

The Bilateral Median Filter

J.J. Francis and G. de Jager

Department of Electrical Engineering, University of Cape Town,
Rondebosch, 7700, South Africa

{jfrancis,gdj}@dip.ee.uct.ac.za

Abstract

An extension of the bilateral filter is described. The proposed filter is a weighted median filter which adaptively estimates the weights in a similar manner to that of the bilateral filter. The proposed filter strikes a compromise between smoothing and preserving important detail.

1 Introduction

The bilateral filter was proposed by Tomasi and Manduchi as an image denoising algorithm [12]. Image denoising is a common preprocessing stage used to improve the visual appearance of images, and to improve subsequent image processing stages such as segmentation or motion estimation. The bilateral filter is a non-linear filter which takes into account local image information in order to build a kernel which smoothes without smoothing across edges.

It is clear from the literature that there is a direct relationship between bilateral filtering and robust estimation [13]. Also clear is the relationship between anisotropic diffusion and robust estimation [2]. It is known that there is a fundamental relationship between the bilateral filter and anisotropic diffusion [1, 5, 10].

This paper considers a natural extension to the bilateral filter. Instead of a weighted summation of the pixels in a neighbourhood, the result is the weighted median of pixels. The median filtering approach helps to prevent outlier pixels from unduly distorting the result. There are a number of weighted median filters described in the image analysis literature (see [3] for some examples). These filters tend not to be sufficiently data adaptive. For example, pixels across a strong edge may still significantly influence the median.

The traditional bilateral filter uses the Gaussian kernel for both spatial and range (or tonal) filtering. Replacing the kernel offers the possibility of reducing computational time without impairing performance severely.

This paper begins by describing the traditional bilateral filter. The modifications are then described in detail. A series of experiments comparing the proposed filter and the original bilateral filter are described.

2 The traditional bilateral filter

The bilateral filter is a non-linear denoising filter. A Gaussian kernel is used for domain and range filtering. This results in data dependent filtering.

The bilateral filter simultaneously weights pixels based on spatial distance from the centre pixel as well as distance in tone (intensity for example). The domain filter weights pixels based on their distance from the centre:

$$v(x - y) = \frac{1}{2} e^{-\frac{(x-y)(x-y)}{2\sigma_D^2}}, \quad (1)$$

where x and y denote pixel spatial positions. The spatial scale is set by σ_D . The range filter weights pixels based on the photometric (tonal) difference:

$$w(f(x) - f(y)) = \frac{1}{2} e^{-\frac{(f(x)-f(y))(f(x)-f(y))}{2\sigma_R^2}}, \quad (2)$$

where $f(\cdot)$ denotes image tonal values (intensity or colour). The degree of tonal filtering is set by σ_R . The bilateral filter is then:

$$\frac{\int_{R^d} f(y)v(x-y)w(f(x)-f(y))dy}{\int_{R^d} v(x-y)w(f(x)-f(y))dy} \quad (3)$$

Note that kernels other than Gaussian kernels are not excluded. See Durand and Dorsey for an example of other kernels applied to bilateral filtering [4].

Boomgaard and Weijer [13] generalise the bilateral filter to space-tonal convolution. This is an attempt to address a fundamental problem with bilateral filters. Noise affects all pixels, even the centre pixel ($f(x)$) used as a reference for the tonal filtering. Noise affecting the centre pixel thus has a disproportionate effect on the result. Boomgaard et al propose replacing the centre pixel with some estimate of the true value. What this may be is not specified. A similar approach was suggested by Perona and Malik to reduce the influence of noisy edges on anisotropic diffusion [9]. Perona and Malik suggested low-pass filtering with a kernel of small spatial extent. Again, Boomgaard et al use a Gaussian kernel in their examples of spatial-tonal convolution.

3 The modified bilateral filter

The traditional bilateral filter performs a weighted averaging of a neighbourhood. Noise influencing the centre pixel has a disproportionate influence on the range filtering. This suggests the following modifications:

- (i) Replacing the summation. In this paper, the pixels are combined using a weighted median. Other methods, for example stack filters [7, 14], are possible.
- (ii) Exploring alternative kernels for both domain and range filtering.

This provides, in theory, a number of advantages. A simpler kernel than the Gaussian may be easier to compute, leading to a performance speed tradeoff. Selecting the median rather than the mean leads to a filter more tolerant of outliers and hence noise perturbed centre pixels. The Gaussian kernel is of infinite extent, this leads to averaging even when the weight allocated is small. Consider the simple case of a step edge, irrespective of how small one makes the domain or range sigmas, the step is rounded, or smoothed. This is because all pixels contribute to the averaging process, even though their weights might be small. One can make a case that strong edges should be left intact. This prompts the consideration of other kernel functions. A similar argument is used by Black et al to justify the use of alternate edge stopping functions for anisotropic diffusion [2].

There appears to be no compelling reason to use kernels from the robust statistical literature only. Maintaining strong edges requires that the range kernel is of finite extent. In effect, any finite extent, low pass kernel will do. The kernel is then selected based on the desired smoothing as well as the computational requirements.

There is also no compelling reason for the range kernel to be based on weighted differences between pixels only.

4 Potential kernels

Two factors are of interest: the degree of outlier rejection by each kernel (or the degree of edge preservation), and the spatial extent of the kernel which selects the spatial scale of the filter. The following robust kernels are used [9, 2, 11]:

$$\text{Andrew's wave: } g(x, \sigma) = \begin{cases} \frac{\sin(\pi x/\sigma)}{\pi x \sigma} & |x| \leq \sigma \\ 0 & |x| > \sigma \end{cases}$$

$$\text{El Fallah Ford } g(x, \sigma) = \frac{1}{\sqrt{1 + (x/\sigma)^2}}$$

$$\text{Gaussian: } g(x, \sigma) = e^{-\frac{x^2}{2\sigma^2}}$$

$$\text{Huber's mini-max: } g(x, \sigma) = \begin{cases} \frac{1}{\sigma} & |x| \leq \sigma \\ \frac{\text{sign}(x)}{x} & |x| > \sigma \end{cases}$$

$$\text{Lorentzian: } g(x, \sigma) = \frac{2}{2\sigma^2 + x^2}$$

$$\text{Tukey bi-weight: } g(x, \sigma) = \begin{cases} \frac{1}{2} \left(1 - \left(\frac{x}{\sigma}\right)^2\right)^2 & |x| \leq \sigma \\ 0 & |x| > \sigma \end{cases}$$

The following kernels are considered as well:

$$\text{Cosine: } g(x, \sigma) = \begin{cases} \cos\left(\frac{\pi x}{2\sigma}\right) & |x| \leq \sigma \\ 0 & |x| > \sigma \end{cases}$$

$$\text{Flat } g(x, \sigma) = \begin{cases} \frac{1}{\sigma} & |x| \leq \sigma \\ 0 & |x| > \sigma \end{cases}$$

The Flat kernel is simple to compute, and when used as a domain filter allows for the effect of various range filters to be considered in isolation.

The kernels are dilated to ensure that the kernels are at the same spatial scale. The dilation is selected by forcing the peaks (where these exist) of the various influence functions ($\psi(x) = xg(x, \sigma)$ [11]) to line up. This allows the rejection of outliers to begin at (nearly) the same point for each kernel. This results in selecting a scale factor (σ_R) for the range filtering. Note that the El-Fallah Ford influence function only asymptotically approaches one. There is thus no extremum, which leads the choice of σ_R to be 0.99 of the maximum. It should be clear that selecting a larger value for σ_R leads compressing the influence function, leading to a resemblance to the Huber mini-max influence function.

The σ_R values are shown in table 1.

Kernel	σ_R	99% radius(pixels)
Andrews Wave	0.5	1
Cosine	0.5477	1
ElFallah Ford	10	20
Flat	1	1
Gaussian	1	3
Huber's mini-max	1	20
Lorentzian	$\sqrt{2}$	17
Tukey's bi-weight	$\frac{1}{\sqrt{5}}$	1

TABLE 1: Kernel tonal (σ_R) and spatial extent.

The spatial kernels are selected to be radially symmetric.¹ The spatial extent sets the size of the structures preserved or removed. Table 1 shows the spatial kernel radius at which kernel values are less than 0.01% of the peak values. This then selects the spatial extent of the filter.

¹Spatially separable kernels are expected to perform similarly.

5 Experimental results

Some variants of the proposed filter are tested. It is quite difficult to quantitatively compare denoising algorithms on real world data. Without ground truth data a comparison is impossible. The algorithms are first compared on artificial data (which to some degree resembles the real world data). Noise is added, and the Peak Signal to Noise Ratios (PSNRs) are compared. Several noise distributions are considered: Gaussian and “Salt and Pepper” noise.

The Peak Signal to Noise Ratio (PSNR) is defined as [3]:

$$\text{PSNR} = 20 \log \frac{\text{Peak value}}{\sqrt{\frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N (f(x, y) - g(x, y))^2}}, \quad (4)$$

where $f(x, y)$ denoted the original data (without noise added) and $g(x, y)$ denotes the filtered data.

This provides a measure of how different the filtered noise added image and the original images are.

5.1 Creation of the artificial data

A series of points are randomly placed on an empty image. The Voronoi tessellation [8] of the image is computed. The centroids of the Voronoi regions are found. The distance function from each centroid is computed. The distance function is then inverted and Phong shaded [6] to yield an artificial image of froth. An example artificial image is shown in figure 1. While the bubble shapes and boundaries are somewhat distorted, the individual bubbles and overall effect are fairly realistic as compared to figure 2.

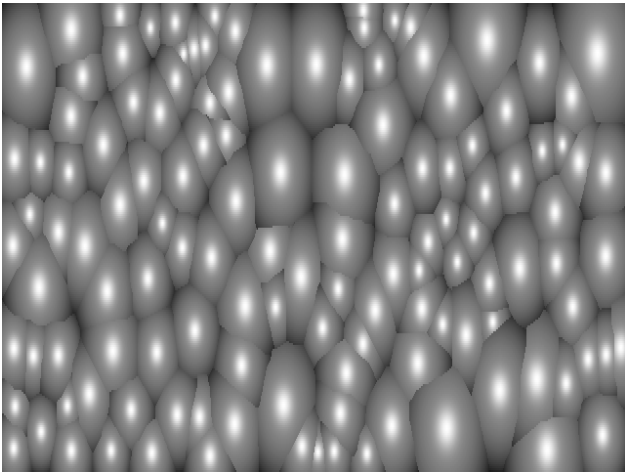


FIGURE 1: An example of artificial froth.

5.2 Experiments on artificial data

Artificial data was generated and analysed to provide a quantitative measurement of the denoising performance of

the proposed filter.

A sequence of ten images was generated and noise added. A range of kernels with different σ values was applied and the results summarised in tables 2 and 3. The first test added Gaussian noise with variance $\sigma^2 = 0.0016$ (with the maximum pixel intensity in the image rescaled to one). Tables 4 and 5 summarise the results for “Salt and Pepper” noise. The noise parameters were 1% probability of a pixel being Salt noise (or set to maximum intensity) and 1% probability of a pixel being Pepper noise (or set to minimum intensity). In the interests of brevity, only the best and the fastest range kernels are shown for each noise and domain filter type.

The traditional and proposed filter perform similarly for the Gaussian noise added case. It is clear that Gaussian domain and range kernels are less efficient than other alternatives, such as Flat domain and range kernels.

The situation for the “Salt and Pepper” noise is different. The Gaussian domain and Tukey Bi-Weight range performs very well for the proposed filter. The best performing kernels for the traditional filter are the Cosine kernels for both domain and range filtering. The Flat kernels again provide a fast alternative with performance close to the best kernel combination.

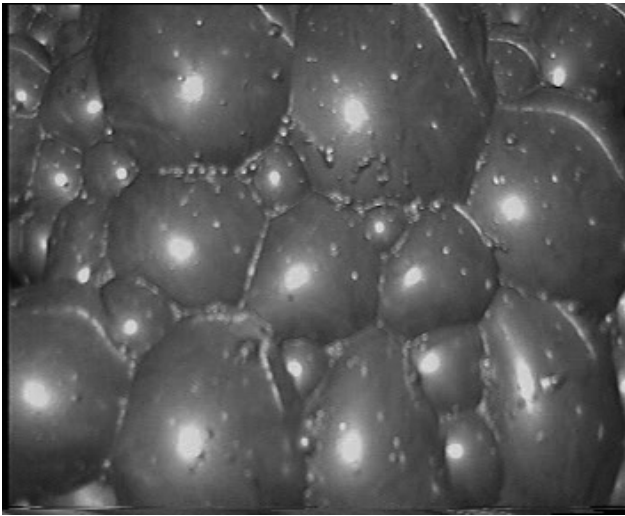
In contrast, a 3×3 median filter yields a PSNR of 34.4dB on the Gaussian added noise, and a PSNR of 41.9dB for the “Salt and Pepper” added noise. In both cases the proposed filter does better.

5.3 Experiments on real world data

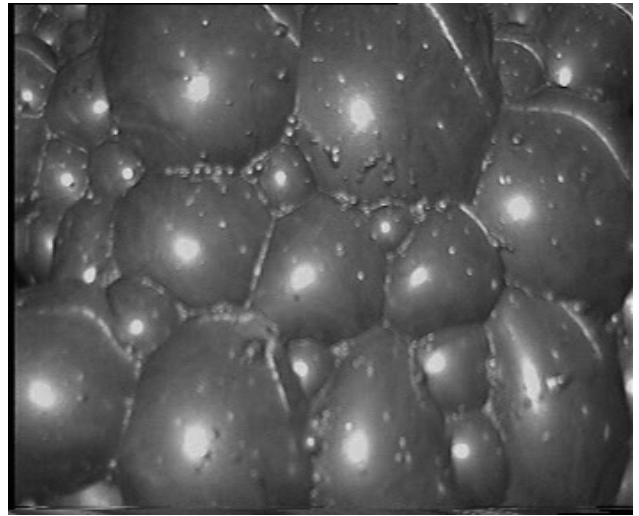
Some real world data was analysed to provide a qualitative understanding of the performance of the proposed filter. The image is of a froth cell with interfering light sources. These are: sunlight at an oblique angle leading to strip highlights on the bubbles, and plant lighting leading to tiny highlights visible on the large bubbles. Since highlights are natural markers for the bubbles, an attempt is made to remove the undesired highlights.

Figure 2 shows an example froth image, denoised version and error image after iterating the traditional bilateral filter four times. The domain and range sigmas are, 14 and 0.235 respectively. The kernel used for both domain and range filtering is the Andrew’s Wave kernel. The error image was enhanced by applying the gamma power law ($y = x^\gamma$) using a gamma of 4. The error image is inverted to make the detail more visible.

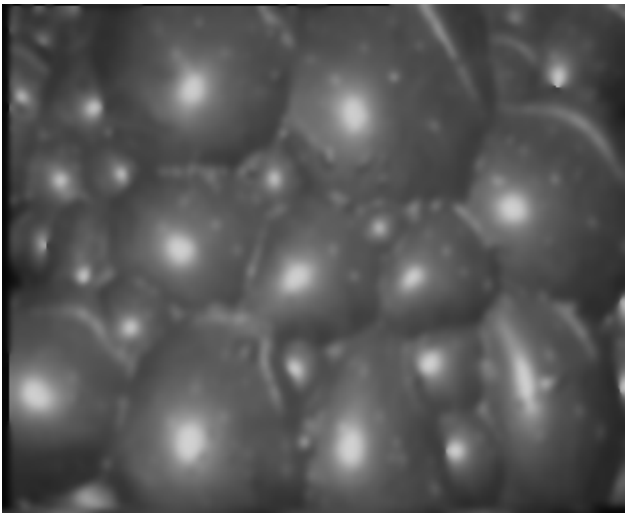
Figure 3 shows the result of applying the proposed bilateral filter four times (the domain and range sigmas and kernels are as before). Comparing the results of the two filters shows that the proposed filter does better at preserving detail (compare the error images), yet is better at removing the small point highlights. The traditional filter is seen to remove structures by blurring or smearing.



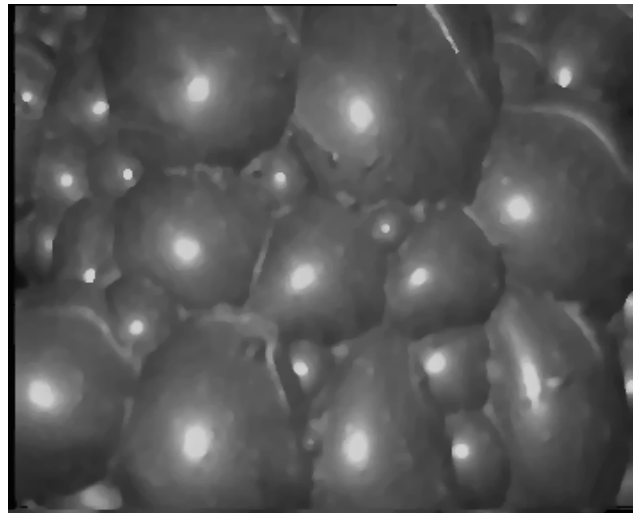
(a) An example image.



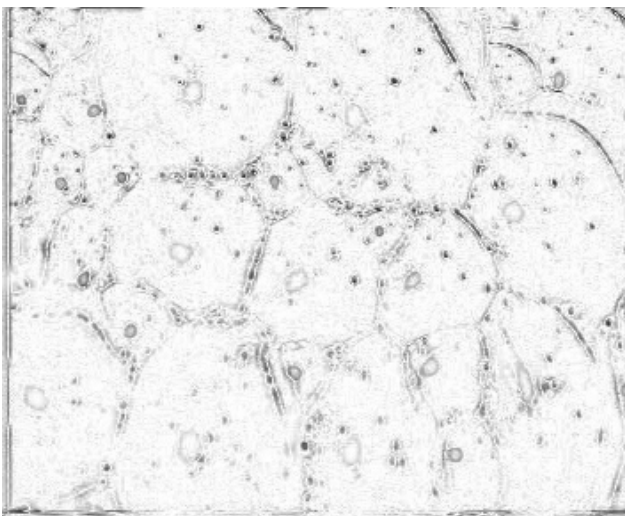
(a) An example image.



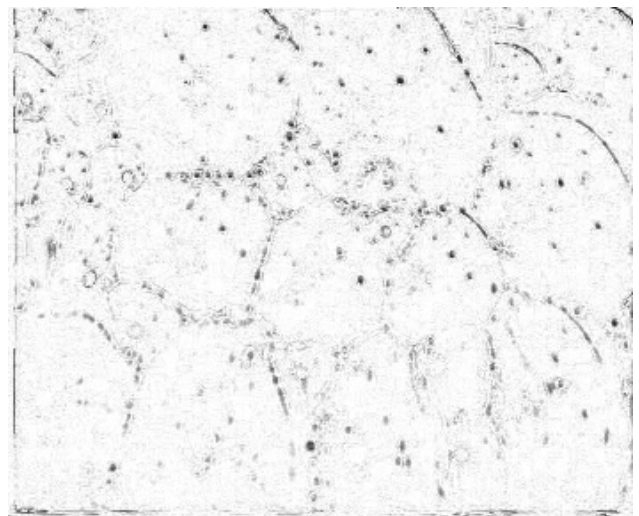
(b) Denoised by traditional bilateral filter.



(b) Denoised by proposed bilateral filter.



(c) Inverted and enhanced error image.



(c) Inverted and enhanced error image.

FIGURE 2: The traditional bilateral filter on a froth image.

FIGURE 3: The proposed bilateral filter on a froth image.

Range kernel	PSNR	σ_D	σ_R	Time(s)
Domain Filter: Andrews Wave				
Cosine	34.41	4.0	0.1	5.1
Flat	34.35	4.0	0.1	3.0
Domain Filter: Cosine				
AndrewsWave	38.39	7.3	0.1	48.1
Flat	37.69	7.3	0.1	22.4
Domain Filter: ElFallahFord				
ElFallahFord	27.95	0.005	0.1	0.54
Flat	27.97	0.005	1.1	0.58
Domain Filter: Flat				
AndrewsWave	37.56	4.0	0.1	17.3
Flat	37.00	4.0	0.1	8.5
Domain Filter: Gaussian				
Cosine	35.99	1.3	0.1	18.2
Flat	35.87	1.3	0.1	11.4
Domain Filter: HuberMiniMax				
Flat	36.48	0.2	0.1	8.2
HuberMiniMax	36.50	0.2	0.1	8.9
Domain Filter: Lorentzian				
Flat	31.79	0.17	3.1	8.3
HuberMiniMax	31.80	0.17	0.1	8.7
Domain Filter: TukeyBiWeight				
AndrewsWave	36.23	8.9	0.1	17.4
Flat	36.06	8.9	0.1	8.5

TABLE 2: The traditional filter on Gaussian noise.

6 Conclusions

A natural modification of the bilateral filter is proposed. The proposed filter is seen to provide a good compromise between detail preservation and removing image structures at the specified scale. Not surprisingly, the proposed filter performs better for ‘‘Salt and Pepper’’ noise as compared to additive Gaussian noise. The performance of the proposed and original filters is seen to be similar for Gaussian noise.

It is clear that alternative kernels can provide a compromise between speed and performance for both the traditional and proposed the bilateral filters.

7 Acknowledgements

Rio Tinto, AngloPlats and the NRF for financial and material support.

References

- [1] Danny Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and the

Range kernel	PSNR	σ_D	σ_R	Time(s)
Domain Filter: Andrews Wave				
Flat	32.74	3.5	0.1	4.8
Lorentzian	32.77	3.5	0.1	6.3
Domain Filter: Cosine				
AndrewsWave	37.15	7.3	0.1	71.8
Flat	36.89	7.3	0.1	46.3
Domain Filter: ElFallahFord				
AndrewsWave	27.96	0.005	0.1	0.61
Flat	27.99	0.005	0.1	0.62
Domain Filter: Flat				
AndrewsWave	36.14	4	0.1	25.3
Flat	35.93	4	0.1	15.8
Domain Filter: Gaussian				
Flat	34.25	1.3	0.1	25.1
HuberMiniMax	34.29	1.3	1.1	25.6
Domain Filter: HuberMiniMax				
Flat	33.55	0.2	0.1	16.1
HuberMiniMax	33.57	0.2	0.1	16.6
Domain Filter: Lorentzian				
Flat	27.89	0.15	0.1	16.1
Lorentzian	27.91	0.15	0.1	21.4
Domain Filter: TukeyBiWeight				
Flat	34.66	8.4	0.1	15.7
Lorentzian	34.79	8.4	0.1	20.6

TABLE 3: The proposed bilateral filter on Gaussian Noise.

non-linear diffusion equation. *PAMI*, 24(6), June 2002.

- [2] Michael J. Black, Guillermo Sapiro, David H. Marimont, and David Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998.
- [3] Al Bovik, editor. *Handbook of image and video processing*. Academic press series in communications, networking and multimedia. Academic Press, 2000.
- [4] Fredo Durand and Julie Dorsey. Fast bilateral filtering for the display of high dynamic range images. In *Proceedings of SIGGRAPH 2002*, pages 844–847, 2002.
- [5] Michael Elad. On the origin of the bilateral filter and ways to improve it. *ITIP*, 11(10):1141–1151, October 2002.
- [6] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles*

Range kernel	PSNR	σ_D	σ_R	Time(s)
Domain Filter: Andrews Wave				
ElFallahFord	29.31	4.0	1.1	4.79
Flat	29.14	4.0	1.1	2.97
Domain Filter: Cosine				
Cosine	34.65	7.3	1.1	42.51
Flat	34.00	7.3	6.1	22.10
Domain Filter: ElFallahFord				
AndrewsWave	22.53	0.0	0.1	0.55
Lorentzian	22.70	0.0	8.1	0.62
Domain Filter: Flat				
Flat	32.93	4.0	3.1	8.52
TukeyBiWeight	33.62	4.0	3.1	16.39
Domain Filter: Gaussian				
AndrewsWave	31.33	1.3	3.1	20.31
Flat	31.08	1.3	7.1	11.11
Domain Filter: HuberMiniMax				
Cosine	31.92	0.2	9.1	15.31
Flat	31.87	0.2	9.1	8.24
Domain Filter: Lorentzian				
Flat	26.36	0.2	2.1	8.89
HuberMiniMax	26.35	0.2	2.1	8.81
Domain Filter: TukeyBiWeight				
Flat	31.26	8.9	9.1	8.45
TukeyBiWeight	31.69	8.9	3.1	16.14

TABLE 4: The traditional bilateral filter on ‘‘Salt and Pepper’’ noise.

and Practice. The systems programming series. Addison-Wesley, second edition in C edition, 1996.

- [7] Jean-Hsang Lin, Nirwan Ansari, and Jinhui Li. Nonlinear filtering by threshold decomposition. *IEEE Transactions on Image Processing*, 8(7):925–933, July 1999.
- [8] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons, 1992.
- [9] P. Perona and J. Malik. Scale-space filtering and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(7):629 – 639, July 1990.
- [10] Nir A. Sochen, Ron Kimmel, and Alfred M. Bruckstein. Diffusions and confusions in signal and image processing. *Journal of Mathematical Imaging and Vision*, 14(3):195–209, 2001.

Range kernel	PSNR	σ_D	σ_R	Time(s)
Domain Filter: Andrews Wave				
Flat	45.87	3.5	1.1	4.72
Lorentzian	46.91	3.5	2.1	6.20
Domain Filter: Cosine				
Cosine	48.25	2.7	3.1	13.67
Flat	48.35	2.7	1.1	9.73
Domain Filter: ElFallahFord				
AndrewsWave	22.66	0.005	0.1	0.69
ElFallahFord	22.52	0.005	0.1	0.60
Domain Filter: Flat				
Flat	46.14	1.5	2.1	4.69
HuberMiniMax	46.22	1.5	1.1	4.90
Domain Filter: Gaussian				
Flat	49.06	0.8	1.1	10.01
TukeyBiWeight	49.69	1.2	8.1	28.20
Domain Filter: HuberMiniMax				
AndrewsWave	48.79	0.2	9.1	15.16
Flat	48.97	0.2	1.1	16.82
Domain Filter: Lorentzian				
AndrewsWave	23.85	0.0	0.1	2.75
Flat	23.69	0.0	0.1	1.89
Domain Filter: TukeyBiWeight				
Cosine	49.19	7.8	4.1	22.83
Flat	48.98	7.8	1.1	16.00

TABLE 5: The proposed bilateral filter on ‘Salt and Pepper’ noise.

- [11] Charles V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.
- [12] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *In Proceedings of the IEEE International Conference on Computer Vision*, pages 839–846, January 1998.
- [13] Rein van Boomgaard and Joost van de Weijer. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In R. Kasturi, D. Laurendeau, and C Suen, editors, *Proceedings 16th International Conference on Pattern Recognition*, volume 3, pages 927–930, Quebec City, Quebec, Canada, 11–15 August 2002. IEEE Computer Society.
- [14] Peter D. Wendt, Edward J. Coyle, and Neal C Gallagher Jr. Stack filters. *ITASSP*, 34(4):989–911, August 1986.