A non-local algorithm for image denoising

Manpreet Singh

***Abstract*—In this paper we propose a new algorithm for image denoising called non-local means. Non-local means filtering takes a weighted average of all the pixels in the image, and weights are assigned based on pixel's similarity to the target pixel. We compare our results with the "local mean" filtering techniques like moving average, Gaussian filter, median filter, and TV denoising. Comparison results clearly show substantially improved post-filtering clarity and less loss of image detail with the Non-local means method.**

I. Introduction

In any digital color image or grayscale image, each pixel is subject to some perturbations. The random nature of the photon counting process in each sensor causes these perturbations. Noise can be enhanced by digital camera adjustments or picture processing software. Removing blur from photos or increasing contrast, results in increased noise in the image.

We can formulate the goal of image-denoising algorithms as recovering the original image from a noisy measurement:

$$v(i) = u(i) + n(i)$$

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 measurement of pixel values in any digital image is subject to some discrepancies. A simple and efficient way to model the effect of noise on digital images is to add a Gaussian white noise. In such a scenario, $n(i)$ are independent and identically distributed Gaussian values with a mean of zero and a variance of σ^2.

A number of methods have been proposed for noise removal and original image (u) recovery. The initial denoising algorithms were based on simply replacing a pixel with an average of the nearby/local pixel values. Some examples are the Gaussian smoothing model (Gabor [4]), the anisotropic filtering (Alvarez et al. [3]) and the neighborhood filtering (Smith et al. [5], Tomasi et al. [6]), Total Variation minimization - by the calculus of variations (Rudin-Osher-Fatemi [7]), the empirical Wiener filters - in the frequency domain (Yaroslavsky [9]), and wavelet thresholding methods (Coiffman-Donoho [11], [12]).

In this paper, we would define a denoising method as:

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

where v is the noisy image and h is the filtering parameter which depends on the standard deviation σ of the noise.

As we use averaging techniques, $D_h v$ ideally should be smoother than v and $n(D_h, v)$ should resemble the realization of white noise. Decomposing image into a smooth and a non-smooth (noisy) part is a current research topic (for example, Osher et al. [10]).

Our goal is to design a denoising technique that does not alter the original image u while recovering the image. Most of the methods mentioned above remove the texture and other fine details from u.

To perform denoising, the non-local means algorithm first takes a neighborhood around a pixel and gathers similar patches within a selected search window size. Then it takes an average of the similar pixels (whose patches are similar to current patch). This helps in restoring textures more efficiently as compared to other denoising methods.

The NL-means algorithm is defined by the formula below:

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

where, $x \in \Omega$,

$$C(x) = \int_\Omega e^{\frac{-\left(Ga^* |u(x+\cdot)-u(y+\cdot)|^2\right)(0)}{h^2}} dz$$

is a normalizing constant. G_a is a Gaussian kernel and h is a filtering parameter.

This formula states that the denoised value at x is the mean of the values of all points whose Gaussian neighborhood is similar to the neighborhood of x. The key difference between the NL-means method and local filters or frequency domain filters is the coordinated usage of all available self-predictions provided by the image, as seen in [8].

II. Methods

Here we shall discuss the methodology of some denoising algorithms like moving average, Gaussian filter, median filter, TV denoiser, and finally the NL-means algorithm.

A. Moving Average Filtering

Moving average or mean filtering works by simply replacing each pixel value in a picture with the mean ('average') value of its neighbors, which includes itself. This eliminates pixel values that are not representative of their surroundings. Mean filtering is usually referred to as a convolution filter. Fig. 1 represents how moving average works in convolution.

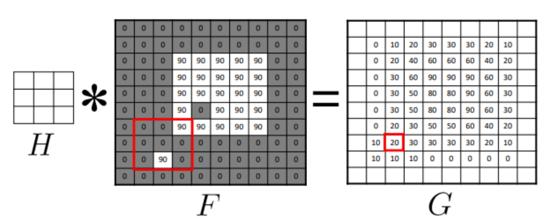


Fig. 1: An example of a 3x3 kernel used for moving average filtering. F is our image and H is the averaging filter, G is the smoothened image. Image has been referenced from [13].

B. Median Filtering

The median filter substitutes a pixel value with the median of nearby pixel values in the neighborhood rather than the mean of those values. The median is derived by first numerically sorting all of the pixel intensities in the surrounding neighborhood and then replacing the pixel under consideration with the middle pixel value. If the neighborhood has an even number of pixels, the average of the two middle pixel values is taken. Fig 2 illustrates how median filtering operates on an image.

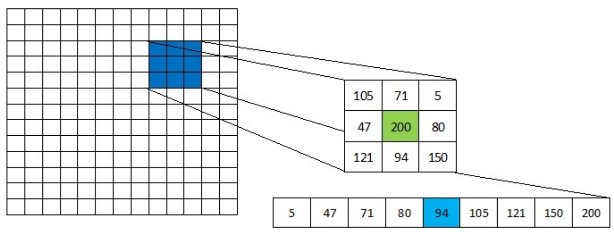


Fig. 2: An example of median filtering with a 3x3 kernel. Image has been referenced from [14].

C. Gaussian Filtering

Image isotropic linear filtering, according to Riesz's theorem, is a convolution of the image by a linear radial kernel. The paradigm for such kernels is the Gaussian $\mathbf{x}$ kernel:

$$G_h(x) = \frac{1}{4\pi h^2} e^{-\frac{|x|^2}{4h^2}}$$

Here G_h has standard deviation h, and we can easily observe the following theorem:

Theorem 1 (Gabor [4]): *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 convolution is ideal in flat parts of the image but blurs edges and texture.

D. Total Variation Denoising

Total variation denoising, also known as total variation regularization, is based on the principle that signals with excessive and potentially false information have a high total variation, i.e. the integral of the image gradient magnitude is large. This theory states that lowering the total variation of the signal—provided it is a close match to the original signal—removes unwanted detail and preserves relevant details such as the edges.

For a given noisy image $v(x)$, the original image $u(x)$ can be recovered as the solution of the minimization problem:

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

where $\mathrm{TV}(u)$ is the total variation of u and λ is a given Lagrange multiplier. A unique minimum value of the above minimization problem exists. The parameter λ is connected to noise statistics and regulates the degree of filtering of the produced solution.

E. Non Local Means Algorithm

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

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

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

The similarity between pixels i and j depends on the similarity of the intensity gray level vectors $v(N_i)$ and $v(N_j)$, where N_k is a square neighborhood of fixed size centered at pixel k. This similarity is measured as a decreasing function of the weighted Euclidean distance,

$$\|v(N_i) - v(N_j)\|_{2,a},$$

where $a > 0$ is the standard deviation of the Gaussian kernel.

The weights are defined as,

$$w(i,j) = \frac{1}{Z(i)} \exp\left(-\frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}\right),$$

where $Z(i)$ is the normalizing constant,

$$Z(i) = \sum_j \exp\left(-\frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}\right),$$

and the parameter h represents the degree of filtering. It regulates the decay of the exponential function and, as a result, the decay of the weights as a function of Euclidean distances.

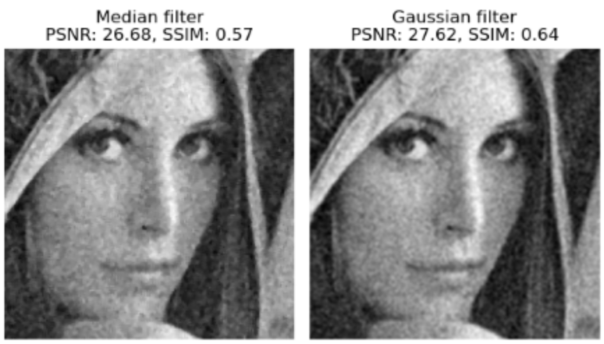
III. Results

In this section, we shall compare the results of our Non-local means denoising algorithm with all the local denoising filters. We shall add Gaussian noise on images with different standard deviations and report the Peak signal to noise ratio (PSNR) and Structural similarity (SSIM) values (by comparing denoised image with the original image).

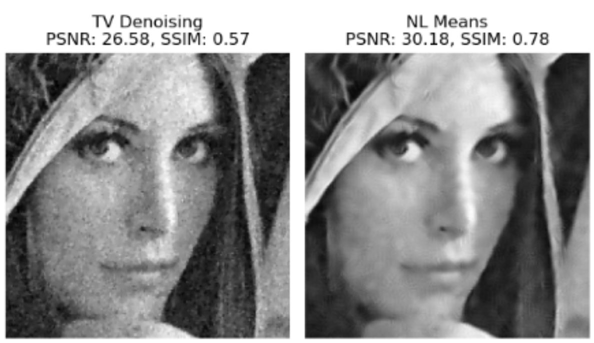
Method	$\sigma = \mathbf{0.1}$	$\sigma = \mathbf{0.15}$	$\sigma = \mathbf{0.2}$
	PSNR		
Moving Average	28.1293	25.49746	23.3919
Median filter	26.6751	23.6982	21.3693
Gaussian filter	27.6241	25.0092	22.9068
TV Denoising	26.5758	21.2882	17.9262
NL Means	30.1773	27.1644	26.1299
	SSIM		
Moving Average	0.6466	0.5112	0.4111
Median filter	0.5712	0.4286	0.3290
Gaussian filter	0.6369	0.4986	0.3977
TV Denoising	0.5713	0.3287	0.2143
NL Means	0.7806	0.6212	0.5878

TABLE I: PSNR and SSIM results of **Lena image**. Gaussian noise with $\sigma = 0.1$, $\sigma = 0.15$ and $\sigma = 0.2$ has been added for comparing denoising results.

(a)



(b)

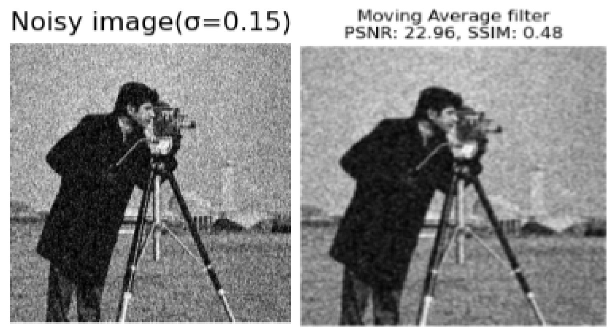


(c)

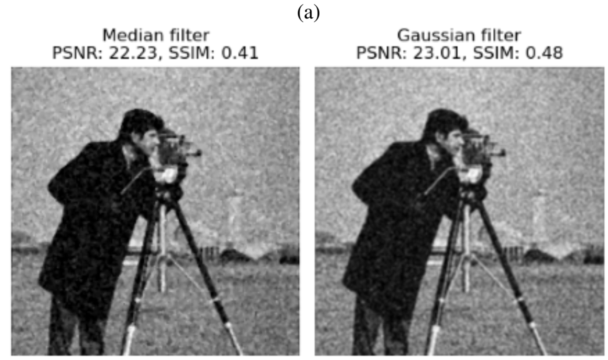
Fig. 3: Denoising performed on **Lena image** with $\sigma = 0.1$ with (a) Moving Average filtering, Median filtering, (b) Gaussian filtering, TV Denoising and (c) Non Local Means.

Method	$\sigma = \mathbf{0.1}$	$\sigma = \mathbf{0.15}$	$\sigma = \mathbf{0.2}$
	PSNR		
Moving Average	24.3714	22.9648	21.5529
Median filter	24.1468	22.2267	20.4809
Gaussian filter	24.6073	23.0103	21.4759
TV Denoising	25.9963	21.2766	18.0651
NL Means	28.0019	25.3159	23.3772
	SSIM		
Moving Average	0.5991	0.4761	0.3937
Median filter	0.5327	0.4133	0.3400
Gaussian filter	0.6073	0.4812	0.3983
TV Denoising	0.6004	0.3813	0.2809
NL Means	0.7614	0.6183	0.5586

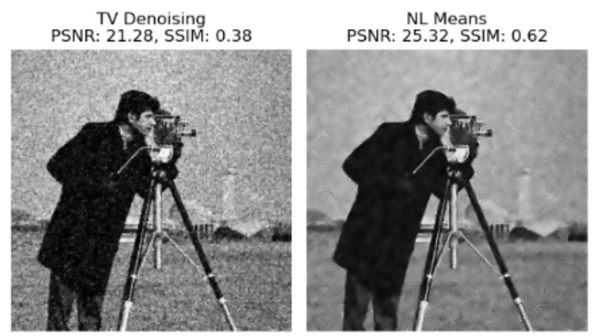
TABLE II: PSNR and SSIM results of **Cameraman image**. Gaussian noise with $\sigma = 0.1$, $\sigma = 0.15$ and $\sigma = 0.2$ has been added for comparing denoising results.



(a)



(b)

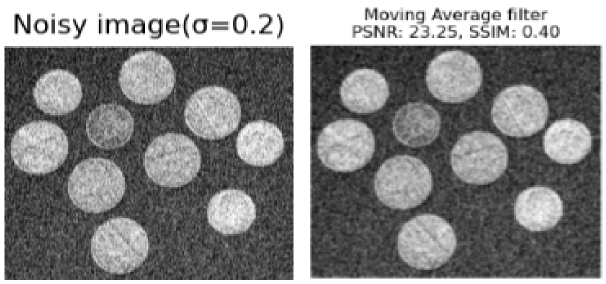


(c)

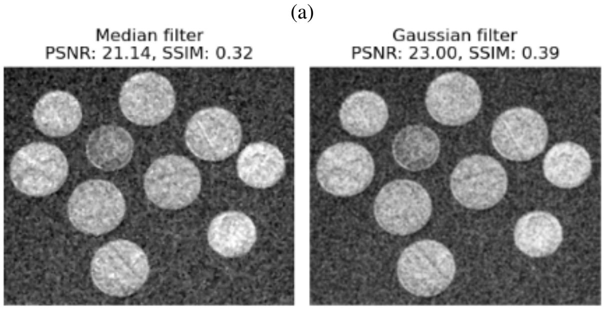
Fig. 4: Denoising performed on **Cameraman image** with $\sigma = 0.15$ with (a) Moving Average filtering, Median filtering, (b) Gaussian filtering, TV Denoising and (c) Non Local Means.

Method	$\sigma = \mathbf{0.1}$	$\sigma = \mathbf{0.15}$	$\sigma = \mathbf{0.2}$
	PSNR		
Moving Average	27.1851	24.9541	23.2457
Median filter	26.2629	23.3740	21.1353
Gaussian filter	27.1702	24.7674	22.9960
TV Denoising	26.4510	21.4027	18.2860
NL Means	30.1909	27.1622	25.8497
	SSIM		
Moving Average	0.6119	0.4813	0.4008
Median filter	0.5456	0.4082	0.3210
Gaussian filter	0.6025	0.4706	0.3913
TV Denoising	0.5450	0.3274	0.2338
NL Means	0.7842	0.6159	0.5915

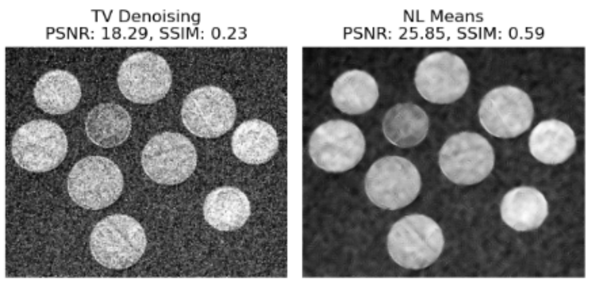
TABLE III: PSNR and SSIM results of **Coins image**. Gaussian noise with $\sigma = 0.1$, $\sigma = 0.15$ and $\sigma = 0.2$ has been added for comparing denoising results.



(a)

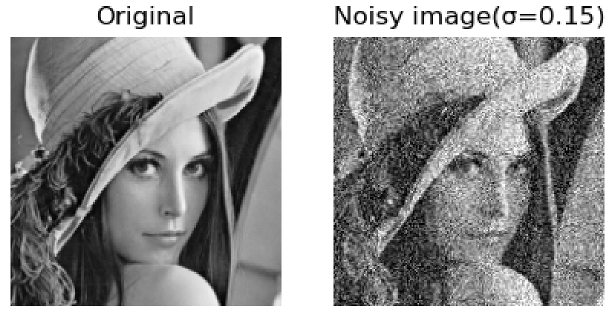


(b)



(c)

Fig. 5: Denoising performed on **Coins image** with $\sigma = 0.2$ with (a) Moving Average filtering, Median filtering, (b) Gaussian filtering, TV Denoising and (c) Non Local Means.



(a)

(b)

Fig. 6: Examples of Parameter sensitivity on (a) **Lena image** with Gaussian noise of $\sigma = 0.15$. (b) NL Means has been applied with different sizes of search window, neighborhood and filtering parameter h.

IV. Discussion

From tables I, II, III we can clearly see that PSNR values corresponding to the Non Local Means method show are higher than the other local denoising techniques. As higher PSNR indicates higher similarity of denoised image to original image, we can say that our NL-Means method provides the best quality of denoised image.

We further compare the SSIM values which compares the pixel-wise similarity and takes into account luminance, contrast, and structure of images. Higher SSIM values indicates higher similarity which again corroborates our claim that NL-Means is a better denoising method.

We use a similarity square neighborhood N_i of size 5 x 5 and we restrict the search of similar windows in a larger window of size 11 x 11. If N^2 is the number of pixels in the image, our final computation complexity becomes 25 x 121 x N^2. We use the Gaussian kernel standard deviation $a = 5$; the kernel is used for putting more weights on the central pixel within a similarity neighborhood.

A few notable limitations of NL-Means algotithm include:

1) **Dependency on Parameters**: We need to find the optimum Search window size, Similarity square neighborhood N_i, degree of filtering h and standard deviation of Gaussian kernel a as per our noise levels on input images to achieve better performance.
2) **High Computational time**: As NL Means has a high computational complexity, it works relatively slower on large images as compared to other denoising methods.
3) **Artefacts in flat regions:** NL Means may introduce artefacts in low contrast regions or regions with uniform textures. It finds difficulty while distinguishing between noise and subtle image features and this may lead to the loss of fine details.

Fig. 6 highlights the above limitations we face with NL Means. Higher window size leads to high computational time and we need an appropriate h value for lowering artefacts. Higher window size with a high h can even lead to over smoothening as shown in Fig. 6b.

V. Conclusion

Non-local means is a novel denoising algorithm proposed that is capable of achieving better results than other local means denoising methods. Despite the limitations mentioned above, we do get higher PSNR and SSIM values with NL Means, provided the appropriate parameters have been chosen.

Here we test the effectiveness of NL-means by denoising images that have Gaussian noise added (with varied standard deviations). However, for other kinds of noise (like salt and pepper noise) we need to readjust our parameters. NL - means mostly works well on images with additive noises.

Future work may include the non-local means algorithm to reduce computational complexity, experimenting with different measures of pixel similarity, artefacts reduction in flat zones and adapting it to different types of noise. This would make it more suitable for real-time applications.

References

[1] A. Buades, B. Coll and J. . -M. Morel, *"A non-local algorithm for image denoising,"* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 60-65 vol. 2, doi: 10.1109/CVPR.2005.38.

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

[3] 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.*

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

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

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

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

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

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

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

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

[13] Snavely, Noah *Filtering and Convolution (CS 5670 Lecture)*, Cornell University, 2021. https://www.cs.cornell.edu/courses/cs5670/2021sp/lectures/lec01_filter_for_web.pdf

[14] Viveko Blog, *Median Filter – edge preserving filter tutorial*, 2023 https://fiveko.com/median-filter/