A non-local algorithm for image denoising

Antoni Buades, Bartomeu Coll
Dpt. Matemàtiques i Informàtica, UIB
Ctra. Valldemossa Km. 7.5,
07122 Palma de Mallorca, Spain
vdmiabc4@uib.es, tomeu.coll@uib.es

Jean-Michel Morel
CMLA, ENS Cachan
61, Av du Président Wilson
94235 Cachan, France
morel@cmla.ens-cachan.fr

Abstract

We propose a new measure, the method noise, *to evaluate and compare the performance of digital image denoising methods. We first compute and analyze this method noise for a wide class of denoising algorithms, namely the local smoothing filters. Second, we propose a new algorithm, the* non local means (NL-means)*, based on a non local averaging of all pixels in the image. Finally, we present some experiments comparing the NL-means algorithm and the local smoothing filters.*

1. Introduction

The goal of image denoising methods is to recover the original image from a noisy measurement,

$$v(i) = u(i) + n(i), \qquad (1)$$

where $v(i)$ is the observed value, $u(i)$ is the "true" value and $n(i)$ is the noise perturbation at a pixel i. The best simple way to model the effect of noise on a digital image is to add a gaussian white noise. In that case, $n(i)$ are i.i.d. gaussian values with zero mean and variance σ^2.

Several methods have been proposed to remove the noise and recover the true image u. Even though they may be very different in tools it must be emphasized that a wide class share the same basic remark : denoising is achieved by averaging. This averaging may be performed locally: the Gaussian smoothing model (Gabor [7]), the anisotropic filtering (Perona-Malik [11], Alvarez et al. [1]) and the neighborhood filtering (Yaroslavsky [16], Smith et al. [14], Tomasi et al. [15]), by the calculus of variations: the Total Variation minimization (Rudin-Osher-Fatemi [13]), or in the frequency domain: the empirical Wiener filters (Yaroslavsky [16]) and wavelet thresholding methods (Coiffman-Donoho [5, 4]).

Formally we define a denoising method D_h as a decomposition

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

where v is the noisy image and h is a filtering parameter which usually depends on the standard deviation of the noise. Ideally, $D_h v$ is smoother than v and $n(D_h, v)$ looks like the realization of a white noise. The decomposition of an image between a smooth part and a non smooth or oscillatory part is a current subject of research (for example Osher et al. [10]). In [8], Y. Meyer studied the suitable functional spaces for this decomposition. The primary scope of this latter study is not denoising since the oscillatory part contains both noise and texture.

The denoising methods should not alter the original image u. Now, most denoising methods degrade or remove the fine details and texture of u. In order to better understand this removal, we shall introduce and analyze the *method noise*. The method noise is defined as the difference between the original (always slightly noisy) image u and its denoised version.

We also propose and analyze the *NL-means* algorithm, which is defined by the simple formula

$$NL[u](x) = \frac{1}{C(x)} \int_\Omega e^{-\frac{(G_a * |u(x+.) - u(y+.)|^2)(0)}{h^2}} u(y) \, dy,$$

where $x \in \Omega$, $C(x) = \int_\Omega e^{-\frac{(G_a * |u(x+.) - u(z+.)|^2)(0)}{h^2}} dz$ is a normalizing constant, G_a is a Gaussian kernel and h acts as a filtering parameter. This formula amounts to say that the denoised value at x is a mean of the values of all points whose gaussian neighborhood looks like the neighborhood of x. The main difference of the NL-means algorithm with respect to local filters or frequency domain filters is the systematic use of all possible self-predictions the image can provide, in the spirit of [6]. For a more detailed analysis on the NL-means algorithm and a more complete comparison, see [2].

Section 2 introduces the method noise and computes its mathematical formulation for the mentioned local smooth-

ing filters. Section 3 gives a discrete definition of the NL-means algorithm. In section 4 we give a theoretical result on the consistency of the method. Finally, in section 5 we compare the performance of the NL-means algorithm and the local smoothing filters.

2. Method noise

Definition 1 (Method noise) *Let u be an image and D_h a denoising operator depending on a filtering parameter h. Then, we define the* method noise *as the image difference*

$$u - D_h u.$$

The application of a denoising algorithm should not alter the non noisy images. So the method noise should be very small when some kind of regularity for the image is assumed. If a denoising method performs well, the method noise must look like a noise even with non noisy images and should contain as little structure as possible. Since even good quality images have some noise, it makes sense to evaluate any denoising method in that way, without the traditional "add noise and then remove it" trick. We shall list formulas permitting to compute and analyze the method noise for several classical local smoothing filters: the Gaussian filtering [7], the anisotropic filtering [1, 11], the Total Variation minimization [13] and the neighborhood filtering [16]. The formal analysis of the method noise for the frequency domain filters fall out of the scope of this paper. These method noises can also be computed but their interpretation depends on the particular choice of the wavelet basis.

2.1. The Gaussian filtering

The image isotropic linear filtering boils down to the convolution of the image by a linear symmetric kernel. The paradigm of such kernels is of course the gaussian kernel $\mathbf{x} \to G_h(\mathbf{x}) = \frac{1}{(4\pi h^2)} e^{-\frac{|\mathbf{x}|^2}{4h^2}}$. In that case, G_h has standard deviation h and it is easily seen that

Theorem 1 (Gabor 1960) *The image method noise of the convolution with a gaussian kernel G_h is*

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

for h small enough.

The gaussian method noise is zero in harmonic parts of the image and very large near edges or texture, where the Laplacian cannot be small. As a consequence, the Gaussian convolution is optimal in flat parts of the image but edges and texture are blurred.

2.2. The anisotropic filtering

The anisotropic filter (AF) attempts to avoid the blurring effect of the Gaussian by convolving the image u at $\mathbf{x}$ only in the direction orthogonal to $Du(\mathbf{x})$. The idea of such filter goes back to Perona and Malik [11]. It is defined by

$$AF_h u(\mathbf{x}) = \int G_h(t) u(\mathbf{x} + t\frac{Du(\mathbf{x})^\perp}{|Du(\mathbf{x})|}) dt,$$

for $\mathbf{x}$ such that $Du(\mathbf{x}) \neq 0$ and where $(x, y)^\perp = (-y, x)$ and G_h is the one-dimensional Gauss function with variance h^2. If one assumes that the original image u is twice continuously differentiable (C^2) at $\mathbf{x}$, it is easily shown by a second order Taylor expansion that

Theorem 2 *The image method noise of an anisotropic filter AF_h is*

$$u(\boldsymbol{x}) - AF_h u(\boldsymbol{x}) = -\frac{1}{2} h^2 |Du| curv(u)(\boldsymbol{x}) + o(h^2),$$

where the relation holds when $Du(\boldsymbol{x}) \neq 0$.

By $curv(u)(\mathbf{x})$, we denote the curvature, i.e. the signed inverse of the radius of curvature of the level line passing by $\mathbf{x}$. This method noise is zero wherever u behaves locally like a straight line and large in curved edges or texture (where the curvature and gradient operators take high values). As a consequence, the straight edges are well restored while flat and textured regions are degraded.

2.3. The Total Variation minimization

The Total Variation minimization was introduced by Rudin, Osher and Fatemi [13]. Given a noisy image $v(\mathbf{x})$, these authors proposed to recover the original image $u(\mathbf{x})$ as the solution of the minimization problem

$$TVF_\lambda(v) = \arg\min_u TV(u) + \lambda \int |v(\mathbf{x}) - u(\mathbf{x})|^2 d\mathbf{x}$$

where $TV(u)$ denotes the total variation of u and λ is a given Lagrange multiplier. The minimum of the above minimization problem exists and is unique. The parameter λ is related to the noise statistics and controls the degree of filtering of the obtained solution.

Theorem 3 *The method noise of the Total Variation minimization is*

$$u(\boldsymbol{x}) - TVF_\lambda(u)(\boldsymbol{x}) = -\frac{1}{2\lambda} curv(TVF_\lambda(u))(\boldsymbol{x}).$$

As in the anisotropic case, straight edges are maintained because of their small curvature. However, details and texture can be over smoothed if λ is too small.

2.4. The neighborhood filtering

We call neighborhood filter any filter which restores a pixel by taking an average of the values of neighboring pixels with a similar grey level value. Yaroslavsky (1985) [16] averages pixels with a similar grey level value and belonging to the spatial neighborhood $B_\rho(\mathbf{x})$,

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

where $\mathbf{x} \in \Omega$, $C(\mathbf{x}) = \int_{B_\rho(\mathbf{x})} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}}d\mathbf{y}$ is the normalization factor and h is a filtering parameter.

The Yaroslavsky filter is less known than more recent versions, namely the *SUSAN filter* (1995) [14] and the *Bilateral filter* (1998) [15]. Both algorithms, instead of considering a fixed spatial neighborhood $B_\rho(\mathbf{x})$, weigh the distance to the reference pixel $\mathbf{x}$,

$$SNF_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})}\int_{\Omega} u(\mathbf{y})e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\rho^2}}e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}}d\mathbf{y},$$
$$(3)$$

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 ρ is now a spatial filtering parameter. In practice, there is no difference between $YNF_{h,\rho}$ and $SNF_{h,\rho}$. If the grey level difference between two regions is larger than h, both algorithms compute averages of pixels belonging to the same region as the reference pixel. Thus, the algorithm does not blur the edges, which is its main scope. In the experimentation section we only compare the Yaroslavsky neighborhood filter.

The problem with these filters is that comparing only grey level values in a single pixel is not so robust when these values are noisy. Neighborhood filters also create artificial shocks which can be justified by the computation of its method noise, see [3].

3. NL-means algorithm

Given a discrete noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL[v](i)$, for a pixel i, is computed as a weighted average of all the pixels in the image,

$$NL[v](i) = \sum_{j \in I} w(i,j)v(j),$$

where the family of weights $\{w(i,j)\}_j$ depend on the similarity between the pixels i and j, and satisfy the usual conditions $0 \leq w(i,j) \leq 1$ and $\sum_j w(i,j) = 1$.

The similarity between two pixels i and j depends on the similarity of the intensity gray level vectors $v(\mathcal{N}_i)$ and $v(\mathcal{N}_j)$, where $\mathcal{N}_k$ denotes a square neighborhood of fixed size and centered at a pixel k. This similarity is measured
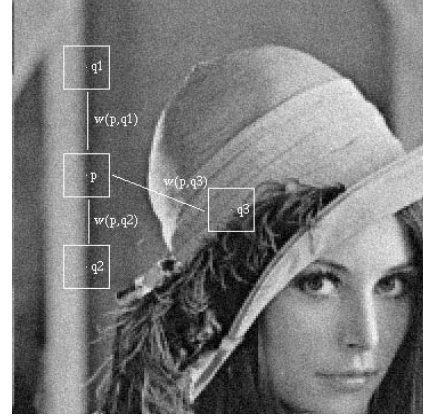

Figure 1. Scheme of NL-means strategy. Similar pixel neighborhoods give a large weight, w(p,q1) and w(p,q2), while much different neighborhoods give a small weight w(p,q3).

as a decreasing function of the weighted Euclidean distance, $||v(\mathcal{N}_i) - v(\mathcal{N}_j)||^2_{2,a}$, where $a > 0$ is the standard deviation of the Gaussian kernel. The application of the Euclidean distance to the noisy neighborhoods raises the following equality

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

This equality shows the robustness of the algorithm since in expectation the Euclidean distance conserves the order of similarity between pixels.

The pixels with a similar grey level neighborhood to $v(\mathcal{N}_i)$ have larger weights in the average, see Figure 1. These weights are defined as,

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

where $Z(i)$ is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{||v(\mathcal{N}_i)-v(\mathcal{N}_i)||^2_{2,a}}{h^2}}$$

and the parameter h acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

The NL-means not only compares the grey level in a single point but the the geometrical configuration in a whole neighborhood. This fact allows a more robust comparison than neighborhood filters. Figure 1 illustrates this fact, the pixel $q3$ has the same grey level value of pixel p, but the neighborhoods are much different and therefore the weight $w(p, q3)$ is nearly zero.
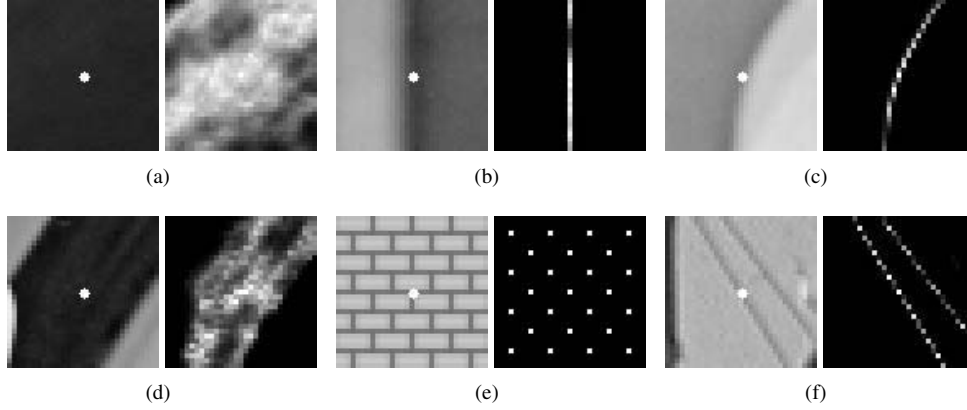
Figure 2. Display of the NL-means weight distribution used to estimate the central pixel of every image. The weights go from 1(white) to zero(black).

4. NL-means consistency

Under stationarity assumptions, for a pixel i, the NL-means algorithm converges to the conditional expectation of i once observed a neighborhood of it. In this case, the stationarity conditions amount to say that as the size of the image grows we can find many similar patches for all the details of the image.

Let V be a random field and suppose that the noisy image v is a realization of V. Let Z denote the sequence of random variables $Z_i = \{Y_i, X_i\}$ where $Y_i = V(i)$ is real valued and $X_i = V(\mathcal{N}_i\backslash\{i\})$ is $\mathbb{R}^p$ valued. The NL-means is an estimator of the conditional expectation $r(i) = E[Y_i \mid X_i = v(\mathcal{N}_i\backslash\{i\})]$.

Theorem 4 (Conditional expectation theorem) *Let* $Z = \{V(i), V(\mathcal{N}_i\backslash\{i\})\}$ *for* $i = 1, 2, \ldots$ *be a strictly stationary and mixing process. Let* NL_n *denote the NL-means algorithm applied to the sequence* $Z_n = \{V(i), V(\mathcal{N}_i\backslash\{i\})\}_{i=1}^n$. *Then,*

$$|NL_n(j) - r(j)| \to 0 \quad a.s$$

for $j \in \{1, \ldots, n\}$.

The full statement of the hypothesis of the theorem and its proof can be found in a more general framework in [12]. This theorem tells us that the NL-means algorithm corrects the noisy image rather than trying to separate the noise (oscillatory) from the true image (smooth).

In the case that an additive white noise model is assumed, the next result shows that the conditional expectation is the function of $V(\mathcal{N}_i\backslash\{i\})$ that minimizes the mean square error with the true image u.

Theorem 5 *Let* V, U, N *be random fields on* I *such that* $V = U + N$, *where* N *is a signal independent white noise. Then, the following statements are hold.*

(i) $E[V(i) \mid X_i = x] = E[U(i) \mid X_i = x]$ *for all* $i \in I$ *and* $x \in \mathbb{R}^p$.

(ii) *The expected random variable* $E[U(i) \mid V(\mathcal{N}_i\backslash\{i\})]$ *is the function of* $V(\mathcal{N}_i\backslash\{i\})$ *that minimizes the mean square error*

$$\min_g E[U(i) - g(V(\mathcal{N}_i\backslash\{i\}))]^2$$

Similar optimality theoretical results have been obtained in [9] and presented for the denoising of binary images. Theoretical links between the two algorithms will be explored in a future work.

5. Discussion and experimentation

In this section we compare the local smoothing filters and the NL-means algorithm under three well defined criteria: the method noise, the visual quality of the restored image and the mean square error, that is, the Euclidean difference between the restored and true images.

For computational purposes of the NL-means algorithm, we can restrict the search of similar windows in a larger "search window" of size $S \times S$ pixels. In all the experimentation we have fixed a search window of 21×21 pixels and a similarity square neighborhood $\mathcal{N}_i$ of 7×7 pixels. If N^2 is the number of pixels of the image, then the final complexity of the algorithm is about $49 \times 441 \times N^2$.

The 7×7 similarity window has shown to be large enough to be robust to noise and small enough to take care of details and fine structure. The filtering parameter h has been fixed to $10 * \sigma$ when a noise of standard deviation σ is added. Due to the fast decay of the exponential kernel, large Euclidean distances lead to nearly zero weights acting as an automatic threshold, see Fig. 2.
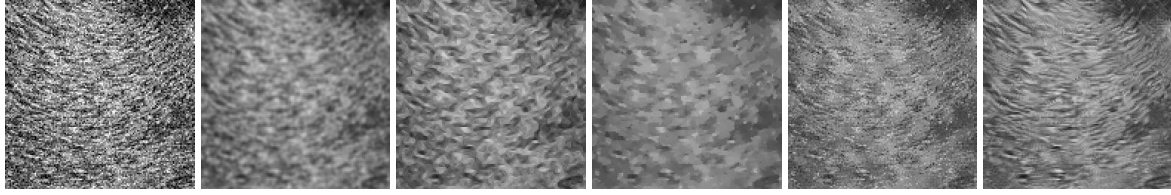
Figure 3. Denoising experience on a natural texture. From left to right: noisy image (standard deviation 35), Gauss filtering, anisotropic filtering, Total variation, Neighborhood filtering and NL-means algorithm.
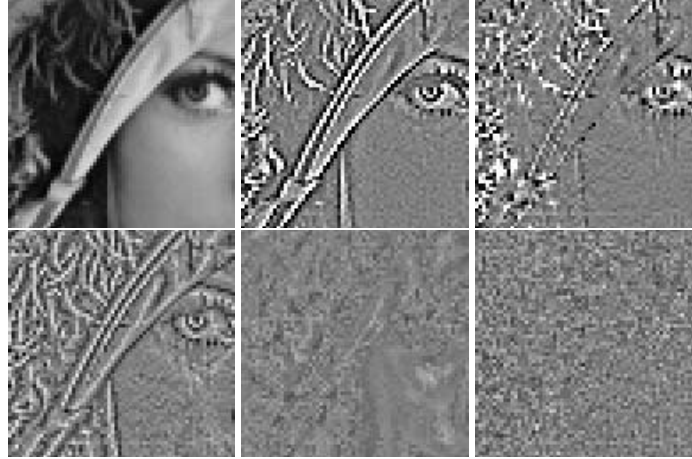

Figure 4. Method noise experience on a natural image. Displaying of the image difference $u - D_h(u)$. From left to right and from top to bottom: original image, Gauss filtering, anisotropic filtering, Total variation minimization, Neighborhood filtering and NL-means algorithm. The visual experiments corroborate the formulas of section 2.

In section 2 we have computed explicitly the method noise of the local smoothing filters. These formulas are corroborated by the visual experiments of Figure 4. This figure displays the method noise for the standard image Lena, that is, the difference $u - D_h(u)$, where the parameter h is been fixed in order to remove a noise of standard deviation 2.5. The method noise helps us to understand the performance and limitations of the denoising algorithms, since removed details or texture have a large method noise. We see in Figure 4 that the NL-means method noise does not present any noticeable geometrical structures. Figure 2 explains this property since it shows how the NL-means algorithm chooses a weighting configuration adapted to the local and non local geometry of the image.

The human eye is the only one able to decide if the quality of the image has been improved by the denoising method. We display some denoising experiences comparing the NL-means algorithm with local smoothing filters. All experiments have been simulated by adding a gaussian white noise of standard deviation σ to the true image. The objective is to compare the visual quality of the restored images, the non presence of artifacts and the correct recon-

struction of edges, texture and details.

Due to the nature of the algorithm, the most favorable case for the NL-means is the textured or periodic case. In this situation, for every pixel i, we can find a large set of samples with a very similar configuration. See Figure 2 e) for an example of the weight distribution of the NL-means algorithm for a periodic image. Figure 3 compares the performance of the NL-means and local smoothing filters for a natural texture.

Natural images also have enough redundancy to be restored by NL-means. Flat zones present a huge number of similar configurations lying inside the same object, see Figure 2 (a). Straight or curved edges have a complete line of pixels with similar configurations, see Figure 2 (b) and (c). In addition, natural images allow us to find many similar configurations in far away pixels, as Figure 2 (f) shows. Figure 5 shows an experiment on a natural image. This experience must be compared with Figure 4, where we display the method noise of the original image. The blurred or degraded structures of the restored images coincide with the noticeable structures of its method noise.

Finally Table 1 displays the mean square error for the

Figure 5. Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation 20), Gauss filtering, anisotropic filtering, Total variation, Neighborhood filtering and NL-means algorithm. The removed details must be compared with the method noise experience, Figure 4.

Image	GF	AF	TVF	YNF	NL
Lena	120	114	110	129	68
Baboon	507	418	365	381	292

Table 1. Mean square error table. A smaller mean square error indicates that the estimate is closer to the original image.

denoising experiments given in the paper. This numerical measurement is the most objective one, since it does not rely on any visual interpretation. However, this error is not computable in a real problem and a small mean square error does not assure a high visual quality. So all above discussed criteria seem necessary to compare the performance of algorithms.

References

[1] L. Alvarez, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion (ii). *Journal of numerical analysis*, 29:845–866, 1992.

[2] A. Buades, B. Coll, and J. Morel. On image denoising methods. Technical Report 2004-15, CMLA, 2004.

[3] A. Buades, B. Coll, and J. Morel. Neighborhood filters and pde's. Technical Report 2005-04, CMLA, 2005.

[4] R. Coifman and D. Donoho. *Wavelets and Statistics*, chapter Translation-invariant de-noising, pages 125–150. Springer Verlag, 1995.

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

[6] A. Efros and T. Leung. Texture synthesis by non parametric sampling. In *Proc. Int. Conf. Computer Vision*, volume 2, pages 1033–1038, 1999.

[7] M. Lindenbaum, M. Fischer, and A. Bruckstein. On gabor contribution to image enhancement. *Pattern Recognition*, 27:1–8, 1994.

[8] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, volume 22. AMS University Lecture Series, 2002.

[9] E. Ordentlich, G. Seroussi, M. W. S. Verdú, and T. Weissman. A discrete universal denoiser and its application to binary images. In *Proc. IEEE ICIP*, volume 1, pages 117–120, 2003.

[10] S. Osher, A. Sole, and L. Vese. Image decomposition and restoration using total variation minimization and the h^{-1} norm. *Multiscale Modeling and Simulation*, 1(3):349–370, 2003.

[11] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12:629–639, 1990.

[12] G. Roussas. Nonparametric regression estimation under mixing conditions. *Stochastic processes and their applications*, 36:107–116, 1990.

[13] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[14] S. Smith and J. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[15] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth Internatinal Conference on Computer Vision*, pages 839–846, 1998.

[16] L. Yaroslavsky. *Digital Picture Processing - An Introduction*. Springer Verlag, 1985.