldsymbol{x}) = f(\frac{h}{\sigma})\,\sigma^2,$$

where

$$f(x) = \frac{1}{(2\pi)^{3/2}} \int_{\mathbb{R}} \frac{1}{\beta^2(a,x)} (e^{2xa} - 1)^2 e^{(a+x)^2} e^{\frac{-a^2}{2}}\, da,$$

is a decreasing function with $f(0) = 1$ and $\lim_{x->\infty} f(x) = 0$.

(ii) The values $NF_h n(\boldsymbol{x})$ and $NF_h n(\boldsymbol{y})$ are uncorrelated for $\boldsymbol{x} \neq \boldsymbol{y}$.

The function $f(x)$ is plotted in Fig. 1.6. The noise reduction increases as the ratio h/σ also does. We see that $f(x)$ is near zero for values of x over 2.5 or 3, that is, values of h over $2.5\,\sigma$ or $3\,\sigma$ which justifies the values proposed in the original papers by Lee and Yaroslavsky. However, for a gaussian variable, the probability of observing values at a distance of the average larger than 2.5 or 3 times the standard deviation is very small. Thus, by taking these values we excessively increase the probability of mismatching pixels of different objects. Thus, close objects with an intensity contrast lower than 3σ will not be correctly denoised. This explains the decreasing performance of the neighborhood filter as the noise standard deviation increases.

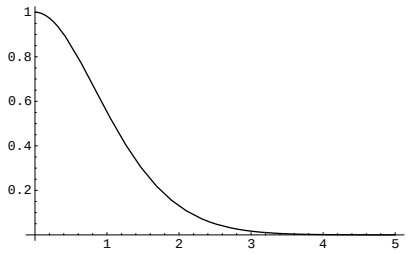

Figure 1.6: Noise reduction function $f(x)$ given by theorem 1.2.1.

The previous theorem also tells us that the denoised noise values are still uncorrelated once the filter has been applied. This is easily justified since we showed that as the size ρ of the neighborhood increases the filtered value tends to the expectation of the Gauss distribution restricted to the interval $(n(\mathbf{x}) - h, n(\mathbf{x}) + h)$. The filtered value is therefore a deterministic function of $n(\mathbf{x})$ and h. Independent random variables are mapped by a deterministic function on independent variables.

This property may seem anecdotic 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 [3].

The only difference between the neighborhood filter and the Bilateral or SUSAN filter is the way the spatial component is treated. While for the classical neighborhood all pixels within a certain distance are treated equally, for the Bilateral filter pixels closer to the reference pixel are more important. Even if this can seem a slight difference, this is crucial from a qualitative point of view, that is, the creation of artifacts.

It is easily shown that by introducing the weighting function on the intensity difference instead of a non weighted average does not modify the second property of Theorem 1.2.1, and the denoised noise values are still uncorrelated if ρ is large enough. However, the introduction of the spatial kernel by the Bilateral or SUSAN filter affects this property. Indeed, the introduction of a spatial decay of the weights makes denoised values at close positions to be correlated.

There are two ways to show how denoising algorithms behave when they are applied to a noise sample. One of them is to find a mathematical proof that the pixels remain independent (or at least uncorrelated) and identically distributed random variables. The experimental device simply is to observe the effect of denoising on the simulated realization of a white noise. Figure 1.11 displays the filtered noises for the neighborhood filter, the Bilateral filter and other state of the art denoising algorithms.

1.2.2 Neighborhood filter extension: the NL-means algorithm

Now in a very general sense inspired by the neighborhood filter, one can define as "neighborhood of a pixel $\mathbf{x}$" any set of pixels $\mathbf{y}$ in the image such that a window around $\mathbf{y}$ looks like a window around $\mathbf{x}$. All pixels in that neighborhood can be used for predicting the value at $\mathbf{x}$, as was shown in [23, 31] for texture synthesis and in [21, 81] for inpainting purposes. The fact that such a self-similarity exists is a regularity assumption, actually more general and more accurate than all regularity assumptions we consider when dealing with local smoothing filters and it also generalizes a periodicity assumption of the image.

Let v be the noisy image observation defined on a bounded domain $\Omega \subset \mathbb{R}^2$, and let $\mathbf{x} \in \Omega$. The NL-means algorithm estimates the value of $\mathbf{x}$ as an average of the values of all the pixels whose Gaussian neighborhood looks like the neighborhood of $\mathbf{x}$,

$$NL(v)(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_\Omega e^{-\frac{(G_a * |v(\mathbf{x}+.) - v(\mathbf{y}+.)|^2)(0)}{h^2}} \, v(\mathbf{y}) \, d\mathbf{y},$$

where G_a is a Gaussian kernel with standard deviation a, h acts as a filtering parameter, and $C(\mathbf{x}) = \int_\Omega e^{-\frac{(G_a * |v(\mathbf{x}+.) - v(\mathbf{z}+.)|^2)(0)}{h^2}} d\mathbf{z}$ is the normalizing factor.

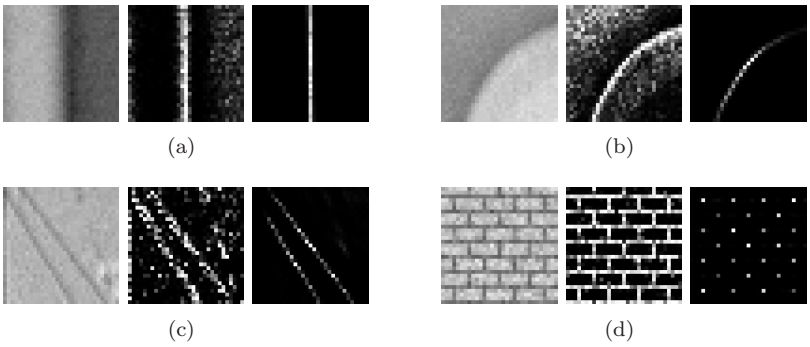

Figure 1.7: Weight distribution of NL-means, the bilateral filter and the anisotropic filter used to estimate the central pixel in four detail images. On the two right-hand side images of each triplet, we display the weight distribution used to estimate the central pixel of the left image by the neighborhood and the NL-means algorithm. (a) In straight edges, the weights are distributed in the direction of the level line (as the mean curvature motion). (c) On curved edges, the weights favor pixels belonging to the same contour or level line, which is a strong improvement with respect to the mean curvature motion. In the cases of (e) and (f), the weights are distributed across the more similar configurations, even though they are far away from the observed pixel. This shows a behavior similar to a nonlocal neighborhood filter or to an ideal Wiener filter.

We recall that

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

We will see that the use of an entire window around the compared points make this comparison more robust to noise. For the moment, we will compare the weighting distributions of both filters. Fig. 1.7 illustrates how the NL-means algorithm chooses in each case a weight configuration adapted to the local geometry of the image. Then, the NL-means algorithm seems to provide a feasible and rational method to automatically take the best of all classical denoising algorithms, reducing for every possible geometric configuration the mismatched averaged pixels. It preserves flat zones as the Gaussian convolution and straight edges as the anisotropic filtering while still restores corners or curved edges and texture.

Due to the nature of the algorithm, one of the most favorable cases is the textural case. Texture images have a large redundancy. For each pixel, many similar samples can be found in the image with a very similar configuration, leading to a noise reduction and a preservation of the original image. In Fig. 1.8 one can see an example with a Brodatz texture. The Fourier transform of the noisy and restored images shows the ability of the algorithm to preserve the main features even in the case of high frequencies.

The NL-means seems to naturally extend the Gaussian, anisotropic and neighborhood filtering. But it is not easily related to other state of the art

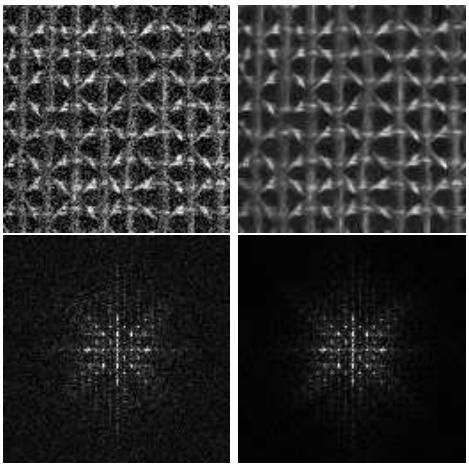

Figure 1.8: NL-means denoising experiment with a Brodatz texture image. Left: Noisy image with standard deviation 30. Right: NL-means restored image. The Fourier transform of the noisy and restored images show how main features are preserved even at high frequencies.

denoising methods as the Total Variation minimization [64], the wavelet thresholding [27, 19] or the local DCT empirical Wiener filters [77]. For this reason we compare these methods visually in artificial denoising experiences (see [12] for a more comprehensive comparison).

Figure 1.9 illustrates the fact that a nonlocal algorithm is needed for the correct reconstruction of periodic images. Local smoothing filters, Wiener and thresholding methods are not able to reconstruct the wall pattern. Only NL-means and the global Fourier–Wiener filter reconstruct the original texture. The Fourier–Wiener filter is based on a global Fourier transform which is able to capture the periodic structure of the image in a few coefficients. But this only is an ideal filter: the Fourier transform of the original image is being used. Figure 1.7(d) shows how NL-means chooses the correct weight configuration and explains the correct reconstruction of the wall pattern.

The NL-means algorithm is not only able to restore periodic or texture images. Natural images also have enough redundancy to be restored. For example, in a flat zone, one can find many pixels lying in the same region and with similar configurations. In a straight or curved edge a complete line of pixels with a similar configuration is found. In addition, the redundancy of natural images allows us to find many similar configurations in far away pixels.

Figure 1.10 shows that wavelet and DCT thresholding are well adapted to the recovery of oscillatory patterns. Although some artifacts are noticeable in both solutions, the stripes are well reconstructed. The DCT transform seems to be more adapted to this type of texture, and stripes are a little better reconstructed. For a much more detailed comparison between sliding window transform domain filtering methods and wavelet threshold methods, we refer the reader to [78].

NL-means also performs well on this type of texture, due to its high degree of redundancy.

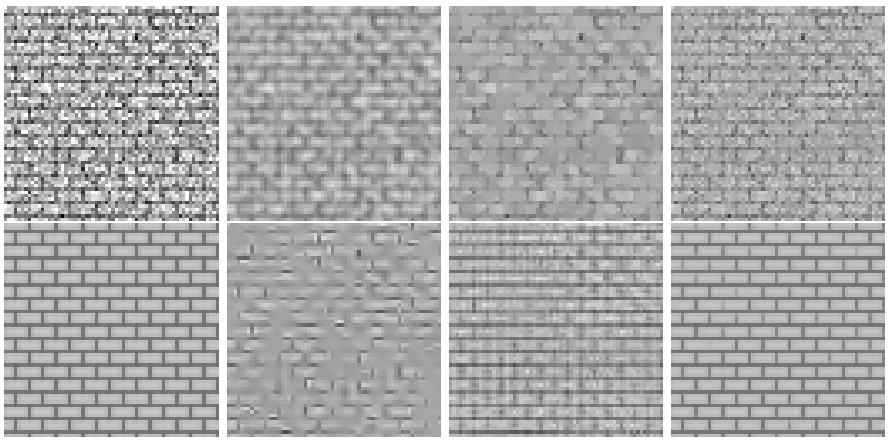

Figure 1.9: Denoising experience on a periodic image. From left to right and from top to bottom: noisy image (standard deviation 35), Gauss filtering, total variation, neighborhood filter, Wiener filter (ideal filter), TIHWT, DCT empirical Wiener filtering, and NL-means.

Figures 1.11 displays the application of the denoising methods to a white noise. We display the filtered noise.

1.2.3 Extension to movies

Averaging filters are easily extended to the denoising of image sequences and video. The denoising algorithms involve indiscriminately pixels not belonging only to same frame but also the previous and posterior ones.

In many cases, this straightforward extension cannot correctly deal with moving objects. For that reason, state of the art movie filters are motion compensated (see [10] for a comprehensive review). 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 an averaging filter can be applied. The motion compensation neighborhood filter was proposed by Ozkan et al [55]. We illustrate in Fig. 1.14 the improvement obtained with the proposed compensation.

One of the major difficulties in motion estimation is the ambiguity of trajectories, the so called *aperture problem*. This problem is illustrated in Fig. 1.12. 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 criterion.

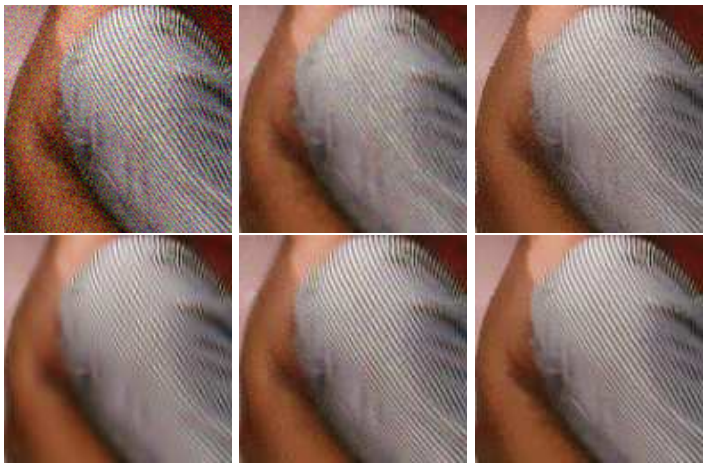

Figure 1.10: Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation 35), total variation, neighborhood filter, translation invariant hard thresholding (TIHWT), empirical Wiener and NL-means.

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 (see [14] for more details on this discussion). The algorithm favors pixels with a similar local configuration, 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 (see Fig. 1.13).

1.3 Asymptotic

1.3.1 PDE models and local smoothing filters

According to Shannon's theory, a signal can be correctly represented by a discrete set of values, the "samples", only if it has been previously smoothed. Let us start with u_0 the physical image, a real function defined on a bounded domain $\Omega \subset \mathbb{R}^2$. Then a blur optical kernel k is applied, i.e. u_0 is convolved with k to obtain an observable signal $k * u_0$. Gabor remarked in 1960 that the difference between the original and the blurred images is roughly proportional

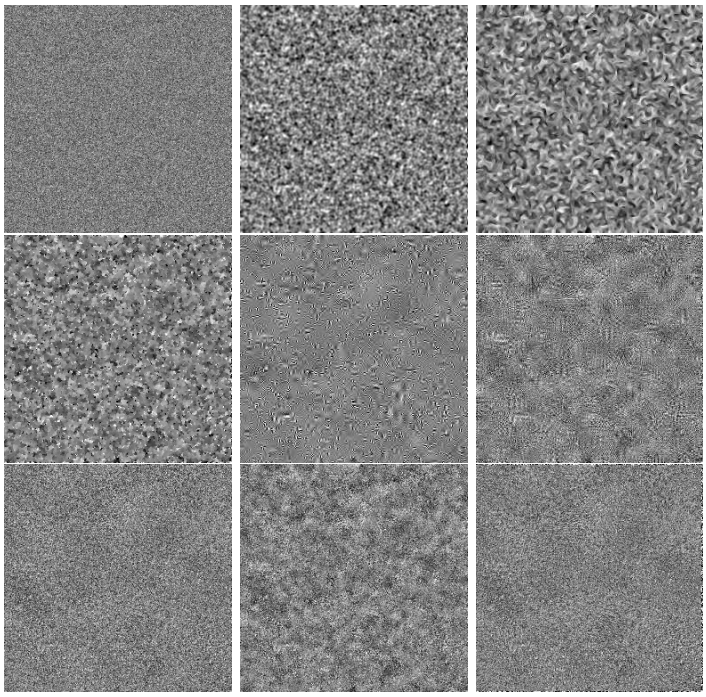

Figure 1.11: The noise to noise criterion. From left to right and from top to bottom: original noise image of standard deviation 20, Gaussian convolution, anisotropic filtering, total variation, TIHWT, DCT empirical Wiener filter, neighborhood filter, Bilateral filter and the NL-means. Parameters have been fixed for each method so that the noise standard deviation is reduced by a factor 4. The filtered noise by the Gaussian filter and the total variation minimization are quite similar, even if the first one is totally blurred and the second one has created many high frequency details. The filtered noise by the hard wavelet thresholding looks like a constant image with superposed wavelets. The filtered noise by the neighborhood filter and the NL-means algorithm looks like a white noise. This is not the case for the Bilateral filter, where low frequencies of noise are enhanced because of the spatial decay.

to its Laplacian, $\Delta u = u_{xx} + u_{yy}$. In order to formalize this remark, we have to notice that k is spatially concentrated, and that we may introduce a scale parameter for k, namely $k_h(\mathbf{x}) = h^{-1}k(h^{-\frac{1}{2}}\mathbf{x})$. If, for instance, u is C^2 and bounded and if k is a radial function in the Schwartz class, then

$$\frac{u_0 * k_h(\mathbf{x}) - u_0(\mathbf{x})}{h} \to c\Delta u_0(\mathbf{x}).$$

Hence, when h gets smaller, the blur process looks more and more like the heat equation

$$u_t = c\Delta u, \quad u(0) = u_0.$$

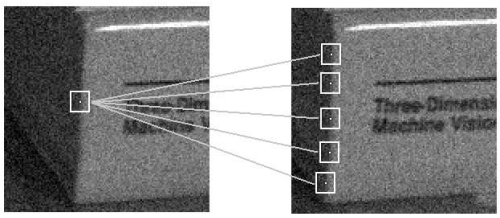

Figure 1.12: Aperture problem and the ambiguity of trajectories are the most difficult problem in motion estimation: There can be many good matches. The motion estimation algorithms must pick one.

Thus, Gabor established a first relationship between local smoothing operators and PDE's. The classical choice for k is the Gaussian kernel.

Remarking that the optical blur is equivalent to one step of the heat equation, Gabor deduced that we can, to some extent, deblur an image by reversing the time in the heat equation, $u_t = -\Delta u$. Numerically, this amounts to subtracting the filtered version from the original image,

$$u - G_h * u = -h^2 \Delta u + o(h^2).$$

This leads to considering the reverse heat equation as an image restoration, ill-posed though it is. The time-reversed heat equation was stabilized in the Osher-Rudin shock filter [54] who proposed

$$u_t = -\text{sign}(\mathcal{L}(u))|Du|, \tag{1.4}$$

where the propagation term $|Du|$ is tuned by the sign of an edge detector $\mathcal{L}(u)$. The function $\mathcal{L}(u)$ changes sign across the edges where the sharpening effect therefore occurs. In practice, $\mathcal{L}(u) = \Delta u$ and the equation is related to a reverse heat equation.

The early Perona-Malik "anisotropic diffusion" [57] is directly inspired from the Gabor remark. It reads

$$u_t = \text{div}(g(|Du|^2)Du), \tag{1.5}$$

where $g : [0, +\infty) \to [0, +\infty)$ is a smooth decreasing function satisfying $g(0) = 1$, $\lim_{s\to+\infty} g(s) = 0$. This model is actually related to the preceding ones. Let us consider the second derivatives of u in the directions of Du and $Du^\perp$,

$$u_{\eta\eta} = D^2u(\frac{Du}{|Du|}, \frac{Du}{|Du|}), \quad u_{\xi\xi} = D^2u(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}).$$

Then, the equation (1.5) can be rewritten as

$$u_t = g(|Du|^2)u_{\xi\xi} + h(|Du|^2)u_{\eta\eta}, \tag{1.6}$$

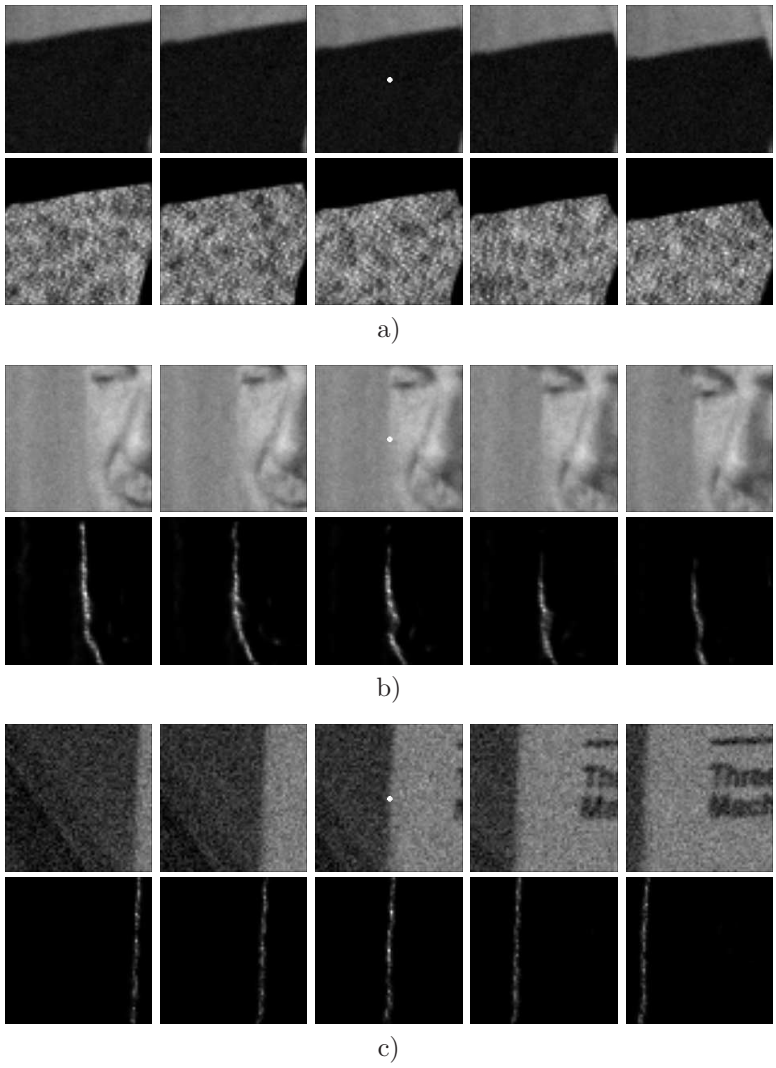

Figure 1.13: Weight distribution of NL-means applied to a movie. In a), b) and c) the first row shows a five frames image sequence. In the second row, the weight distribution used to estimate the central pixel (in white) of the middle frame is shown. The weights are equally distributed over the successive 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.

where $h(s) = g(s) + 2sg'(s)$. Perona and Malik proposed the function $g(s) = \frac{1}{1+s/k}$. In this case, the coefficient of the first term is always positive and this term therefore appears as a one dimensional diffusion term in the orthogonal

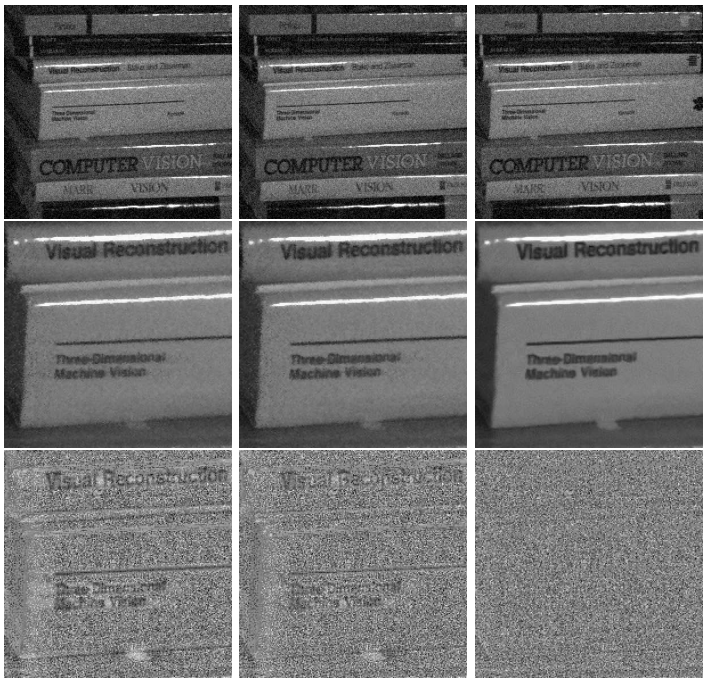

Figure 1.14: Comparison of static filters, motion compensated filters and NL-means applied to an image sequence. Top: three frames of the sequence are displayed. Middle and left to right: neighborhood filter, motion compensated neighborhood filter and the NL-means. (AWA). Bottom: the noise removed by each method (difference between the noisy and filtered frame). Motion compensation improves the static algorithms by better preserving the details and creating less blur. 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) has almost no noticeable structure in its removed noise. As a consequence, the filtered sequence has kept more details and is less blurred.

direction to the gradient. The sign of the second coefficient, however, depends on the value of the gradient. When $|Du|^2 < k$ this second term appears as a one dimensional diffusion in the gradient direction. It leads to a reverse heat equation term when $|Du|^2 > k$.

The Perona-Malik model has got many variants and extensions. Tannenbaum and Zucker [45] proposed, endowed in a more general shape analysis framework, the simplest equation of the list,

$$u_t = |Du|\mathrm{div}\left(\frac{Du}{|Du|}\right) = u_{\xi\xi}.$$

This equation had been proposed some time before in another context by Sethian [67] as a tool for front propagation algorithms. This equation is a "pure" diffusion in the direction orthogonal to the gradient and is equivalent to the

anisotropic filter *AF* [40],

$$AF_h u(\mathbf{x}) = \int G_h(t)\, u(\mathbf{x} + t\xi) dt,$$

where $\xi = Du(\mathbf{x})^{\perp}/|Du(\mathbf{x})|$ and $G_h(t)$ denotes the one-dimensional Gauss function with variance h^2.

This diffusion is also related to two models proposed in image restoration. The Rudin-Osher-Fatemi [64] total variation model leads to the minimization of the total variation of the image $TV(u) = \int |Du|$, subject to some constraints. The steepest descent of this energy reads, at least formally,

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{Du}{|Du|}\right) \tag{1.7}$$

which is related to the mean curvature motion and to the Perona-Malik equation when $g(|Du|^2) = \frac{1}{|Du|}$. This particular case, which is not considered in [57], yields again (1.7). An existence and uniqueness theory is available for this equation [1].

1.3.2 Asymptotic behavior of neighborhood filters (dimension 1)

Let u denote a one-dimensional signal defined on an interval $I \subset \mathbb{R}$ and consider the neighborhood filter

$$NF_{h,\rho} u(x) = \frac{1}{C(x)} \int_{x-\rho}^{x+\rho} u(y) e^{-\frac{|u(y)-u(x)|^2}{h^2}} dy, \tag{1.8}$$

where $C(x) = \int_{x-\rho}^{x+\rho} e^{-\frac{|u(y)-u(x)|^2}{h^2}} dy$.

The following theorem describes the asymptotical behavior of the neighborhood filter in 1D. The proof of this theorem and next ones in this section can be found in [13]. 1

Theorem 1.3.1 *Suppose $u \in C^2(I)$, and let $\rho, h, \alpha > 0$ such that $\rho, h \to 0$ and $h = O(\rho^\alpha)$. Consider the continuous function $g(t) = \frac{te^{-t^2}}{E(t)}$, for $t \neq 0$, $g(0) = \frac{1}{2}$, where $E(t) = 2\int_0^t e^{-s^2}\, ds$. Let f be the continuous function*

$$f(t) = \frac{g(t)}{t^2} + g(t) - \frac{1}{2t^2}, \quad f(0) = \frac{1}{6}.$$

Then, for $x \in \mathbb{R}$,

1. *If $\alpha < 1$, $\quad NF_{h,\rho} u(x) - u(x) \simeq \frac{u''(x)}{6}\rho^2$.*
2. *If $\alpha = 1$, $\quad NF_{h,\rho} u(x) - u(x) \simeq f(\frac{\rho}{h}|u'(x)|)\, u''(x)\, \rho^2$.*
3. *If $1 < \alpha < \frac{3}{2}$, $\quad NF_{h,\rho} u(x) - u(x) \simeq g(\rho^{1-\alpha}|u'(x)|)\, u''(x)\, \rho^2$.*

According to Theorem 1.3.1, the neighborhood filter makes the signal evolve proportionally to its second derivative. The equation $u_t = cu''$ acts as a smoothing or enhancing model depending on the sign of c. Following the previous theorem, we can distinguish three cases depending on the values of h and ρ. First, if h is much larger than ρ the second derivative is weighted by a positive constant and the signal is therefore filtered by a heat equation. Second, if h and ρ have the same order, the sign and magnitude of the weight is given by $f(\frac{\rho}{h}|u'(x)|)$. As the function f takes positive and negative values (see Fig. 1.18), the filter behaves as a filtering/enhancing algorithm depending on the magnitude of $|u'(x)|$. If B denotes the zero of f, then a filtering model is applied wherever $|u'| < B\frac{h}{\rho}$ and an enhancing model wherever $|u'| > B\frac{h}{\rho}$. The intensity of the enhancement tends to zero when the derivative tends to infinity. Thus, points x where $|u'(x)|$ is large are not altered. The transition of the filtering to the enhancement model creates a singularity in the filtered signal. In the last case, ρ is much larger than h and the sign and magnitude of the weight is given by $g(\frac{\rho}{h}|u'(x)|)$. Function g is positive and decreases to zero. If the derivative of u is bounded then $\frac{\rho}{h}|u'(x)|$ tends to infinity and the intensity of the filtering to zero. In this case, the signal is hardly modified.

In summary, a neighborhood filter in dimension 1 shows interesting behavior only if ρ and h have the same order of magnitude; in which case the neighborhood filter behaves like a Perona-Malik equation. It enhances edges with a gradient above a certain threshold and smoothes the rest.

Figure 1.15 illustrates the behavior of the one dimensional neighborhood filter. The algorithm is iterated until the steady state is attained on a sine signal for different values of the ratio ρ/h. The results of the experiment corroborate the asymptotical expansion of Theorem 1.3.1. In the first experiment, $\rho/h = 10^{-8}$ and the neighborhood filter is equivalent to a heat equation. The filtered signal tends to a constant. In the second experiment, $\rho/h = 10^8$ and the value $g(\frac{\rho}{h}|u'|)$ is nearly zero. As predicted by the theorem, the filtered signal is nearly identical to the original one. The last two experiments illustrate the filtering/enhancing behavior of the algorithm when h and ρ have similar values. As predicted, an enhancing model is applied where the derivative is large. Many singularities are being created because of the transition of the filtering to the enhancing model. Unfortunately, the number of singularities and their position depend upon the value of ρ/h. This behavior is explained by Theorem 1.3.1(2). Figure 1.22 illustrates the same effect in the 2D case.

The filtering/enhancing character of the neighborhood filter is very different from a pure enhancing algorithm like the Osher-Rudin shock filter. Figures 1.16 and 1.17 illustrate these differences. In Fig. 1.16, the minimum and the maximum of the signal have been preserved by the shock filter, while these two values have been significantly reduced by the neighborhood filter. This filtering/enhancing effect is optimal when the signal is noisy. Figure 1.17 shows how the shock filter creates artificial steps due to the fluctuations of noise, while the neighborhood filter reduces the noise avoiding any spurious shock. Parameter h has been chosen larger than the amplitude of noise in order to remove it. Choosing an intermediate value of h, artificial steps could also be

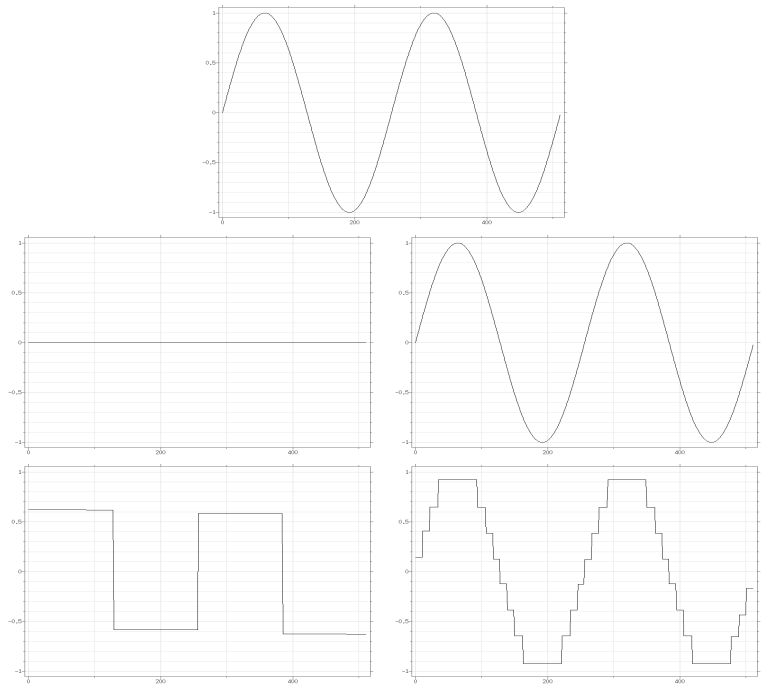

Figure 1.15: One dimensional neighborhood filter experiment. The neighborhood filter is iterated until the steady state is attained for different values of the ratio ρ/h. Top: Original sine signal. Middle left: filtered signal with $\rho/h = 10^{-8}$. Middle right: filtered signal with $\rho/h = 10^{8}$. Bottom left: filtered signal with $\rho/h = 2$. Bottom right: filtered signal with $\rho/h = 5$. The examples corroborate the results of Theorem 1.3.1. If ρ/h tends to zero the algorithm behaves like a heat equation and the filtered signal tends to a constant. If, instead, ρ/h tends to infinity the signal is hardly modified. If ρ and h have the same order, the algorithm presents a filtering/enhancing dynamic. Singularities are created due to the transition of smoothing to enhancement. The number of enhanced regions strongly depends upon the ratio $\frac{\rho}{h}$ as illustrated in the bottom figures.

generated on points where the noise amplitude is above this parameter value.

1.3.3 The two dimensional case

The following theorem extends previous result to the two dimensional case.

Theorem 1.3.2 *Suppose $u \in C^2(\Omega)$, and let $\rho, h, \alpha > 0$ such that $\rho, h \to 0$ and $h = O(\rho^\alpha)$. Let us consider the continuous function $\tilde{g}$ defined by $\tilde{g}(t) = \frac{1}{3}\frac{te^{-t^2}}{E(t)}$, for $t \neq 0$, $\tilde{g}(0) = \frac{1}{6}$, where $E(t) = 2\int_0^t e^{-s^2}\, ds$. Let $\tilde{f}$ be the continuous function*

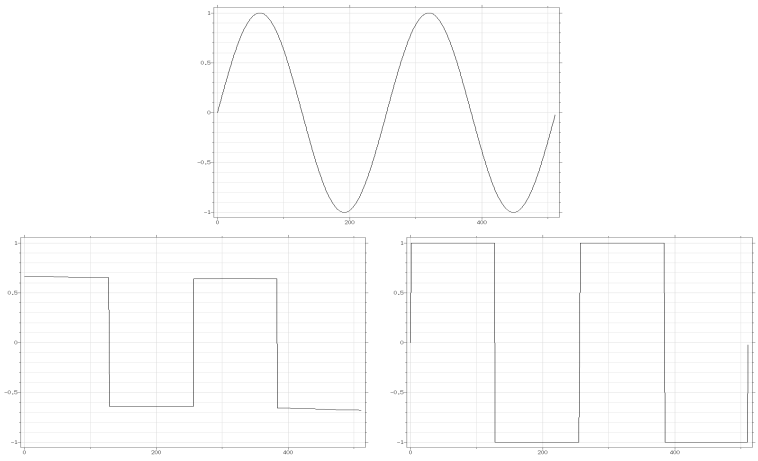

Figure 1.16: Comparison between the neighborhood filter and the shock filter. Top: Original signal. Bottom left: application of the neighborhood filter. Bottom right: application of the shock filter. The minimum and the maximum of the signal have been preserved by the shock filter and reduced by the neighborhood filter. This fact illustrates the filtering/enhancing character of the neighborhood filter compared with a pure enhancing filter.

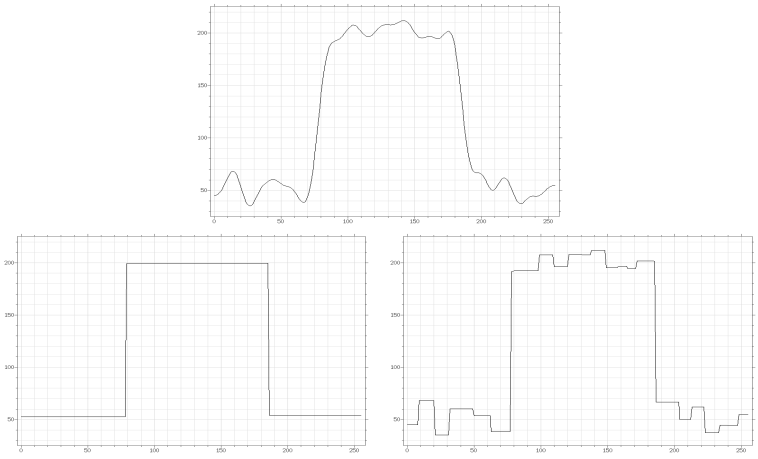

Figure 1.17: Comparison between the neighborhood filter and the shock filter. Top: Original signal. Bottom left: application of the neighborhood filter. Bottom right: application of the shock filter. The shock filter is sensitive to noise and creates spurious steps. The filtering/enhancing character of the neighborhood filter avoids this effect.

defined by

$$\tilde{f}(t) = 3\tilde{g}(t) + \frac{3\tilde{g}(t)}{t^2} - \frac{1}{2t^2}, \quad \tilde{f}(0) = \frac{1}{6}.$$

Then, for $\boldsymbol{x} \in \Omega$,

1. *If* $\alpha < 1$,
$$NF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \frac{\triangle u(\boldsymbol{x})}{6}\rho^2.$$

2. *If* $\alpha = 1$,
$$NF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \left[\tilde{g}(\frac{\rho}{h}\,|Du(\boldsymbol{x})|)\; u_{\xi\xi}(\boldsymbol{x}) + \tilde{f}(\frac{\rho}{h}\,|Du(\boldsymbol{x})|)\; u_{\eta\eta}(\boldsymbol{x})\right]\rho^2$$

3. *If* $1 < \alpha < \frac{3}{2}$,
$$NF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \tilde{g}\left(\rho^{1-\alpha}|Du(\boldsymbol{x})|\right)\left[\; u_{\xi\xi}(\boldsymbol{x}) + 3u_{\eta\eta}(\boldsymbol{x})\right]\rho^2.$$

being $\xi = Du(\boldsymbol{x})^{\perp}/|Du(\boldsymbol{x})|$ *and* $\eta = Du(\boldsymbol{x})/|Du(\boldsymbol{x})|$.

According to Theorem 1.3.2 the two-dimensional neighborhood filter acts as an evolution PDE with two terms. The first term is proportional to the second derivative of u in the direction $\xi = Du(\mathbf{x})^{\perp}/|Du(\mathbf{x})|$, which is tangent to the level line passing through $\mathbf{x}$. The second term is proportional to the second derivative of u in the direction $\eta = Du(\mathbf{x})/|Du(\mathbf{x})|$ which is orthogonal to the level line passing through $\mathbf{x}$. Like in the one dimensional case, the evolution equations $u_t = c_1 u_{\xi\xi}$ and $u_t = c_2 u_{\eta\eta}$ act as filtering or enhancing models depending on the signs of c_1 and c_2. Following the previous theorem, we can distinguish three cases, depending on the values of h and ρ.

First, if h is much larger than ρ, both second derivatives are weighted by the same positive constant. Thus, the sum of the both terms is equivalent to the Laplacian of u, Δu, and we get back to Gaussian filtering.

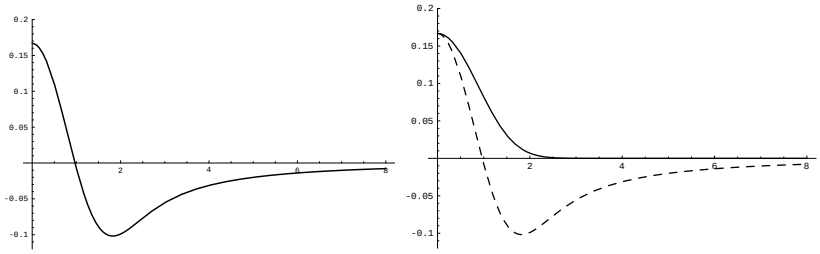

Figure 1.18: Weight functions of Theorems 1.3.1 and 1.3.2 when h and ρ have the same order. Left: Function f of Theorem 1.3.1. Right: Functions $\tilde{g}$ (continuous line) and $\tilde{f}$ (dashed line) of Theorem 1.3.2.

Second, if h and ρ have the same order of magnitude, the neighborhood filter behaves as a filtering/enhancing algorithm. The coefficient of the diffusion in the tangential direction, $u_{\xi\xi}$, is given by $\tilde{g}(\frac{\rho}{h}|Du|)$. The function $\tilde{g}$ is positive and decreasing. Thus, there is always diffusion in that direction. The weight of

the normal diffusion, $u_{\eta\eta}$, is given by $\tilde{f}(\frac{\rho}{h}|Du|)$. As the function $\tilde{f}$ takes positive and negative values (see Fig. 1.18), the filter behaves as a filtering/enhancing algorithm in the normal direction and depending on $|Du|$. If $\tilde{B}$ denotes the zero of $\tilde{f}$, then a filtering model is applied wherever $|Du| < \tilde{B}\frac{h}{\rho}$ and an enhancing strategy wherever $|Du| > \tilde{B}\frac{h}{\rho}$. The intensity of the filtering in the tangent diffusion and the enhancing in the normal diffusion tend to zero when the gradient tends to infinity. Thus, points with a very large gradient are not altered.

Finally, if ρ is much larger than h, the value $\frac{\rho}{h}$ tends to infinity and then the filtering magnitude $\tilde{g}(\frac{\rho}{h}|Du|)$ tends to zero. Thus, the original image is hardly altered. Let us mention that similar calculations were performed in a particular case for the neighborhood median filter by Masnou [52].

We observe that when ρ and h have the same order, the neighborhood filter asymptotically behaves like a Perona-Malik model. Let us be more specific about this comparison. Taking $g(s) = \tilde{g}(s^{\frac{1}{2}})$ in the Perona-Malik equation (1.6), we obtain

$$u_t = \tilde{g}(|Du|)u_{\xi\xi} + \tilde{h}(|Du|)u_{\eta\eta}, \tag{1.9}$$

where $\tilde{h}(s) = \tilde{g}(s) + s\tilde{g}'(s)$. Thus, the Perona-Malik model and the neighborhood filter can be decomposed in the same way and with exactly the same weight in the tangent direction. Then the function $\tilde{h}$ has the same behavior as $\tilde{f}$ (Theorem 1.3.2), as can be observed in Fig. 1.19. Thus, in this case, a neighborhood filter has the same qualitative behavior as a Perona-Malik model, even if we cannot rewrite it exactly as such.

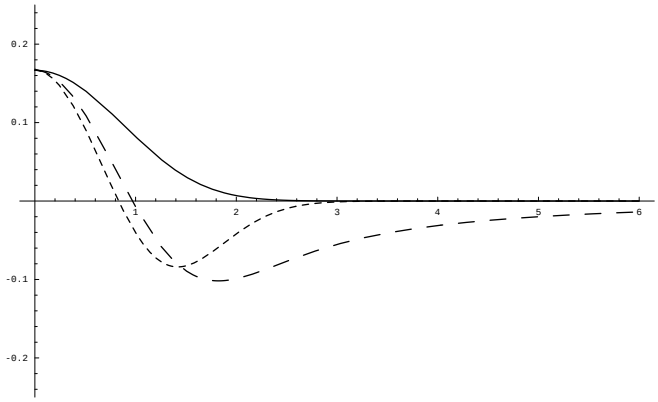

Figure 1.19: Weight comparison of the neighborhood filter and the Perona Malik equation. Magnitude of the tangent diffusion (continuous line, identical for both models) and normal diffusion (dashed line – –) of Theorem 1.3.2. Magnitude of the tangent diffusion (continuous line) and normal diffusion (dashed line - - -) of the Perona-Malik model (1.9). Both models show nearly the same behavior.

Figure 1.22 displays a comparison of the neighborhood filter and the Perona-Malik model. We display a natural image and the filtered images by both models. These solutions have a similar visual quality and tend to display flat zones

and artificial contours inside the smooth regions. Figure 1.23 corroborates this visual impression. We display the level lines of both filtered solutions. As expected from the above consistency theorems, for both models the level lines of the original image tend to concentrate, thus creating large flat zones separated by edges. The solutions are very close, up to the obvious very different implementations. The neighborhood filter is implemented exactly as in its definition and the Perona-Malik model by the explicit difference scheme proposed in the original paper.

1.3.4 A regression correction of the neighborhood filter

In the previous sections we have shown the enhancing character of the neighborhood filter. We have seen that the neighborhood filter, like the Perona-Malik model, can create large flat zones and spurious contours inside smooth regions. This effect depends upon a gradient threshold which is hard to fix in such a way as to always separate the visually smooth regions from edge regions. In order to avoid this undesirable effect, let us analyze in more detail what happens with the neighborhood filter in the one-dimensional case.

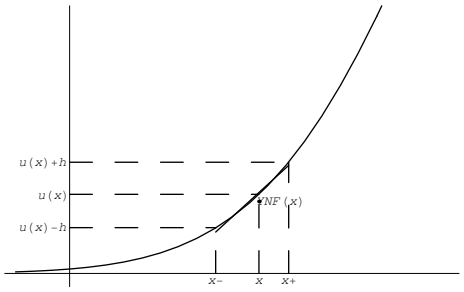

Figure 1.20: Illustration of the shock effect of the YNF on the convex of a signal. The number of points y satisfying $u(x) - h < u(y) \leq u(x)$ is larger than the number satisfying $u(x) \leq u(y) < u(x) + h$. Thus, the average value $YNF(x)$ is smaller than $u(x)$, enhancing that part of the signal. The regression line of u inside (x_-, x_+) better approximates the signal at x.

Figure 1.20 shows a simple illustration of this effect. For each x in the convex part of the signal, the filtered value is the average of the points y such that $u(x) - h < u(y) < u(x) + h$ for a certain threshold h. As it is illustrated in the figure, the number of points satisfying $u(x) - h < u(y) \leq u(x)$ is larger than the number of points satisfying $u(x) \leq u(y) < u(x)+h$. Thus, the average value $YNF(x)$ is smaller than $u(x)$, enhancing this part of the signal. A similar argument can be applied in the concave parts of the signal, dealing to the same enhancing effect. Therefore, shocks will be created inside smooth zones where concave and convex parts meet. Figure 1.20 also shows how the mean is not a good estimate of $u(x)$ in this case. In the same figure, we display the regression line approximating u inside $(u^{-1}(u(x) - h), u^{-1}(u(x) + h))$. We see how the

value of the regression line at x better approximates the signal. In the sequel, we propose to correct the neighborhood filter with this better estimate.

In the general case, this linear regression strategy amounts to finding for every point $\mathbf{x}$ the plane locally approximating u in the following sense,

$$\min_{a_0,a_1} \int_{B_\rho(\mathbf{x})} w(\mathbf{x},\mathbf{y})(u(\mathbf{y}) - a_1 y_1 - a_0)^2 \, d\mathbf{y}, \qquad w(\mathbf{x},\mathbf{y}) = e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} \quad (1.10)$$

and then replacing $u(\mathbf{x})$ by the filtered value $a_1 x_1 + a_0$. The weights used to define the minimization problem are the same as the ones used by the neighborhood filter. Thus, the points with a grey level value close to $u(x)$ will have a larger influence in the minimization process than those with a further grey level value. We denote the above linear regression correction by $LNF_{h,\rho}$. Taking $a_1 = 0$ and then approximating u by a constant function, the minimization (1.10) goes back to the neighborhood filter.

This minimization was originally proposed by Cleveland [18] with a weight family not depending on the function u but only on the spatial distance of $\mathbf{x}$ and $\mathbf{y}$. A similar scheme incorporating u in the weight computation has been statistically studied in [61]. The authors propose an iterative procedure that describes for every point the largest possible neighborhood in which the initial data can be well approximated by a parametric function.

Another similar strategy is the interpolation by ENO schemes [41]. The goal of ENO interpolation is to obtain a better adapted prediction near the singularities of the data. For each point it selects different stencils of fixed size M, and for each stencil reconstructs the associated interpolation polynomial of degree M. Then the *least oscillatory* polynomial is selected by some prescribed numerical criterion. The selected stencils tend to escape from large gradients and discontinuities.

The regression strategy also tends to select the right points in order to approximate the function. Instead of choosing a certain interval, all the points are used in the polynomial reconstruction, but weighted by the grey level differences.

As in the previous sections, let us analyze the asymptotic behavior of the linear regression correction. We compute the asymptotic expansion of the filter when $0 < \alpha \leq 1$. We showed that when $\alpha > 1$ the signal is hardly modified.

For the sake of completeness, we first compute the asymptotic expansion in the one dimensional case.

Theorem 1.3.3 *Suppose $u \in C^2(I)$, and let $\rho, h, \alpha > 0$ such that $\rho, h \to 0$ and $h = O(\rho^\alpha)$. Let $\tilde{f}$ be the continuous function defined as $\tilde{f}(0) = \frac{1}{6}$,*

$$\tilde{f}(t) = \frac{1}{4t^2}\left(1 - \frac{2t\, e^{-t^2}}{E(t)}\right),$$

for $t \neq 0$, where $E(t) = 2\int_0^t e^{-s^2}\, ds$. Then, for $x \in \mathbb{R}$,

1. *If $\alpha < 1$, $\quad LNF_{h,\rho}u(x) - u(x) \simeq \frac{u''(x)}{6}\rho^2$.*

2. If $\alpha = 1$, $\quad NF_{h,\rho}u(x) - u(x) \simeq \tilde{f}(\frac{\rho}{h}\,|u'(x)|)\,u''(x)\,\rho^2$.

Theorem 1.3.3 shows that the $LNF_{h,\rho}$ filter lets the signal evolve proportionally to its second derivative, as the neighborhood filter does. When h is larger than ρ the filter is equivalent to the original neighborhood filter and the signal is filtered by a heat equation. When ρ and h have the same order the sign and magnitude of the filtering process is given by $\tilde{f}(\frac{\rho}{h}\,|u'(x)|)$ (see Fig. 1.21). This function is positive and quickly decreases to zero. Thus, the signal is filtered by a heat equation of decreasing magnitude and is not altered wherever the derivative is very large.

The same asymptotic expansion can be computed in the two dimensional case.

Theorem 1.3.4 *Suppose* $u \in C^2(\Omega)$*, and let* $\rho, h, \alpha > 0$ *such that* $\rho, h \to 0$ *and* $h = O(\rho^\alpha)$*. Let* $\tilde{f}$ *be the continuous function defined as* $\tilde{f}(0) = \frac{1}{6}$*,*

$$\tilde{f}(t) = \frac{1}{4t^2}\left(1 - \frac{2t\,e^{-t^2}}{E(t)}\right),$$

for $t \neq 0$*, where* $E(t) = 2\int_0^t e^{-s^2}\,ds$*. Then, for* $\boldsymbol{x} \in \Omega$*,*

1. If $\alpha < 1$*,*

$$LNF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \frac{\triangle u(\boldsymbol{x})}{6}\rho^2.$$

2. If $\alpha = 1$*,*

$$LNF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \left[\tilde{f}(\frac{\rho}{h}\,|Du(\boldsymbol{x})|)\;u_{\eta\eta}(\boldsymbol{x})(\boldsymbol{x}) + \frac{1}{6}u_{\xi\xi}(\boldsymbol{x})\right]\rho^2$$

According to the previous theorem, the filter can be written as the sum of two diffusion terms in the direction of ξ and η. When h is much larger than ρ the linear regression correction is equivalent to the heat equation like the original neighborhood filter. When ρ and h have the same order, the behavior of the linear regression algorithm is very different from the original neighborhood filter. The function weighting the tangent diffusion is a positive constant. The function weighting the normal diffusion is positive and decreasing (see Fig. 1.21), and therefore there is no enhancing effect. The algorithm combines the tangent and normal diffusion wherever the gradient is small. Wherever the gradient is larger the normal diffusion is canceled and the image is filtered only in its tangent direction. This subjacent PDE was already proposed as a diffusion equation in [4]. This diffusion makes the level lines evolve proportionally to their curvature. In the Perona-Malik model the diffusion is stopped near the edges. In this case, the edges are filtered by a mean curvature motion.

It may be asked whether the modified neighborhood filter still preserves signal discontinuities. The answer is yes. It is easily checked that for small enough h, all piecewise affine functions with smooth jump curves are steady.

Thus, the behavior is the same as for the classical neighborhood filter. Our asymptotic analysis is of course not valid for such functions, but only for smooth functions.

As a numerical scheme the linear regression neighborhood filter allows the implementation of a mean curvature motion without the computation of gradients and orientations. When the gradient is small the linear regression filter naturally behaves like the heat equation. This effect is introduced on typical schemes implementing the mean curvature motion. In flat zones the gradient is not well defined and some kind of isotropic diffusion must be applied. Therefore, the linear regression neighborhood filter naturally extends the mean curvature motion and yields a stable numerical scheme for its computation, independent of gradient orientations.

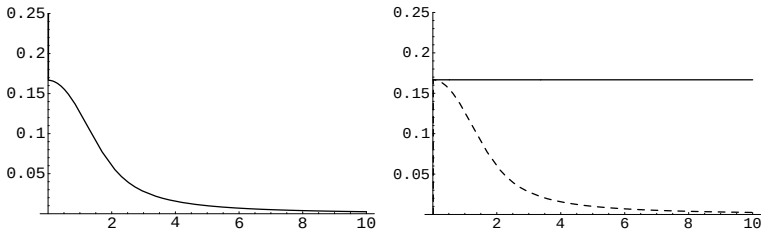

Figure 1.21: Weighting functions of Theorem 1.3.3 and 1.3.4. Left: Function $\tilde{f}$ of Theorem 1.3.3. Right: Constant function 1/6 (continuous line) and function $\tilde{f}$ (dashed line) of Theorem 1.3.4.

Figure 1.22 displays an experiment comparing the $LNF_{h,\rho}$ with the neighborhood filter and the Perona-Malik equation. The linear correction does not create any contour or flat zone inside the smooth regions. Figure 1.23 displays the level lines of the previous experiment. The level lines of the $LNF_{h,\rho}$ are filtered by a mean curvature motion and they do not get grouped creating flat zones.

1.3.5 The vector valued case

Let u be a vector valued function defined on a bounded domain $\Omega \subset \mathbb{R}^2$, $u : \Omega \to \mathbb{R}^n$. The vector neighborhood filter can be written as

$$NF_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{B_\rho(\mathbf{x})} u(\mathbf{y}) e^{-\frac{||u(\mathbf{y})-u(\mathbf{x})||^2}{h^2}} d\mathbf{y}, \tag{1.11}$$

where $||u(\mathbf{y}) - u(\mathbf{x})||^2$ is now the Euclidean vector norm and each component function u_i is filtered with the same weight distribution. The linear regression correction is defined as in the scalar case, and each component is locally approximated by a plane with the same weight distribution.

In order to compute the asymptotic expansion of the linear regression filter we must fix a coordinate system for $\mathbb{R}^2$. In the scalar case we used the reference

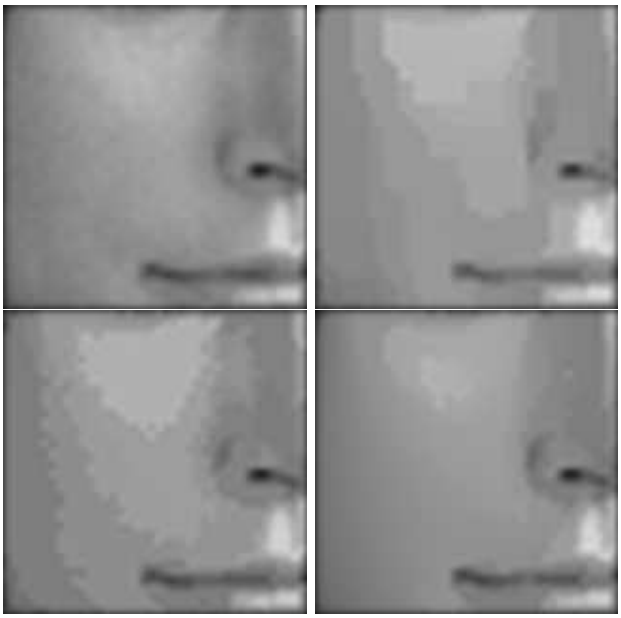

Figure 1.22: Comparison experiment. Top left: original image. Top right: Perona-Malik filtered image. Bottom left: filtered image by the neighborhood filter. Bottom right: filtered image by the linear regression neighborhood filter. The neighborhood filter experiments are performed by iterating the discrete version of definitions (1.1) and (1.10). Both the neighborhood filter and its linear regression correction have been applied with the same value of h and ρ. The displayed images have been attained within the same number of iterations. The Perona-Malik equation is implemented by the explicit difference scheme proposed in the original paper. The Perona-Malik model and the neighborhood filter create artificial contours and flat zones. This effect is almost completely avoided by the linear regression neighborhood filter.

system given by the gradient of the image at $\mathbf{x}$ and its orthogonal direction. In addition, this reference allows us to relate the obtained diffusion to the evolution of the level lines of the image and the mean curvature motion. Now, we cannot use the same reference and we need to define a new one. By analogy with the scalar case we choose the directions of minimum and maximum variation of the vector function.

Definition 1.3.1 *We define the normal direction η and the tangent direction ξ as the vectors that respectively maximize and minimize the following variation*

$$\sum_{i=1}^{n} \| \frac{\partial u_i}{\partial v}(\boldsymbol{x}) \|^2$$

under the constraint $\|v\| = 1$.

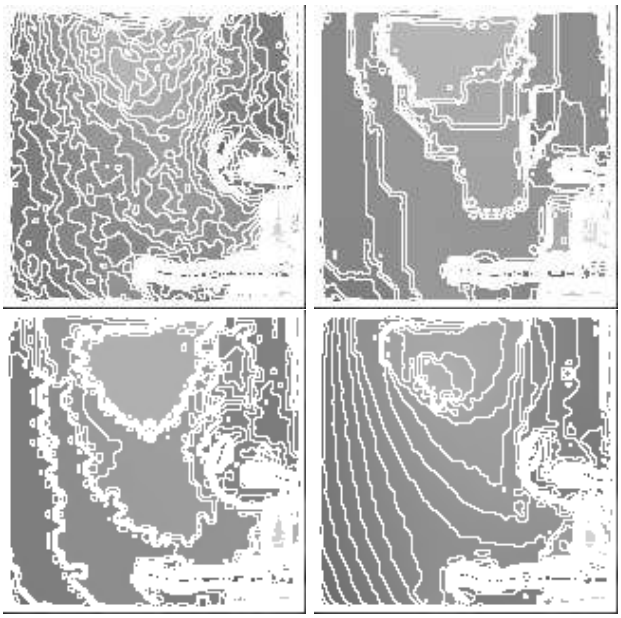

Figure 1.23: Level lines of the images in Fig. 1.22. By the Perona-Malik filter and the neighborhood filter the level lines tend to group, creating flat zones. The regression correction filters the level lines by a curvature motion without creating any flat zone.

It is easily seen that this constrained optimization leads to the computation of the eigenvectors of the matrix

$$A = \begin{pmatrix} \|\frac{\partial u}{\partial x}\|^2 & \langle \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \rangle \\ \langle \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \rangle & \|\frac{\partial u}{\partial y}\|^2 \end{pmatrix},$$

where $\frac{\partial u}{\partial x} = \left(\frac{\partial u_1}{\partial x}, \ldots, \frac{\partial u_n}{\partial x}\right)$ and $\frac{\partial u}{\partial y} = \left(\frac{\partial u_1}{\partial y}, \ldots, \frac{\partial u_n}{\partial y}\right)$. The two positive eigenvalues of A, λ_+ and λ_-, are the maximum and the minimum of the vector norm associated to A and the maximum and the minimum variations as defined in Definition 1.3.1. The corresponding eigenvectors are orthogonal leading to the above defined normal and tangent directions. This orthonormal system was first proposed for vector valued image analysis in [25]. Many PDE equations have been proposed for color image filtering using this system. We note the Coherence Enhancing Diffusion [73], the Beltrami Flow [46] and an extension of the mean curvature motion [66].

Theorem 1.3.5 *Suppose $u \in C^2(\Omega, \mathbb{R}^n)$, and let $\rho, h, \alpha > 0$ such that $\rho, h \to 0$ and $h = O(\rho^\alpha)$. Let $\tilde{f}$ be the continuous function defined as $\tilde{f}(0) = \frac{1}{6}$,*

$$\tilde{f}(t) = \frac{1}{4t^2}\left(1 - \frac{2t\, e^{-t^2}}{E(t)}\right),$$

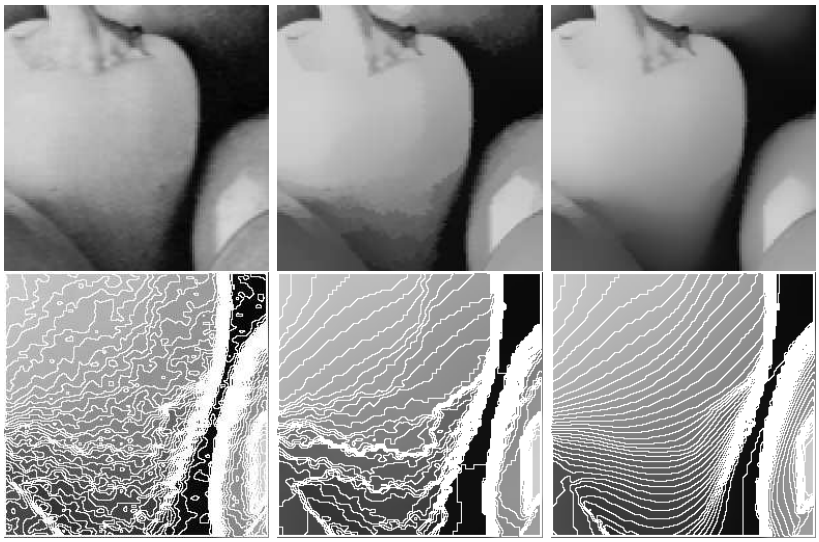

Figure 1.24: Comparison of the neighborhood filter and the linear regression correction. Top left: original image. Top middle: filtered image by the neighborhood filter. Top right: filtered image by the regression neighborhood filter. Bottom: level lines of a part of the images on the above line. Both neighborhood filters have been performed with the same filtering parameters and the same number of iterations. The linear regression neighborhood algorithm has filtered the image while preserving the main boundaries as the original neighborhood filter. No enhancing has been applied by the linear correction avoiding the shock effect. The level lines of the neighborhood filter tend to group and create large flat zones. In addition, these level lines oscillate while those of the linear regression algorithm have been correctly filtered.

for $t \neq 0$, *where* $E(t) = 2\int_0^t e^{-s^2}\ ds$. *Then, for* $\boldsymbol{x} \in \Omega$,

1. *If* $\alpha < 1$,
$$LNF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \frac{\triangle u(\boldsymbol{x})}{6}\rho^2.$$

2. *If* $\alpha = 1$,
$$LNF_{h,\rho}u(\boldsymbol{x}) - u(\boldsymbol{x}) \simeq \left[\tilde{f}(\frac{\rho}{h}\|\frac{\partial u}{\partial \xi}(\boldsymbol{x})\|)\ D^2u(\xi,\xi)(\boldsymbol{x}) + \tilde{f}(\frac{\rho}{h}\|\frac{\partial u}{\partial \eta}(\boldsymbol{x})\|)D^2u(\eta,\eta)(\boldsymbol{x})\right]\rho^2$$

where $\Delta u(\boldsymbol{x}) = (\Delta u_i(\boldsymbol{x}))_{1\leq i\leq n}$ *and* $D^2u(v,v)(\boldsymbol{x}) = \left(D^2u_i(v,v)(\boldsymbol{x})\right)_{1\leq i\leq n}$ *for* $v \in \{\eta, \xi\}$.

Interpretation
When h is much larger than ρ, the linear regression neighborhood filter is equivalent to the heat equation applied independently to each component. When h

and ρ have the same order the subjacent PDE acts as an evolution equation with two terms. The first term is proportional to the second derivative of u in the tangent direction ξ. The second term is proportional to the second derivative of u in the normal direction η. The magnitude of each diffusion term depends on the variation in the respective direction, $\lambda_- = \|\frac{\partial u}{\partial \xi}(\mathbf{x})\|$ and $\lambda_+ = \|\frac{\partial u}{\partial \eta}(\mathbf{x})\|$. The weighting function $\tilde{f}$ is positive and decreases to zero (see Fig. 1.21). We can distinguish the following cases depending on the values of λ_+ and λ_-.

- If $\lambda_+ \simeq \lambda_- \simeq 0$ then there are very few variations of the vector image u around $\mathbf{x}$. In this case, the linear regression neighborhood filter behaves like a heat equation with maximum diffusion coefficient $\tilde{f}(0)$.
- If $\lambda_+ >> \lambda_-$ then there are strong variations of u around $\mathbf{x}$ and the point may be located on an edge. In this case the magnitude $\tilde{f}(\frac{\rho}{h}\lambda_+)$ tends to zero and there is no diffusion in the direction of maximal variation. If $\lambda_- >> 0$ then $\mathbf{x}$ may be placed on an edge with different orientations depending on each component and the magnitude of the filtering in both directions tends to zero, so that the image is hardly altered. If $\lambda_- \simeq 0$ then the edges have similar orientations in all the components and the image is filtered by a directional Laplacian in the direction of minimal variation.
- If $\lambda_+ \simeq \lambda_- >> 0$ then we may be located on a saddle point and in this case the image is hardly modified. When dealing with multi-valued images one can think of the complementarity of the different channels leading to the perception of a corner.

In the scalar case the theorem gives back the result studied in the previous sections. The normal and tangent directions are respectively the gradient direction and the level line direction. In this case, $\frac{\partial u}{\partial \xi}(\mathbf{x}) = 0$ and $\frac{\partial u}{\partial \eta}(\mathbf{x}) = |Du(\mathbf{x})|$ and we get back to

$$LNF_{h,\rho}u(\mathbf{x}) - u(\mathbf{x}) \simeq \left[\frac{1}{6}D^2u(\xi,\xi)(\mathbf{x}) + \tilde{f}(\frac{\rho}{h}|Du(\mathbf{x})|)D^2u(\eta,\eta)(\mathbf{x})\right]\rho^2.$$

1.4 Variational and linear diffusion

The relationship of neighborhood filters to classic local PDE's has been discussed in previous section. Yet, the main interest has shifted to defining *non-local PDE's*. The extension of the neighborhood filter and the NL-means method to define nonlocal image-adapted differential operators and non-local variational methods starts with [47], which proposes to perform denoising and deblurring by non-local functionals.

The general goal of this development is actually to give a variational to all neighborhood filters, and to give a non local form to the total variation as well. More precisely, the neighborhood filters derive from the functional

$$J(u) = \int_{\Omega\times\Omega} g(\frac{|u(x)-u(y)|^2}{h^2})w(|x-y|)dxdy,$$

where g and w have a Gaussian decay. In the same line, a functional yields a (variational) interpretation to NL-means:

$$JNL(u) = \int_{\Omega\times\Omega} \left(1 - e^{-\frac{G_\sigma * |u(x-.)-u(y-.)|^2(0)}{h^2}}\right) w(|x-y|)dxdy.$$

In a similar variational framework, Gilboa et al. [36] consider the general kind of quadratic non local functional

$$J(u) := \int_{\Omega\times\Omega} (u(\mathbf{x}) - u(\mathbf{y}))^2 w(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y}, \tag{1.12}$$

where $w(\mathbf{x},\mathbf{y})$ is any fixed weight distribution, which in most application writes as the neighborhood or NL-means weight distribution. The resolution of the graph heat equation or the variational minimization (1.12) is given by

$$u_{n+1}(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_\Omega u_n(\mathbf{y}) w(\mathbf{x},\mathbf{y})d\mathbf{y}, \tag{1.13}$$

where $C(\mathbf{x}) = \int_\Omega w(\mathbf{x},\mathbf{y})d\mathbf{y}$ is a normalizing factor. The freedom having a totally decoupled weight distribution makes this formulation a linear and powerful tool for image processing. In fact, this formulation rewrites as the Dirichlet integral of the following non local gradient, $\nabla_w u(x,y) = (u(\mathbf{x}) - u(\mathbf{y}))w(x,y)$. The whole process relates to a graph Laplacian where each pixel is considered as the node of a weighted graph and the weights of the edge between two pixels x and y respectively are decreasing functions of the distances of patches around x and y, $w(x,y)$. Then a graph Laplacian can be calculated on this graph, seen as the sampling of a manifold, and the linear diffusion can be interpreted as the heat equation on the set of blocks endowed with these weights. The eigenvalues and eigenvectors of such a Laplacian can be computed and used for designing spectral algorithms as Wiener and thresholding methods, see [70] and [59].

The non local term (1.12) has shown to be very useful as a regularization term for many image processing tasks. The non-local differential operators permit to define a total variation or a Dirichlet integral. Several articles on deblurring have followed this variational line [44], [53], [36] (for image segmentation), [8] (in fluorescence microscopy), [82], again for nonlocal deconvolution and [50] for deconvolution and tomographic reconstruction. In [33], a paper dedicated to another notoriously ill-posed problem, the super-resolution, the non-local variational principle is viewed as "an emerging powerful family of regularization techniques", and the paper "proposes to use the example-based approach as a new regularising principle in ill-posed image processing problems such as image super-resolution from several low resolution photographs." For all these methods, the weight distribution is computed in the first iteration and are maintained during the whole iteration process.

In this section, we will concentrate in the last non local functional as a linear diffusion process and therefore as a heat equation is the associated graph to the image, that is, no fidelity term will be added to the functional.

1.4.1 Linear diffusion: seed growing

In [37], [39], a novel method was proposed for performing multi-label, semi-automated medical image segmentation. The Grady segmentation method is a linearised sigma filter applied to propagate seed regions.

Given a small number of pixels with user-defined labels which are called seeds, this method computes the probability that a random walker starting at each unlabeled pixel will first reach one of the pre-labeled pixels. By assigning each pixel to the label for which the greatest probability is calculated, a high-quality image segmentation can be obtained.

With each unlabeled pixel, a K-tuple vector is assigned that represents the probability that a random walker starting from this unlabeled pixel first reaches each of the K seed points. A final segmentation may be derived from these K-tuples by selecting for each pixel the most probable seed destination for random walker. By biasing the random walker to avoid crossing sharp intensity gradients, a quality segmentation is obtained that respects object boundaries (including weak boundaries). The image (or volume) is treated as a graph with a fixed number of vertices and edges. Each edge is assigned real-valued weight corresponding to the likelihood that a random walker will cross that edge (e.g., a weight of zero means that the walker may not move along that edge). By a classical result the probability that a random walker first reaches a seed point exactly equals the solution to the heat equation [9] with boundary Dirichlet conditions at the locations of the seed points, the seed point in question being fixed to unity, while the others seeds are set to zero.

This idea was not quite new. Region competition segmentation is an old concept [83]. One can also refer to an algorithm developed for machine learning by Zhu et. al [84] which also finds clusters based upon harmonic functions, using boundary conditions set by a few seed points. [68] also involves weights in the image considered as a graph and takes seed points. The method is also directly related to the Sapiro et al. recent image coloring method by diffusion from seeds [79] (see also [65]).

Thus, the Grady segmentation method is a linearized sigma filter applied to propagate seed regions. Figure 1.25 taken from [38] illustrates the process on a two chamber view of a cardiac image. The grey curves are user defined seed regions roughly denoting the ventricles in the image. In that case one of the seed regions is put to 1 and the other to zero. A diffusion with sigma filter weights computed on the original image u_0 is applied until a steady state is attained. This gives at each pixel $\mathbf{y}$ a value $p_1(\mathbf{y})$ between 0 and 1 which is interpreted as the probability for $\mathbf{y}$ to belong to the region of the first seed. In this binary case a single threshold at 0.5 gives the black curves separating the regions of both seeds. Like the active contour method, this method is highly dependent on the initial seeds. It is, however, much less sensitive to noise than the snakes method [16] and permits to initialize fairly far from the desired contours. We will see that by the histogram concentration phenomenon one can get similar or better results without any initialization.

The very same process as illustrated allows to diffuse initial chromatic in-

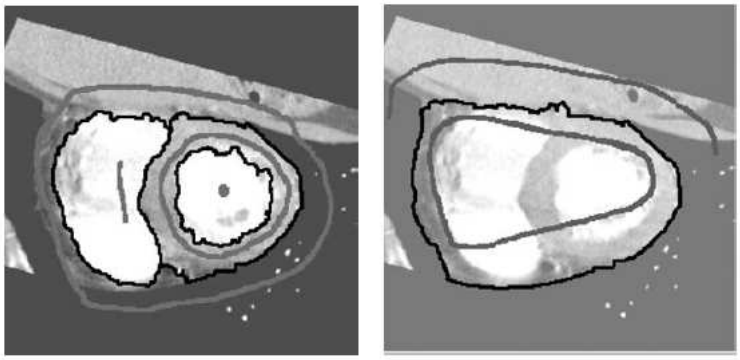

Figure 1.25: (Taken from [38].) The Grady segmentation method is a linearized sigma filter applied to propagate seed regions. The grey curves are user defined seed region. A diffusion with sigma filter weights computed on the original image u_0 is applied until a steady state is attained. A threshold gives the black curves separating the regions of initial seeds.

formation on an initial grey image as we exposed in the introduction. Figure 1.26 illustrates this application and compares the obtained solution by using the NL-means and the neighborhood filter.

1.4.2 Linear diffusion: histogram concentration

The segmentation process can be accomplished by iterating the neighborhood filter and computing the weight distribution in the initial image as displayed in Fig. 1.27. The top image shows one slice of a 3D CT image with interest area surrounded by a parallelepiped. The next row shows several slices of this area of interest. It can be appreciated, first that the background of arteries has a lot of oscillating clutter and, second, that the grey level value in arteries varies a lot, thus making an automatic threshold problematic. The best way actually to convince oneself that even in this small area a direct threshold wouldn't do the job is to refer to the histograms of Fig. 1.29. The first histogram which is Gaussian-like and poorly concentrated corresponds to the background. The background mode decreases slowly. On the far right part of the histogram one can see a small pick corresponding to very white arteries. The fixing of an accurate threshold in the slowly decreasing background mode is problematic. The top right histogram shows what happens after the application of a median iterative filtering (the mean curvature motion). The histogram does not concentrate at all. The bottom left histogram is obtained after applying the linearized neighborhood filter. The bottom right histogram the one obtained by the linearized NL-means described in the same section. In both cases, one observes that the background mode of the histogram is strongly concentrated on a few grey level values. An automatic threshold is easily fixed by taking the first local minimum after the main histogram peak. This histogram concentration is very

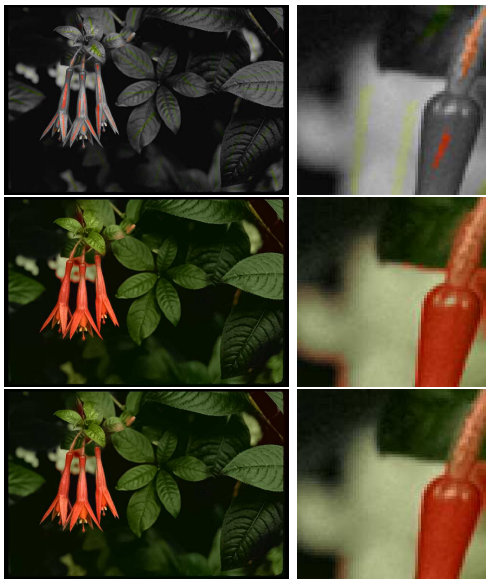

Figure 1.26: Left and from top to bottom: initial chromatic data on the gray image, linear diffused seeds by using neighborhood filter weights on the gray image and the same for the NL-means weights. Right: details of left hand images. The neighborhood filter weights are not robust since just a single points from different objects can be easily confused and iteration lead to a incorrect colorization.

similar to the obtained by the mean-shift approach [20] where the neighborhood filter is nonlinearly iterated. In that case, the authors show that clusters tend to its mean yielding piecewise constant image.

The histogram concentration phenomenon is actually visible in the comparative evolution of some slices under the various considered filters, as shown in Fig. 1.27. The first row shows these slices picked in the interest area. The topology killing effect of the median filter (mean curvature motion): small arteries tend to vanish and larger ones shrink and become circular as shown in the third slice showing an artery section. The third row is dedicated to the linear sigma filter, which corresponds to Grady's method applied directly to the image instead of using seeds. It is quite apparent that well contrasted objects are well maintained and the contrast augmented, in agreement with the consistency of this recursive filter with the Perona-Malik equation. However, the less contrasted objects tend to vanish because, on them, the evolution becomes similar to an isotropic heat equation. The fourth row is the result of applying the 3D non-local linear heat equation, where the Laplacian coefficients are computed from the original image. The whole sequence has been treated as a 3D image with a weight support of $(7 \times 7 \times 3)$ and a comparison window of $3 \times 3 \times 3$. Clearly the background is flattened and blood vessels are enhanced

on this background. A threshold just above the homogeneous made background level should give back arteries, and this indeed occurs. Thus in that case the 3d visualization of objects with complex topology like the cerebral arteries can be achieved by an automatic threshold. The exact segmentation of the artery is a more difficult problem. Even if the histogram is concentrated, a different choice of the visualization threshold can produce slightly different surfaces.

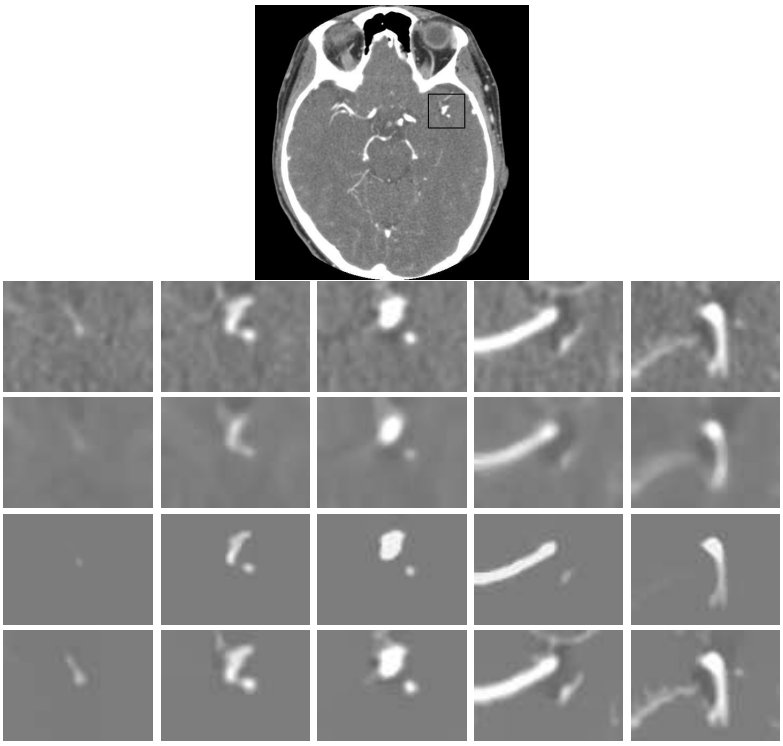

Figure 1.27: Comparative behavior of discussed methods in 3D. Application to a 3D angiograhy CT image of the head where blood vessels should be segmented. Top: One slice image of the CT volume data with marked interested area. Middle: Display of interest area for several slices of the 3D image. Second row: filtered slices by using median filter. Third row: sigma filter. Fourth row: 3D nonlocal heat equation. Bottom: filtered slices by using the linear method with 3D NL-means weights. The whole sequence has been treated as a 3D image with a weight support of $(5\times5\times3)$ and a comparison window of $3\times3\times3$. The background is flattened and blood vessels are enhanced. Thus, a better segmentation is possible by a simple threshold as justified by Fig. 1.29.

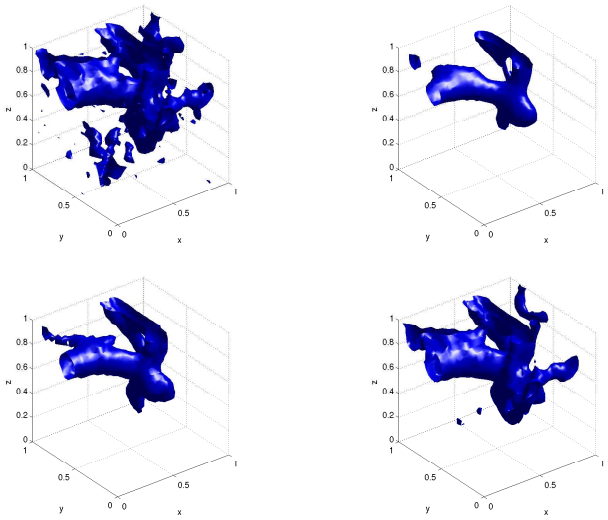

Figure 1.28: From top to bottom and left to right: original iso-surface of the 3D image, same iso-surface filtered by iterative median filter, by linear sigma filter, and by linear NL-means. The iso-surface extracted from the original image presents many irregularities due to noise. The median filter makes them disappear but makes important parts disappear and some vessels disconnect or fuse. Linear NL-means keeps most vessels and maintains the topology.

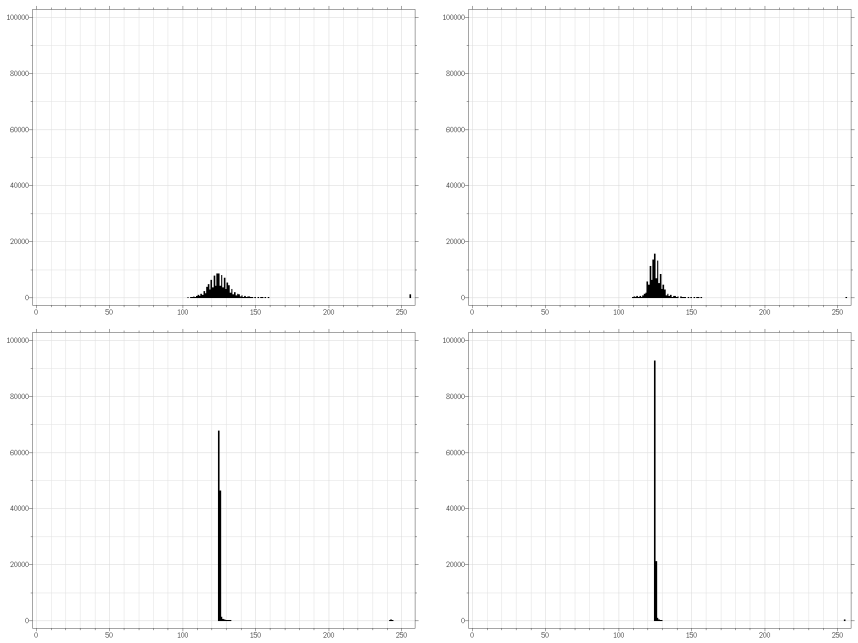

Figure 1.29: Grey level histogram of 3D areas of interest. Top left: original 3D image before. Top right: after median filtering. Bottom left: after proposed method with sigma filter weights. Bottom right; proposed method with NL-means weights The background is now represented by a few grey level values when the volume is filtered by the proposed method. A threshold can therefore be more easily and automatically applied.

Bibliography

[1] F. Andreu, C. Ballester, V. Caselles, and J.M. Mazon. Minimizing total variation flow. *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, 331(11):867–872, 2000.

[2] P. Arias, V. Caselles, and G. Sappiro. A variational framework for non-local image inpainting. *Proc. of EMMCVPR. Springer, Heidelberg*, 2009.

[3] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183–193, 1954.

[4] G. Aubert and P. Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer-Verlag New York Inc, 2006.

[5] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics (TOG)*, 25(3):645, 2006.

[6] D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 844–847, 2002.

[7] E.P. Bennett, J.L. Mason, and L. McMillan. Multispectral bilateral video fusion. *IEEE Transactions on Image Processing*, 16(5):1185, 2007.

[8] J. Boulanger, J.B. Sibarita, C. Kervrann, and P. Bouthemy. Non-parametric regression for patch-based fluorescence microscopy image sequence denoising. In *5th IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008*, pages 748–751, 2008.

[9] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in NDimages. *International Conference on Computer Vision*, 1:105–112, 2001.

[10] JC Brailean, RP Kleihorst, S. Efstratiadis, AK Katsaggelos, and RL Lagendijk. Noise reduction filters for dynamic image sequences: a review. *Proceedings of the IEEE*, 83(9):1272–1292, 1995.

[11] A. Buades, B. Coll, J. Lisani, and C. Sbert. Conditional image diffusion. *Inverse Problems and Imaging*, 1(4):593, 2007.

[12] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling Simulation*, 4(2):490–530, 2005.

[13] A. Buades, B. Coll, and J.M. Morel. Neighborhood filters and PDE's. *Numerische Mathematik*, 105(1):1–34, 2006.

[14] A. Buades, B. Coll, and J.M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[15] A. Buades, B. Coll, JM Morel, and C. Sbert. Self-Similarity Driven Color Demosaicking. *IEEE Transactions on Image Processing*, 18(6):1192–1202, 2009.

[16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[17] P. Choudhury and J. Tumblin. The trilateral filter for high contrast images and meshes. In *ACM SIGGRAPH 2005 Courses*, page 5. ACM, 2005.

[18] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[19] R.R. Coifman and D.L. Donoho. Translation-invariant de-noising. *Lecture Notes In Statistics*, pages 125–125, 1995.

[20] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[21] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9), 2004.

[22] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian. Image And Video Super-Resolution Via Spatially Adaptive Block-Matching Filtering. In *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[23] J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, page 368. ACM Press/Addison-Wesley Publishing Co., 1997.

[24] J. Delon and A. Desolneux. Flicker stabilization in image sequences. *hal.archives-ouvertes.fr*, 2009.

[25] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, 1986.

[26] B. Dong, J. Ye, S. Osher, and I. Dinov. Level set based nonlocal surface restoration. *Multiscale Modeling & Simulation*, 7:589, 2008.

[27] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[28] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 257–266. ACM New York, NY, USA, 2002.

[29] M. Ebrahimi and E.R. Vrscay. Solving the Inverse Problem of Image Zooming Using" Self-Examples". *Lecture Notes in Computer Science*, 4633:117, 2007.

[30] M. Ebrahimi and E.R. Vrscay. Multi-frame super-resolution with no explicit motion estimation. In *Proceedings of the 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2008.

[31] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 2, pages 1033–1038. Corful, Greece, 1999.

[32] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics (TOG)*, 23(3):673–678, 2004.

[33] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal*, 2007.

[34] A. Elmoataz, O. Lezoray, S. Bougleux, and V.T. Ta. Unifying local and nonlocal processing with partial difference operators on weighted graphs. In *International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[35] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics (TOG)*, 22(3):950–953, 2003.

[36] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling Simulation*, 6(2):595–630, 2007.

[37] L. Grady. Random Walks for Image Segmentation. *IEEE Trans. on Pattern Anal. and Mach. Intel.*, 28(11):1, 2006.

[38] L. Grady and G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, ECCV*, pages 230–245, 2004.

[39] L.J. Grady. *Space-variant Computer Vision: A Graph-theoretic Approach*. PhD thesis, Boston University, 2004.

[40] F. Guichard, J.M. Morel, and R. Ryan. Contrast invariant image analysis and PDEs. *Book in preparation.*

[41] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71(2):231–303, 1987.

[42] B. Huhle, T. Schairer, P. Jenke, and W. Straßer. Robust non-local denoising of colored depth data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2008.

[43] T.R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3):943–949, 2003.

[44] M. Jung and LA Vese. Nonlocal variational image deblurring models in the presence of Gaussian or impulse noise, 2009.

[45] B.B. Kimia, A. Tannenbaum, and S.W. Zucker. On the evolution of curves via a function of curvature, I: the classical case. *Journal of Mathematical Analysis and Applications*, 163(2):438–458, 1992.

[46] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, 2000.

[47] S. Kindermann, S. Osher, and P.W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling and Simulation*, 4(4):1091–1115, 2005.

[48] J.S. Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269, 1983.

[49] O. Lezoray, V.T. Ta, and A. Elmoataz. Nonlocal graph regularization for image colorization. *International Conference on Pattern Recognition*, 2008.

[50] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi. Image recovery via nonlocal operators. *Journal of Scientific Computing*, 42(2):185–197, 2010.

[51] J. Mairal, M. Elad, G. Sapiro, et al. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53, 2008.

[52] S. Masnou. *Filtrage et désocclusion d'images par méthodes d'ensembles de niveau.* PhD thesis.

[53] M. Mignotte. A non-local regularization strategy for image deconvolution. *Pattern Recognition Letters*, 29(16):2206–2212, 2008.

[54] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

[55] M.K. Ozkan, M.I. Sezan, and A.M. Tekalp. Adaptive motion-compensated filtering of noisy image sequences. *IEEE transactions on circuits and systems for video technology*, 3(4):277–290, 1993.

[56] H. Peng, R. Rao, and D.W. Messinger. Spatio-spectral bilateral filters for hyperspectral imaging.

[57] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. PAMI*, 12(7):629–639, 1990.

[58] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics (TOG)*, 23(3):664–672, 2004.

[59] G. Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009.

[60] G. Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, 2009.

[61] J. Polzehl and V. Spokoiny. Varying coefficient regression modeling by adaptive weights smoothing. *Preprint*, 818, 2002.

[62] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Transactions on Image Processing*, 18(1):35–51, 2009.

[63] R. Ramanath and W.E. Snyder. Adaptive demosaicking. *Journal of Electronic Imaging*, 12:633, 2003.

[64] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[65] G. Sapiro and DL Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *Image Processing, IEEE Transactions on*, 5(11):1582–1586, 1996.

[66] G. Sapiro and DL Ringach. Anisotropic diffusion of multivalued images with applications tocolor filtering. *IEEE Transactions on Image Processing*, 5(11):1582–1586, 1996.

[67] JA Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, 1985.

[68] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[69] S.M. Smith and J.M. Brady. SUSAN: A new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[70] A.D. Szlam, M. Maggioni, and R.R. Coifman. A general framework for adaptive regularization based on diffusion processes on graphs. *Yale technichal report*, 2006.

[71] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, 1998.

[72] R. van den Boomgaard and J. van de Weijer. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In *International Conference on Pattern Recognition*, volume 16, pages 927–930. Citeseer, 2002.

[73] J. Weickert. *Anisotropic diffusion in image processing*. Citeseer, 1998.

[74] H. Winnemoller, S.C. Olsen, and B. Gooch. Real-time video abstraction. *ACM Transactions on Graphics (TOG)*, 25(3):1226, 2006.

[75] A. Wong and J. Orchard. A nonlocal-means approach to exemplar-based inpainting. In *15th IEEE International Conference on Image Processing, 2008*, pages 2600–2603, 2008.

[76] L. P. Yaroslavsky. *Digital Picture Processing*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1985.

[77] L.P. Yaroslavsky. Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window. In *Proceedings of SPIE*, volume 2825, page 2, 1996.

[78] L.P. Yaroslavsky, K.O. Egiazarian, and J.T. Astola. Transform domain image restoration methods: review, comparison, and interpretation. In *Proc. of SPIE*, volume 4304, page 155, 2001.

[79] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *Image Processing, IEEE Transactions on*, 15(5):1120–1129, 2006.

[80] S. Yoshizawa, A. Belyaev, and H.P. Seidel. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In *IEEE International Conference on Shape Modeling and Applications*, pages 9–9, 2006.

[81] D. Zhang and Z. Wang. Image information restoration based on long-range correlation. *IEEE Transactions on circuits and systems for video technology*, 12(5):331–341, 2002.

[82] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *UCLA CAM Report*, pages 09–03, 2009.

[83] S.C. Zhu and A. Yuille. Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

[84] X. Zhu, J. Lafferty, and Z. Ghahramani. *Semi-supervised Learning: From Gaussian Fields to Gaussian Processes*. School of Computer Science, Carnegie Mellon University, 2003.