



## **Predictable blur behaviour for the bilateral filter**

**Researching a method for linear behaviour between the blurriness and spatial filter size of the bilateral filter**

**Bram Snelten**

**Supervisor(s): Prof.dr. E. Eisemann, MSc. M.L. Molenaar**

**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,  
In Partial Fulfilment of the Requirements  
For the Bachelor of Computer Science and Engineering  
June 23, 2024

Name of the student: Bram Snelten, TU Delft  
Final project course: CSE3000 Research Project  
Thesis committee: Prof.dr. E. Eisemann, MSc. M.L. Molenaar, Prof. J. Sun

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

## Abstract

Unlike traditional blur filters, the bilateral filter exhibits non-linear blur behaviour as its kernel size increases. This atypical blur behaviour makes it challenging to find a good  $\sigma_r$ . This paper investigates the underlying reasons for this behaviour and proposes methods to align the bilateral filter’s blur scaling linearly with its spatial filter size. Using local frequency analyses to quantify blur levels, we introduce an approach that finds the best  $\sigma_r$  through iterative search. Results demonstrate that the proposed method effectively counters the atypical blur behaviour. However, the proposed method does not perform sufficiently when handling very large kernel sizes. The proposed method can be used to abstract away the  $\sigma_r$  parameter when seeking linear blur behaviour in the bilateral filter. Further research is needed to make it functional for very large kernel sizes.

## 1 Introduction

Filtering is an operation used to modify images by applying a specific mathematical operation to each pixel. The goal of filtering is often to enhance certain aspects of the image, such as reducing noise, sharpening edges, or blurring details. A key concept in image filtering is the kernel, which is typically a square matrix of numbers; see Figure 1. Each number in the kernel represents a weight.

	0	0	1	2	1	0	0
	0	3	13	22	13	3	0
	1	13	59	97	59	13	1
1/1003	2	22	97	159	97	22	2
	1	13	59	97	59	13	1
	0	3	13	22	13	3	0
	0	0	1	2	1	0	0

Figure 1: A visualisation of a 7x7 Gaussian kernel.

The kernel is placed over pixels in the image, and their values are multiplied by the corresponding kernel weights. The products are summed to compute a weighted average, which is used to update the central pixel’s value in the output image. A popular noise reduction technique is the Gaussian blur, which utilises a 2D bell-shaped distribution as the kernel; see Figure 1. When applied to an image, the Gaussian blur results in a smoothed version of the original image, an example of which is shown in Figure 2.

This paper is about bilateral filtering [10] which blurs images while preserving edges. To blur an image while keeping

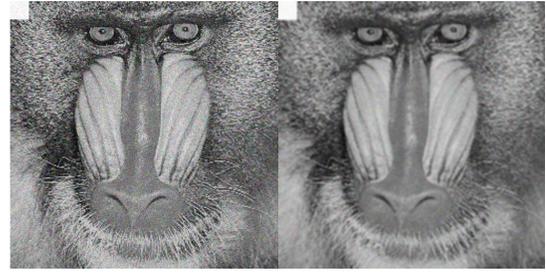


Figure 2: Left: Image with noise. Right: Image denoised using Gaussian blur.

edges, the bilateral filter multiplies two Gaussian kernels: the spatial kernel which weighs spatial distance, and the range kernel weighing difference in intensity. The range kernel aims to prevent blurring across edges by only averaging pixels in the kernel with similar intensities. The output of the bilateral filter centered at current pixel  $p$  is formally expressed as:

$$BF[I]_p = \frac{\sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q}{\sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|)} \quad (1)$$

Where  $BF[I]_p$  represents the pixel intensity at position  $p$  after bilateral filtering.  $S$  are the pixels that fall under the kernel neighborhood.  $q$  is a pixel that falls under the kernel.  $I_p$  and  $I_q$  are the intensity values at pixels  $p$  and  $q$  respectively.  $G$  is the Gaussian function.  $\sigma_s$  and  $\sigma_r$  are respectively the spatial and range standard deviations. The bilateral filter has several parameters that control its behaviour:

- $\sigma_s$ : The spatial sigma changes the standard deviation of the Gaussian spatial kernel. A larger value means that farther pixels will influence each other.
- $\sigma_r$ : The range sigma changes the standard deviation of the Gaussian range kernel. A larger value means that more different intensities will be mixed together.
- ‘Diameter’: The diameter of the kernel that is used during filtering. In this paper, we use OpenCV’s approach of computing it from the spatial sigma using the formula ‘diameter = 2 \* round(spatial sigma \* 1.5) + 1’ [8]. Note that when we refer to ‘spatial filter size’, we are considering both the kernel diameter and spatial sigma as a single parameter.

As with standard blur filters like a Gaussian blur, one would expect the image to get more blurred when increasing the spatial filter size. But this is not the case for the bilateral filter. This phenomenon is illustrated in Figure 3.

As you increase spatial filter size, the bilateral filter starts blurring more and more, as expected. This lasts up to a certain spatial filter size, the elbow point, depending on the image. Increasing the spatial filter size even more after that point results in the image getting less and less blurred. This phenomenon will hereafter be referred to as the atypical blur behaviour. In our observations of different test images, the blur level tends to decrease from a  $\sigma_s$  higher than 9. The reasons for this are discussed in Section 3.



Figure 3: Left: Image bilaterally filtered with parameters  $\sigma_r = 75$  and  $\sigma_s = 5$ . Right: Image bilaterally filtered with parameters  $\sigma_r = 75$  and  $\sigma_s = 29$ .

The bilateral filter has a multitude of applications, such as denoising in medical imaging or movie restoration and tone mapping [3]. When the bilateral filter is used to denoise medical images, it often uses small spatial filter sizes. This paper aims to explore methods to ensure the bilateral filter behaves predictably with larger spatial filter sizes. Therefore, this research could potentially expand the filter’s applications in medical imaging to include larger spatial filter sizes. Tone mapping applications often use a large  $\sigma_s$ , as large as 2% of the image diagonal [3]. ”This property is quite important because the user does not have to set a complex parameter” [3]. When testing different images, the atypical blur behaviour has been observed starting at  $\sigma_s$  9, meaning that images with as small diagonals as 450 pixels could already exhibit the atypical blur behaviour. Choosing the perfect range sigma for the bilateral filter is therefore challenging because the filter displays different behaviour depending on spatial filter size. This brings us to the research question.

This paper’s main question is: how to adapt the bilateral filter to have its perceived blur scale linearly with respect to its spatial filter size? The first sub-question of the paper is: Why does the bilateral filter exhibit this atypical blur behaviour? Next, we answer the sub-question: How do we measure perceived blur? The second sub-question, inspired by the similar approach researched by Liu et al. [5], is how to adapt the range kernel of the bilateral filter to counter the atypical blur behaviour.

After discussing related work in Section 2, Section 3 delves into why the bilateral filter exhibits the atypical blur behaviour. After which, Section 4 reports the research done on how to measure blur level of images filtered by the bilateral filter. Thirdly, Section 5 reports the development of an algorithm that automatically chooses the value for the range sigma parameter. Iterating upon the findings of the first algorithm, the development of a second algorithm that uses image content will be reported on in Section 6. Sections 4, 5 and 6 are complemented with their own results and discussion section. Section 7 discusses the integrity, reproducibility and ethics of this paper. Lastly, Section 8 is dedicated to concluding the paper, highlighting limitations and making recommendations for future work.

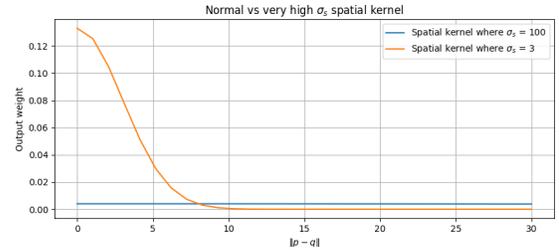


Figure 4: Comparison of normal,  $\sigma_s = 3$ , and very high,  $\sigma_s = 100$ , spatial kernels. The x-axis represents  $\|p - q\|$ , and the y-axis represents the output weight.

## 2 Related Work

Most work on adapting the range kernel has focused on making a faster bilateral filter implementation [1][4]. Fredo Durand and Julie Dorsey researched differences between influence functions for kernel types such as Gaussian and Huber [3]. This is useful to this paper because they also looked into blurring behaviour per influence function, stating: ”We can see that larger influences of outliers result in estimates that are more blurred and further from the input pixels” [3].

Our work is also related to the contributions by Tao Dai and Weizhi Lu, who developed an adapting bilateral filter that addresses the sensitivity of the standard bilateral filter’s range kernel to noise [2]. Dai and Lu’s work assumes that leveraging the difference between the noisy image and a denoised estimate will enhance denoising performance. However, their paper does not delve into the atypical blur behaviour.

Liu et al. [5] developed an adaptation of the bilateral filter that automatically chooses a range sigma based on local noise levels. Liu et al.’s approach is based on the assumption that noise levels are the primary factor affecting the performance of vision algorithms. This work relates to our paper as we also develop an algorithm that chooses a range sigma, but based on local intensity similarities instead of local noise levels.

Mir et al. conducted an empirical evaluation of blur metrics for digital cameras [7]. Their findings showed the effectiveness of simple first-derivative-based blur metrics. Pertuz et al. focused on comparing blur metrics for depth recovery and 3D reconstruction in computer vision [9]. However, the compared metrics in these related works do not mention the bilateral filter in any way, which could mean that their researched metrics might not perform well for images filtered with the bilateral filter.

Marziliano et al. introduced a blur metric for images that analyses the spread of edges in an image [6]. Their metric shows promise in enhancing autofocus capabilities in cameras. Their metric was validated through subjective perceptual experiments, which inspired the method of validating the researched metrics in this work.

## 3 Explanation atypical blur behaviour

We analysed the formal definition of the bilateral filter (see Equation 1) to find out why the atypical blur behaviour exists.

As  $\sigma_s$  increases, the spatial Gaussian function becomes wider. This means that pixels further away from the cen-

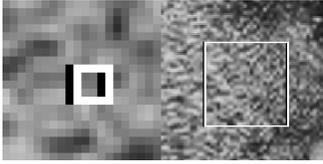


Figure 5: Left: Small black and white kernels drawn on two neighbouring pixels. Right: Big black and white kernels drawn on two neighbouring pixels

the pixel  $p$  will have higher weights than they would with a smaller  $\sigma_s$ . Now take  $\sigma_s$  very high. This ensures that  $G_{\sigma_s}(\|p - q\|)$  evaluates to the same output for different inputs  $\|p - q\|$  with negligible deviation, see Figure 4. The filter simplifies to:

$$BF[I]_p = \frac{\sum_{q \in S} G_{\sigma_r}(|I_p - I_q|)I_q}{\sum_{q \in S} G_{\sigma_r}(|I_p - I_q|)} \quad (2)$$

Another influence increasing  $\sigma_s$  has is indirectly increasing the kernel size, adding more pixels  $q$  to  $S$ . A larger  $S$  implies that neighbouring pixels have more similar neighbourhoods than if  $S$  were smaller; see Figure 5. This holds true for similarities such as Jaccard similarity or overlap coefficient.

With Equation 2 and the finding about increased neighbourhood similarity, we will conclude our explanation. Take neighbouring pixels  $p1$  and  $p2$ . With a high kernel size, their neighbourhoods  $S1$  and  $S2$  are similar with negligible difference, say  $S$ . This means that for  $p1$  and  $p2$  the only difference in  $BF[I]_p$  is  $I_p$  and therefore the input of the range kernel  $|I_p - I_q|$ . This results in different weights for the same pixels  $q$  in  $S$ . Which in turn makes  $BF[I]_{p1}$  output a different value than  $BF[I]_{p2}$ . This explains why neighbouring pixels keep the same structure instead of getting blurred.

$$GB[I]_p = \frac{\sum_{q \in S} G_{\sigma}(\|p - q\|)I_q}{\sum_{q \in S} G_{\sigma}(\|p - q\|)} \quad (3)$$

Lets contrast this to the predictable behaviour of the Gaussian blur, Equation 3, with high kernel sizes. The fact that neighbourhoods of neighbouring pixels are similar also holds for the Gaussian blur. This means that for  $p1$  and  $p2$  the only difference in  $GB[I]_p$  is the location of  $p$ . But because  $p1$  and  $p2$  are neighbours, that difference is very small. This results in similar weights for the same pixels  $q$  in  $S$ . Which in turn makes  $GB[I]_{p1}$  output a value similar to  $GB[I]_{p2}$ . This explains why these neighbouring pixels obtain similar values and get blurred.

## 4 Measuring blur

This research aims to improve the bilateral filter for linear blur scaling with kernel size. We define linear blur scaling by looking at the Gaussian blur, which does get more blurred as you increase its kernel size. The Gaussian blur serves as a good baseline since it is a standard blurring filter and people are well known with its behaviour. Therefore, it can be deemed to have predictable blur behaviour. In this paper, we defined linear scaling perceived blur as: the slope of the Gaussian blur levels. This means that our implementation should

try to match the slope of the Gaussian blur levels at each filter size interval.

After reading existing papers comparing blur metrics [7] [9] we picked four different blur metrics to try; Laplacian variance, gradient magnitude, local frequency analysis and the Brenner focus measure. These specific metrics were chosen to try and cover the most characteristics of all available blur metrics. The Laplacian variance for its simplicity and for being fast to compute. The gradient magnitude because it is more comprehensive but still simple to compute. The local frequency analysis was included because it involved a different perspective on detecting blur by using frequencies. And lastly, the Brenner focus measure because it was specifically designed to measure focus, the antonym of blur.

$$\text{Laplacian Kernel} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4)$$

The Laplacian variance metric is based on the observation that sharp images have more rapid intensity changes, while blurred images have fewer rapid intensity changes due to the smoothing effect of the blur. After convolving an image with a Laplacian kernel (see Equation 4), the resulting image will have high values at edges and low values in flat regions. A lower resulting variance indicates fewer areas of rapid change, meaning the image is more blurred.

$$\text{Sobel Kernel (x-direction)} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (5)$$

The way the gradient magnitude metric models blur is by measuring intensity transitions, also known as gradients. Smoother transitions indicate more blur. By convolving an image with Sobel kernels in both the x and y directions (see Equation 5 for the x direction Sobel kernel), we obtain the gradients along these directions. Then we take the mean of these gradients. A lower mean gradient magnitude indicates fewer pronounced edges, meaning the image is more blurred.

The local frequency analysis metric measures blur by analysing the frequency content of local regions in the image. The idea is that blurred images have more energy concentrated in lower frequencies, while sharp images have more energy distributed across higher frequencies. We divide the image into small blocks and perform a Fourier transform on each block to obtain its frequency spectrum. A higher ratio of low-frequency energy to total energy indicates more blur. By averaging these ratios across all blocks, we obtain an overall measure of the image blur.

The Brenner focus measure measures image sharpness by calculating squared differences in pixel intensities between neighbouring pixels along both horizontal and vertical directions. This approach emphasises rapid intensity changes or edges, whereas the gradient magnitude metric focuses on overall intensity changes across the image.

### 4.1 Method

When looking for the correct blur metric, it is important that the metric is accurate to the perceived blur. This means that the metric should result in a higher blur with small increases

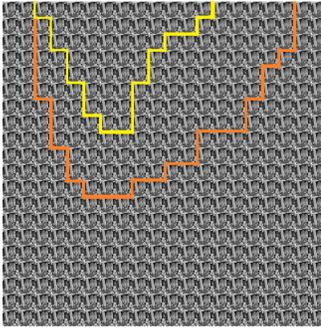


Figure 6: Grid of image filtered with the bilateral filter. Ascending  $\sigma_s \in [0, 40]$  along the horizontal axis, ascending  $\sigma_r \in [0, 40]$  along the vertical axis. Subjective contour lines are drawn, grouping images with similar perceived blur together. Images inside the yellow contour line are the most blurred.

in spatial filter size, up to the elbow point. Increasing the spatial filter size even more after that point should result in the blur levels decreasing. To evaluate the effectiveness of these blur metrics on bilaterally filtered images, we first made a grid of images filtered with the bilateral filter, with increasing sigmas along both axes in domains  $\sigma_s, \sigma_r \in [0, 40]$ . On a subjective basis, contour lines were drawn on the image grid, grouping images with the same perceived blur levels together; see Figure 6. Lastly, we made a contour plot for every blur metric, with the same parameter domain as the image grid. Then the contour plots were compared to the contours of the image grid.

## 4.2 Results and discussion

Figure 7 shows the contour plots for all four blur metrics. The metrics measure an increasing blur level with an increasing  $\sigma_r$ . This observation can be attributed to the fact that increasing  $\sigma_r$  in bilateral filtering results in more smoothing across intensities, thereby reducing the sharpness of edges and transitions and reducing higher frequencies in the image. However, only the local frequency analysis metric captures decreasing blur levels with increasing  $\sigma_s$ , which shows that the local frequency analysis metric does identify the atypical blur behaviour. A reason why these metrics do or do not capture the atypical blur behaviour is left for further research. When comparing the metric contour plots in Figure 7 to the subjectively drawn contour lines in Figure 6, we determined that the self-drawn contour lines matched most closely the shape of the local frequency analysis metric. Additionally, this metric also captures the atypical blur behaviour, which is a strong requirement. Therefore, we will develop our algorithm in Section 5 using the local frequency analysis metric to measure blur.

## 5 Brute force algorithm

Using the local frequency analysis method to quantify blur that matches our perception, we focus on the main question: how to adapt the bilateral filter to have its blur level scale linearly with kernel size. As stated in the first paragraph of Section 4, we made our definition of linear scaling the slope of

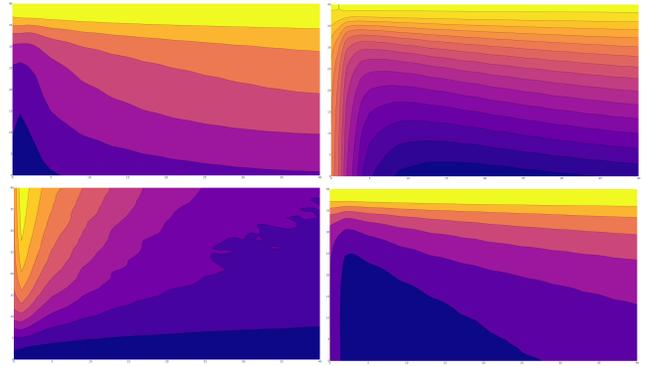


Figure 7: Contour plots of different blur metrics. Top left: Laplacian variance. Top right: gradient magnitude. Bottom left: local frequency analysis. Bottom right: Brenner focus measure. Ascending  $\sigma_s \in [0, 40]$  along the horizontal axis, ascending  $\sigma_r \in [0, 40]$  along the vertical axis. Images inside the yellow contour line are the most blurred.

Blurriness vs. Kernel size measured using local\_fourier\_analysis

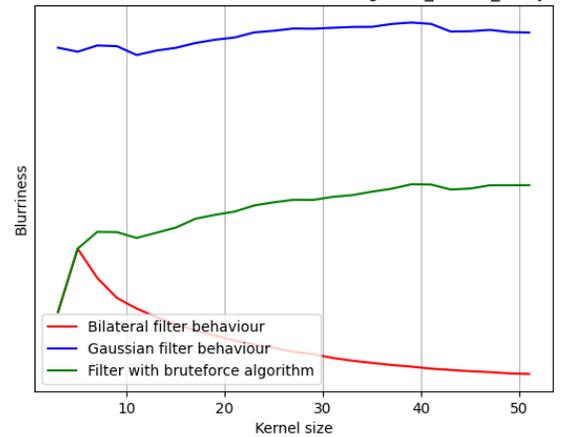


Figure 8: Kernel sizes and their blur levels measured for the normal bilateral filter in red, Gaussian blur in blue and the developed algorithm in green.

the Gaussian blur levels. When increasing spatial size, the bilateral filter behaves predictably until the elbow point. Therefore, after identifying the elbow point, the algorithm should alter the bilateral filter's blur levels to follow the same slope as the Gaussian blur level.

One way to alter blur level is by increasing  $\sigma_r$ , as can be seen in Figure 7. When you increase  $\sigma_r$ , the Gaussian range kernel becomes flatter. This means that the intensity differences between pixels have less influence on the weighting. In the limit as  $\sigma_r$  approaches infinity, the range kernel function  $G_{\sigma_r}$  returns the same value for every input. Meaning the bilateral filter turns into a regular Gaussian filter. The following algorithm builds upon this knowledge to map a  $\sigma_r$  to a specified spatial filter size to make the blur scale linearly.

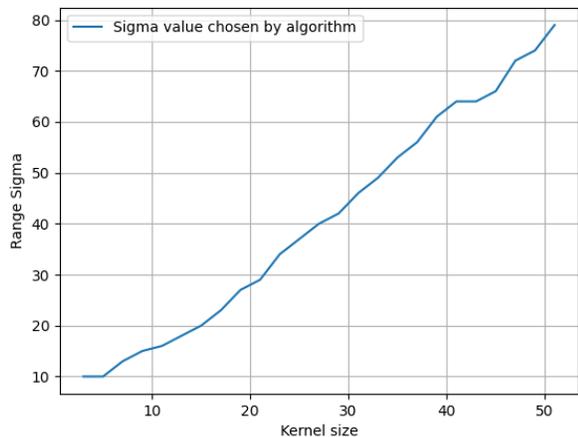


Figure 9: Spatial filter sizes and their corresponding  $\sigma_r$  values chosen by the algorithm.

## 5.1 Method

The first step of this algorithm is blurring the image using Gaussian blurs iterated over kernel sizes  $\in [1, \text{specified filter size}]$ . Each iteration the blur level is measured and saved. The second step of the algorithm is blurring using the bilateral filter and iterating over the same kernel sizes. To choose the best  $\sigma_r$ , the algorithm compares the slope of the bilateral filter’s blur levels between successive kernel sizes. When it detects the slope of the blur levels decreasing, it identifies the spatial size where the elbow point is located. From the elbow point onwards,  $\sigma_r$  is increased slightly to increase the slope of the bilateral filter’s blur levels. When the slope matches the slope of the Gaussian blur levels at that same filter size, it accepts the current  $\sigma_r$ . Incrementing  $\sigma_r$  in very small steps ensures that the lowest existing sufficient  $\sigma_r$  is detected while still matching the slope of the Gaussian blur levels.

The preliminary results in Figure 8 show that by increasing  $\sigma_r$ , the blur levels can be increased in a controlled manner to accurately recreate the slope of the Gaussian blur. However, when specifying a large kernel size, the algorithm takes a very long time. This happens because the search includes bilaterally filtering many times with large kernel sizes.

Noting Figure 9, the values of  $\sigma_r$  seem to relate linearly to the kernel size. This linear relationship was consistently observed across multiple test images, indicating a general pattern. This pattern inspires a simple improvement to the performance of the brute force algorithm. We can extrapolate  $\sigma_r$  values for each kernel size by sampling blur for only a few small kernel sizes and then fitting a line through these samples. For multiple images, this method gave reliable estimates for chosen  $\sigma_r$ , when taking samples of kernel sizes  $\in [1, 19]$ . This specific domain was determined by testing the time taken and the approximated error to the fully brute-forced sigma value. This improvement means that the algorithm only has to brute force search the corresponding  $\sigma_r$  for kernel sizes up until nineteen, after which it extrapolates the  $\sigma_r$  that corresponds to the specified kernel size. This im-

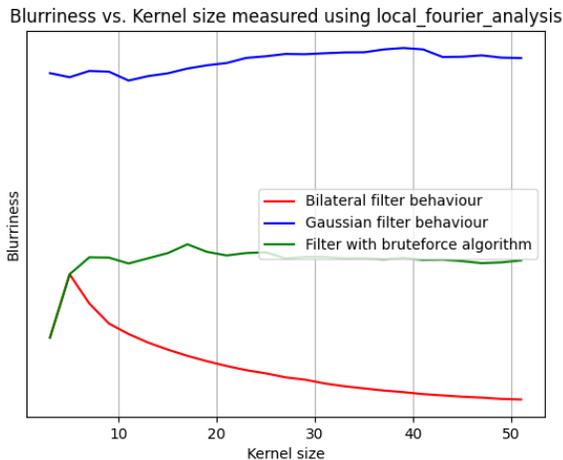


Figure 10: Kernel sizes and their blur levels measured for the normal bilateral filter in red, Gaussian blur in blue and the developed algorithm in green. The developed algorithm uses extrapolated  $\sigma_r$  values.



Figure 11: Left: image filtered with normal bilateral filter, kernel diameter = 51,  $\sigma_r = 10$  and  $\sigma_s = 17$ . Right: image filtered with brute force algorithm, kernel diameter = 51, starting  $\sigma_r = 10$  and blur metric = local frequency analysis.

provement means that the algorithm no longer has to bilaterally filter images with kernel sizes higher than nineteen.

## 5.2 Results and discussion

Using the code available at this GitHub repository, with parameters: max kernel size = 51, starting  $\sigma_r = 10$ , blur metric = local frequency analysis, the results in Figures 10 and 11 were obtained.

For the smallest kernel sizes, there is little difference between the algorithm and the normal bilateral filter. The algorithm detects that there is no need to increase  $\sigma_r$ . After the elbow point, the blur level slope matches the slope of the Gaussian blur level slope. Which, according to our definition of linear blur scaling, means that the blur levels scale linearly with respect to kernel size.

This partly answers the main research question. The resulting image in Figure 11 is clearly more blurred than its counterpart with the normal bilateral filter. However, when

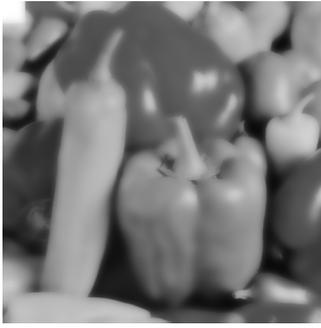


Figure 12: Image filtered with brute force algorithm, kernel diameter = 125, starting  $\sigma_r = 10$  and blur metric = local frequency analysis.

specifying a large kernel size with this algorithm, the resulting image lacks the distinctive edge preservation of the bilateral filter and instead resembles the Gaussian blur to a greater extent; see Figure 12. This implies the ineffectiveness of this adaptation in a scenario such as denoising a noised step function. One would use the bilateral filter to filter out noise while preserving the step. For very large kernel sizes, this adaptation will result in the step not being preserved.

This brute force algorithm can be useful in scenarios where users do not know which  $\sigma_r$  to choose, but know that their  $\sigma_s$  is relatively high. In such cases, the algorithm abstracts away the complexity of manually selecting  $\sigma_r$ . However, there is a risk associated with deploying this algorithm in important fields such as medical imaging. While the atypical blur behaviour would not be an issue, the brute force algorithm may unpredictably blur details that the normal bilateral filter would not blur.

After attempting this approach, we concluded that it does not achieve the desired bilateral blur effect. We found that for very large specified kernel sizes, the chosen  $\sigma_r$  was large enough to eliminate the influence of the range kernel, rendering the algorithm ineffective.

## 6 Adapting $\sigma_r$ algorithm

The brute-force algorithm finds the correct  $\sigma_r$  to counter the atypical blur behaviour. However, for very large kernel sizes, the characteristic edge preservation of the bilateral filter gets lost. This section reports on the failed development of a new algorithm that tries to counter the atypical blur behaviour while also preserving edges. This algorithm’s approach is to lower the range sigma for each pixel based on how close it is to an edge. For a noised step function, this should result in the function getting flattened, except when nearing the noised step.

### 6.1 Method

Firstly, we preprocess the image to find the edges. We do this in a novel way: for each pixel  $p$  in the image, the mean intensity  $\mu$  of the neighbourhood  $S_p$  of size  $k \times k$  is considered. Then the difference between  $\mu$  and  $I_p$  is computed. A higher difference indicates that pixel  $p$  is located near more pixels with another intensity. Therefore, it is closer to an edge or step.



Figure 13: Left: Image filtered using the brute-force algorithm, kernel diameter = 151, starting  $\sigma_r = 10$  and blur metric = local frequency analysis. Right: Image filtered using the adapting  $\sigma_r$  algorithm, kernel diameter = 151, base  $\sigma_r = 142$ ,  $k = 8$

Next, for each pixel  $p$ , an adaptive range sigma  $\sigma_r(p)$  is calculated based on the obtained differences. Pixels that are very similar to its neighbourhood  $S_p$  are assigned the  $\sigma_r$  computed by the previous algorithm. Whereas more different pixels receive a lower  $\sigma_r(p)$ . This adaptive  $\sigma_r(p)$  ensures that the bilateral filter blurs linearly in areas with similar structures and blurs less near areas with different structures.

Lastly, the image gets filtered using the normal bilateral filter. But with a different  $\sigma_r(p)$  for each pixel.

## 6.2 Results and Discussion

When first running the brute-force algorithm, using the code available at this GitHub repository, with parameters: max kernel size = 151, starting  $\sigma_r = 10$ , blur metric = local frequency analysis, the resulting  $\sigma_r$  is 142. Then running the adapting  $\sigma_r$  algorithm with max kernel size = 151, base  $\sigma_r = 142$  and  $k = 8$ . Results in Figure 13

The resulting image has some edges that are preserved compared to its counterpart with the brute force algorithm. However, not all edges are preserved and there are some very unpleasant artifacts. After attempting this approach, we concluded that it does not imitate the desired bilateral blur effect.

## 7 Responsible Research

The code used to obtain most results for this project can be found at <https://github.com/BramyBoyGG/Research-Project-Bilateral-Filter-Blur>. The code does not integrate any form of randomness, which means that the results in Subsection 5.2 and Figure 7 can be exactly reproduced using the provided code and the parameters specified in the figure captions. It should be noted that the results in Subsection 4.2 largely depend on Figure 6, where the blur level groupings were created subjectively. This makes exact reproduction of these results challenging. However, the full version of Figure 6 is available on GitHub, allowing for review of the contour lines. To improve transparency, the failed attempt to develop an algorithm that counters the atypical blur behaviour while preserving edges is also reported in 6. The code required to obtain the results in this section is not added to the complementary GitHub repository. Reasons for this include time constraints and the poor results.

The images of peppers [12] and mandrill [11] were obtained from the USC-SIPI Image Database. It should be noted that the copyright status of these images is unknown, as information on the source has been lost. The algorithms in this research require approximately an order of magnitude more computing power than the normal bilateral filter. This increased computational demand raises concerns regarding sustainability, as it leads to higher energy consumption.

Lastly, it must be stated that although this project was conducted for academic purposes, it was initiated as a mandatory course in the CSE bachelor's programme at TU Delft. In addition to the author stating that there are no external motivations, the integrity of the research process was supervised by the experienced academics Prof.dr. E. Eisemann and MSc. M.L. Molenaar.

## 8 Conclusions, limitations and Future Work

This paper focused on three areas: explaining the atypical blur behaviour of bilateral filters, identifying effective blur metrics, and adapting the bilateral filter to counter this behaviour. Firstly, analysing the formal definition of the bilateral filter yielded two interesting properties. For large spatial sizes, the bilateral filter's equation can be simplified to exclude the spatial kernel term. And for large spatial sizes, the neighbourhoods of neighbouring pixels are practically similar. These properties were combined to form the explanation of the atypical blur behaviour. This explanation was then contrasted with the typical blur behaviour of the Gaussian blur.

Secondly, after reviewing existing literature on blur metrics, we selected four different metrics for evaluation: Laplacian variance, gradient magnitude, local frequency analysis and the Brenner focus measure. We determined that the local frequency analysis metric most accurately captured the atypical blur behaviour and closely matched perceived blur levels.

Thirdly, we developed a brute-force algorithm that adjusts  $\sigma_r$  to ensure linear perceived blur scaling, effectively countering the atypical blur behaviour. Despite this success, this algorithm behaves like a Gaussian filter for very large spatial filter sizes. Losing its distinctive edge-preserving characteristics. This makes the adaptation less effective for applications for which one would use the bilateral filter. Iterating upon the first algorithm, we attempted to develop a new algorithm that addresses this limitation. This approach involved adjusting  $\sigma_r$  for each pixel based on its proximity to edges. Despite some success in preserving certain edges, the algorithm failed to mimic the desired bilateral blur effect.

A point of future work would be improving the attempted adapting  $\sigma_r$  algorithm. One could start by integrating known methods of detecting edges, such as Sobel or Canny edge detection, instead of the attempted novel way. Improving this could result in an algorithm that counters the atypical blur behaviour, while preserving edges like the bilateral filter. Another recommendation is to gain more understanding of the atypical blur behaviour by experimenting with range kernels other than Gaussian, such as the kernels used in the paper of Durand and Dorsey [3]. With insight into this one might create a specific range kernel that does not display the atypical blur behaviour.

## References

- [1] Kunal Narayan Chaudhury, Daniel Sage, and Michael Unser. Fast  $o(1)$  bilateral filtering using trigonometric range kernels. *IEEE Transactions on Image Processing*, 20(12):3376–3382, 2011.
- [2] Tao Dai, Weizhi Lu, Wei Wang, Jilei Wang, and Shu-Tao Xia. Entropy-based bilateral filtering with a new range kernel. *Signal Processing*, 137:223–234, 2017.
- [3] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics*, 21(3):257–266, July 2002.
- [4] Sanjay Ghosh and Kunal N. Chaudhury. On fast bilateral filtering using fourier kernels. *IEEE Signal Processing Letters*, 23(5):570–573, 2016.
- [5] Ce Liu, W.T. Freeman, R. Szeliski, and Sing Bing Kang. Noise estimation from a single image. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 901–908, 2006.
- [6] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. A no-reference perceptual blur metric. In *Proceedings. International Conference on Image Processing*, volume 3, pages III–III, 2002.
- [7] Hashim Mir, Peter Xu, and Peter van Beek. An extensive empirical evaluation of focus measures for digital photography. In Nitin Sampat, Radka Tezaur, Sebastiano Battiato, and Boyd A. Fowler, editors, *Digital Photography X*, volume 9023, page 90230I. International Society for Optics and Photonics, SPIE, 2014.
- [8] OpenCV. Bilateral filtering. [https://docs.opencv.org/4.x/d4/d86/group\\_imgproc\\_filter.html#ga9d7064d478c95d60003cf839430737ed](https://docs.opencv.org/4.x/d4/d86/group_imgproc_filter.html#ga9d7064d478c95d60003cf839430737ed). Accessed: 2024-06-11.
- [9] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013.
- [10] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, 1998.
- [11] Unknown. Baboon (mandrill) image, n.d. Retrieved from USC-SIPI Image Database.
- [12] Unknown. Peppers image, n.d. Retrieved from USC-SIPI Image Database.