On the Origin of the Bilateral Filter and Ways to Improve It

Parvik Dave
Electrical Engineering
Indian Institute of Technology Bombay
190070044

Rishabh Ravi
Electrical Engineering
Indian Institute of Technology Bombay
200260041

***Abstract*—Additive noise removal is an important problem in signal processing, for which many adaptive methods have been proposed. Iterative algorithms include anisotropic diffusion (AD), weighted least squares (WLS), and robust estimation (RE). This paper presents a theoretical bridge to the non-iterative bilateral filter that was proposed by Tomasi and Manduchi for noise removal. Since the filter was initially only presented as an intuitive approach, this paper shows that the bilateral filter emerges from the Bayesian Approach, as a single iteration of the Jacobi algorithm using a special loss function.**

I. Introduction

The paper discusses the problem of additive noise removal from a given signal in signal and image processing. The Bayesian approach is used to solve this problem, which involves applying a statistical estimator on a Gibbs distribution resulting in a penalty function. The function is then minimized by a numerical optimization algorithm to yield the restored signal. Noise removal is an important aspect that can improve the performance of many signal-processing algorithms, such as compression, detection, enhancement, and recognition.

Advanced methods for noise removal aim to preserve the signal details while removing the noise. These methods are based on locally adaptive recovery paradigms such as anisotropic diffusion, weighted least squares, robust estimation, and the Mumford–Shah function. These methods use iterative algorithms to take into account local relations between the samples.

The bilateral filter, proposed by Tomasi and Manduchi, is an alternative non-iterative filter that uses a weighted average of local neighborhood samples. The weights are computed based on temporal (or spatial) and radiometric distances between the center sample and the neighboring samples. This filter is locally adaptive and has been shown to give similar or better results than iterative approaches. In this paper, the author proposes a theoretical bridge between the bilateral filter and the AD, WLS, and RE techniques, showing that the bilateral filter also emerges from the Bayesian approach using a novel penalty function.

The author shows that a single iteration of the Jacobi algorithm yields the bilateral filter, and he proposes improvements to the filter to further speed up its smoothing operation. He also extends the filter to treat piece-wise linear signals and more general reconstruction problems such as image restoration, image scaling, and super-resolution.

II. Noise Suppression via the Bilateral Filter

To introduce the Bilateral Filter, we consider the 1D case, which can readily be extended to higher dimensions. An unknown signal $\underline{X}$ gets noisy by the addition of zero-mean AWGN $\underline{V}$. The corrupted signal is given by:

$$\underline{Y} = \underline{X} + \underline{V} \tag{1}$$

The bilateral filter uses a normalized weighted average of a neighbourhood of $2N+1$ samples around the kth sample. To recover the image $\underline{X}$:

$$\hat{X}[k] = \frac{\sum_{n=-N}^{N} W[k,n]Y[k-n]}{\sum_{n=-N}^{N} W[k,n]} \tag{2}$$

The weight $W[k,n]$ is composed of two weights W_S and W_R, which represent the spatial and radiometric weights. In summary, $W[k,n]$ is computed as:

$$W_S[k,n] = \exp\left(-\frac{n^2}{2\sigma_S^2}\right) \tag{3}$$

$$W_R[k,n] = \exp\left(-\frac{(Y[k]-Y[k-n])^2}{2\sigma_R^2}\right) \tag{4}$$

$$W[k,n] = W_S[k,n] \cdot W_R[k,n] \tag{5}$$

The weight is calculated using a Gaussian function, but other symmetric and smoothly decaying functions can be used as well. The kernel applied on the input signal at the kth sample has several properties. Firstly, the sum of the kernel's coefficients is 1 due to normalization. Secondly, the coefficient multiplying the center sample is the largest, and its size depends on the others due to normalization. Finally, subject to the first two constraints, the kernel can take any form. This property may be the basis for better performance, as is shown in later sections of the paper.

III. Derivation of the Bilateral Filter

In order to derive $W(Y)$, the WLS and RE approaches are compared. Using the WLS approach, the penalty function is:

$$\epsilon_{WLS}(\underline{X}) = \frac{1}{2}[\underline{X}-\underline{Y}]^T[\underline{X}-\underline{Y}] + \frac{\lambda}{2}[\underline{X}-D\underline{X}]^T W(Y)[\underline{X}-D\underline{X}] \tag{6}$$

And using the RE approach, the penalty function is:

$$\epsilon_{RE}(\underline{X}) = \frac{1}{2}[\underline{X}-\underline{Y}]^T[\underline{X}-\underline{Y}] + \frac{\lambda}{2}\rho(\underline{X}-D\underline{X}) \tag{7}$$

The matrix D stands for a one-sample shift to the right (toward the origin) operation. The matrix W is a diagonal matrix which weights the local gradients, with its main diagonal depending on the unknown image Y. Its construction method will be demonstrated later. The function $\rho(\alpha)$ in the RE is a non-negative, symmetric function that penalizes gradient values. Choosing $\rho(\alpha) = 0.5\alpha^2$ yields the trivial LS approach. This algorithm requires the computation of the first derivative of the penalty functions:

$$\frac{\partial \epsilon_{WLS}(\underline{X})}{\partial \underline{X}} = [\underline{X} - \underline{Y}]^T + \lambda [I - D]^T W(\underline{Y}) [I - D] \underline{X} \tag{8}$$

$$\frac{\partial \epsilon_{RE}(\underline{X})}{\partial \underline{X}} = [\underline{X} - \underline{Y}]^T + \lambda [I - D]^T \rho'([I - D] \underline{X}) \tag{9}$$

The steepest descent performed once with the initialization $\underline{Y}$ gives:

$$\hat{\underline{X}}_1^{WLS} = \underline{Y} - \mu\lambda (I - D)^T W(\underline{Y}) (I - D) \underline{Y} \tag{10}$$

$$\hat{\underline{X}}_1^{WLS} = \underline{Y} - \mu\lambda (I - D)^T \rho'((I - D) \underline{Y}) \tag{11}$$

The two iterative procedures provide the same estimate after one iteration provided that:

$$W(\underline{Y}) = \frac{\rho'((I - D)\underline{Y})}{(I - D)\underline{Y}} \tag{12}$$

, where the division performed is element-by-element.

Now, to formally derive the bilateral filter, a new penalty function similar to the WLS approach is introduced:

$$\epsilon(\underline{X}) = \frac{1}{2} [\underline{X} - \underline{Y}]^T [\underline{X} - \underline{Y}] + \frac{\lambda}{2} \sum_{n=1}^{N} [\underline{X} - D^n \underline{X}]^T W(\underline{Y}, n) [\underline{X} - D^n \underline{X}] \tag{13}$$

As opposed to the original WLS formulation, several scales of derivatives are used, all directly applying to the signal. Taking the first derivative with respect to the unknown $\underline{X}$, we get:

$$\frac{\partial \epsilon(\underline{X})}{\partial \underline{X}} = \left[I + \lambda \sum_{n=1}^{N} (I - D^{-n}) W(\underline{Y}, n) (I - D^n) \right] \underline{X} - \underline{Y} \tag{14}$$

where the relation $(D^n)^T = (D^T)^n = (D^{-1})^n = D^{-n}$ has been used. Assuming a single iteration of the steepest descent algorithm with step size μ and initialization $\underline{Y}$ we get:

$$\hat{X}_1 = \left[I - \mu\lambda \sum_{n=1}^{N} (I - D^{-n}) W(\underline{Y}, n) (I - D^n) \right] \underline{Y} \tag{15}$$

Speeding-up the above iteration can be done using locally adaptive step sizes. The inverse of the main diagonal of the Hessian matrix – the second derivative of the penalty function can be used. This algorithm is known as the Jacobi algorithm, or the diagonal normalized steepest descent (DNSD) algorithm. The second derivative is the following matrix:

$$\frac{\partial^2 \epsilon(\underline{X})}{\partial \underline{X}^2} = H(\underline{Y}) = I + \lambda \sum_{n=1}^{N} (I - D^{-n}) W(\underline{Y}, n) (I - D^n) \tag{16}$$

Extracting the main diagonal from this matrix, which by definition is known to contain real and positive values if all the weights are positive, we define a step size matrix, which extends the notion of the previously used by $M(Y)$. The additional term relaxes the step size matrix and ensures stability.

$$M(\underline{Y}) = [\zeta I + \operatorname{diag}\{H(\underline{Y})\}]^{-1} \tag{17}$$

Substituting $\mu = M(\underline{Y})$ in Eq. 15 will give the final DNSD iteration. In both the SD and the DNSD the solution is obtained via a linear operation on the distorted image. Referring to this linear operation as a signal-dependent linear filer, the kernel applied on a neighborhood of a sample is indeed the bilateral filter. To show this, it needs to be shown how the weights are chosen. We can use Eq. 12 to define the weights. However, the weights here should also reflect our decreased confidence in the smoothness penalty term as n grows toward N. Thus, a reasonable choice is:

$$W(\underline{Y}, n) = \frac{\rho'(I - D^n)\underline{Y}}{(I - D^n)\underline{Y}} V(n) \tag{18}$$

where $V(n)$ is a non-negative, symmetric, and monotonically decreasing function. With this, we have a one-to-one correspondence to the bilateral filter described in section II. We have:

$$W_S[k, l] \implies V(l)$$

$$W_R[k, l] \implies \frac{\rho'(Y[k] - Y[k - l])}{Y[k] - Y[k - l]}$$

Thus, if:

$$V(l) = \exp\left(-\frac{l^2}{2\sigma_S^2}\right) \tag{19}$$

and

$$\rho(\alpha) = -\sigma_R^2 \exp\left(-\frac{\alpha^2}{2\sigma_R^2}\right) \tag{20}$$

we get the same bilateral filter, as described by Tomasi and Manduchi.

IV. Proposed Idea

The bilateral filter is almost like a smoothing filter that averages each pixel in a grid. The weights, taken while averaging, decay as the pixel being considered moves further from the original pixel. Moreover, the weights also depend on the correlation between the two pixels. The averaging is one-dimensional, i.e., is taken along one of the axial directions.

One major drawback of the Bilateral filter is its selective performance towards images with a 'certain' symmetry. While it blurs images with the opposite symmetry. The symmetry here is the color gradient being either in the direction where averages are taken, or orthogonal to it.

To overcome this, we propose extending this averaging scheme to include axial and non-axial elements. This solves the issue of the selective performance of the filter. Furthermore, unlike the bilateral filter on images with orthogonal symmetry, it retains resolution on the boundary. This is because the

$$\epsilon\{X\} = \frac{1}{2}\left[X-Y\right]^T\left[X-Y\right] + \sum_{n=1}^{N}\sum_{m=1}^{N}\frac{\lambda}{2}\left[Y - \frac{1}{4}(D_L^n + D_R^n)X - \frac{1}{4}X(D_L^m + D_R^m)\right]^T \cdot W(Y,m,n)\cdot\left[Y - \frac{1}{4}(D_L^n + D_R^n)X - \frac{1}{4}X(D_L^m + D_R^m)\right]$$

averaging on boundary elements results in the cancellation of terms. As a result, there is almost close to little averaging done there.

Considering both the axial components without the non-axial elements, we get the newly chosen penalty function ϵ as shown above. Here D_R^n and D_L^n are rotational matrices that rotate the matrix n times right and left, respectively. We avoided the non-axial elements just to test our hypothesis and see if it does work.

V. RESULTS

We first tested our implementation of the bilateral filter against a standard library implementation of the same filter. The following image eas tested upon, Given in 2, we show

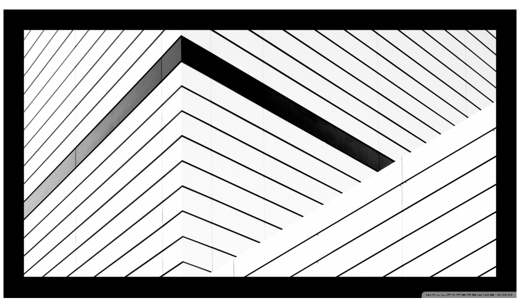

Figure 1. Original Image

the comparison of our implementation of the bilateral filter, the noisy image, and OPENCV's implementation of the same

It was visible that our implementation of the bilateral filter was accurate, and we could modify it to work on the modified penalty function.

We tested the above for different sections of the original image and verified the consistency of the proposed algorithm.

VI. CONCLUSION AND FUTURE WORK

The proposed algorithm had a significant upper hand over the bilateral filter. It successfully denoised the image, maintaining high resolution and decreasing the overall MSE. The only drawback is the time-complexity of the algorithm having to compute the n^{th} power of the rotation matrix D. This can be overcome by having a lookup table that stores the powers of D upto n, as after this D cycles back to its original form.

We would like to include more terms in the penalty function to incorporate non-axial elements and obtain a more clear image that has an even lesser MSE.

We would also like to extend the proposed algorithms to work on RGB images, which would involve extending the algorithm for higher dimensions as RGB images are higher dimensional.

ACKNOWLEDGEMENTS

We would like to thank our professor, Prof. Satish Mulleti for teaching us this course, without which we would not have been able to complete this project. We would also like to thank the TA Parth involved in this course, who helped us a lot in doing the various assignments, and completing this course.

REFERENCES

[1] M. Elad, "On the origin of the bilateral filter and ways to improve it," in IEEE Transactions on Image Processing, vol. 11, no. 10, pp. 1141-1151, Oct. 2002, doi: 10.1109/TIP.2002.801126.

[2] https://www.tutorialspoint.com/opencv/opencv_bilateral_filter.htm

[3] https://www.geeksforgeeks.org/python-bilateral-filtering/

[4] https://colab.research.google.com/drive/1ySROjor7H-UmmmnyvndzOzzRxKXu9XV0?usp=sharing

[5] https://colab.research.google.com/drive/1UO3Gjv05KpSKn9dO4L1mYfoF8-IQsaOA?usp=sharing

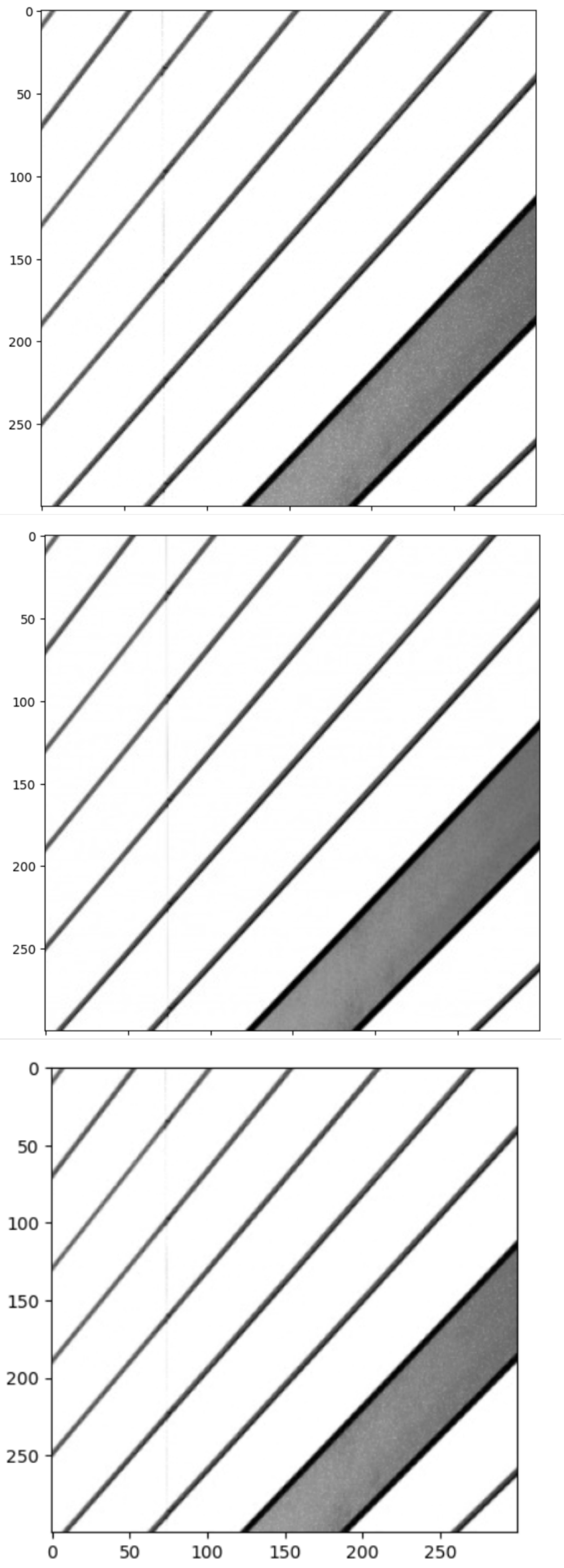

Figure 2. Shown above in sequence are a) Noisy image b) OPEN CV's bilateral filter c) Our implementation of bilateral filter

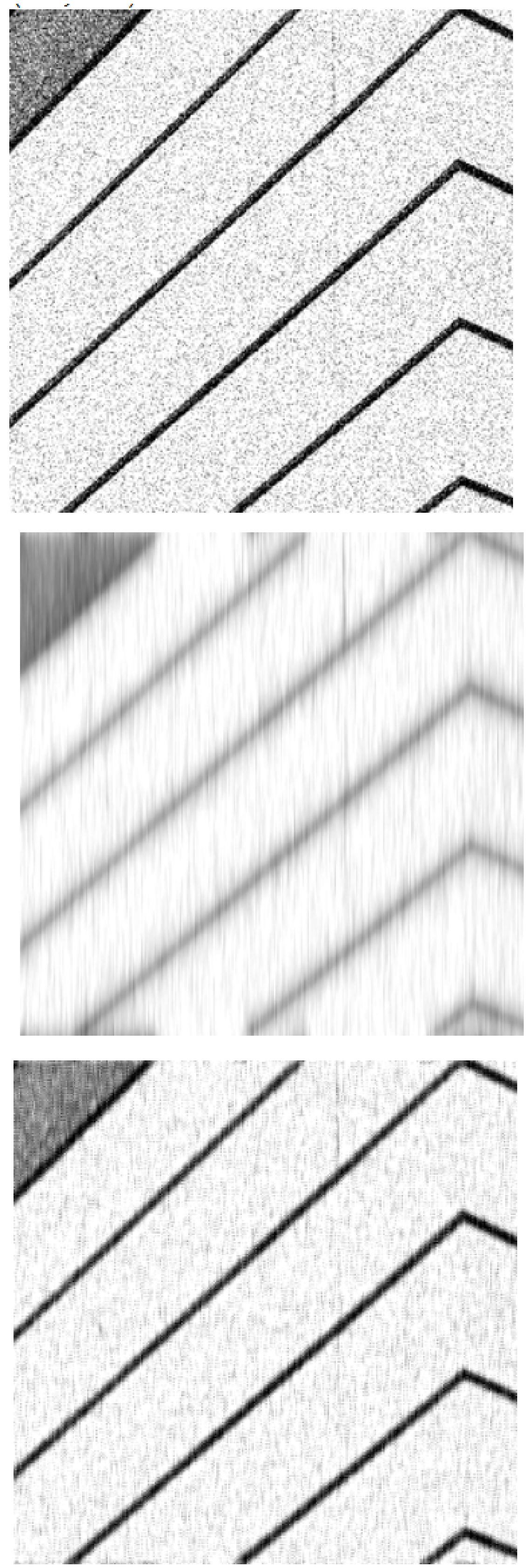

Figure 3. Shown above in sequence are a) Noisy image b) bilateral filter c) Proposed Idea