



## **Curvature-Based Bilateral Filter for Image Smoothing**

**Dmytro Maksymchuk**

**Supervisors: Elmar Eisemann, Mathijs Molenaar**

**<sup>1</sup>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: Dmytro Maksymchuk,  
Final project course: CSE3000 Research Project  
Thesis committee: Elmar Eisemann, Mathijs Molenaar, Jing Sung

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

## Abstract

This paper introduces the Quadrilateral filter, an advanced extension of the Bilateral and Trilateral filters, aimed at addressing limitations in high-gradient regions of images. While the Bilateral filter effectively preserves edges during smoothing, it struggles with intensity variations, leading to blunted image details. The trilateral filter improves upon this by incorporating local plane geometry approximations but assumes linear pixel intensity distributions, limiting its effectiveness. The proposed Quadrilateral filter utilizes curvature-based geometry approximations to enhance noise reduction, contrast preservation, artifact reduction, and image reconstruction by accounting for nonlinear pixel value distributions. The development of this filter represents the main contribution of the paper while exploring whether the established Bilateral and Trilateral filters' performance can be further improved through curvature-based local geometry approximations. The findings demonstrate improvements in image quality and detail preservation, with broad implications for applications in image de-noising, tone-mapping, multimedia processing, and beyond.

## 1 Introduction

Image processing is a fundamental area in computer science, with wide-ranging applications including medical imaging, multimedia processing, and computational photography. Among the various techniques used in image processing, the bilateral filter, proposed by Tomasi and Manduchi [6], is renowned for its ability to preserve edges while smoothing images in a single pass. This filter has proven effective across numerous applications such as image denoising [9], tone mapping [3], low-light photography [4], deinterlacing [7], and cartoon renditions of images [8].

Despite its versatility, the bilateral filter struggles in high-gradient regions, where it fails to smooth images effectively due to its inability to incorporate varying intensity levels of neighbouring pixels. Another pitfall of the bilateral filter is its tendency to blunt the peaks and valleys in the image, which can reduce overall image contrast and detail. To address these challenges, Choudhury and Tumblin [2] proposed the Trilateral filter. This filter extends the bilateral filter by incorporating local plane geometry approximation. However, the Trilateral filter assumes that pixel intensity distributions in images are locally linear, which is often not the case. This assumption limits its effectiveness in more complex scenarios.

This paper expands upon the Trilateral filter by introducing a novel Quadrilateral filter, which further refines the filtering process through curvature-based geometry approximations. This approach aims to enhance the filter's capabilities in noise reduction, contrast preservation, artifact reduction, and improved image reconstruction. By considering the nonlinear distribution of pixel values from neighbouring pixels, the Quadrilateral filter addresses the limitations of its prede-

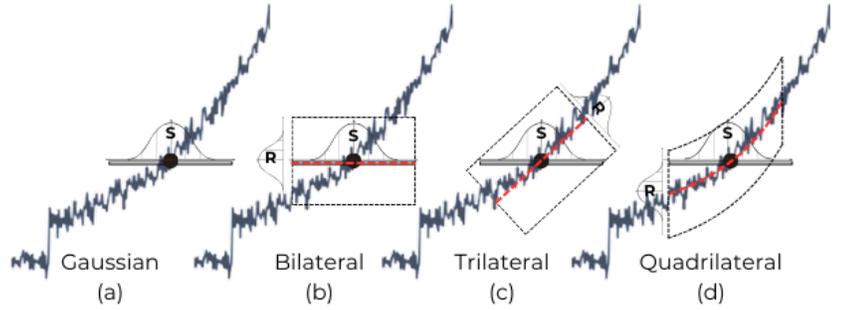


Figure 1: Filter window comparison.

cessors and provides a more robust solution for high-gradient regions.

The key research question addressed in this study is: *Can the performance of the Bilateral and Trilateral filters be further enhanced by utilizing curvature-based local geometry approximations for the pixel intensity distribution?*

By exploring this question, the study aims to develop advanced filtering techniques that improve image quality and detail preservation, ultimately benefiting a wide range of applications from medical imaging to multimedia processing.

## 2 Filter Preliminaries

Filters produce an output for each point of the signal  $I_{out}$  by combining together the original value of the point  $I_{in}$  and the values of the neighbouring points in the signal. The way the combination occurs and the selected neighbourhood depends on the filter. This concept of filters extends to N-dimensions; however, for simplicity of illustration, it will be explained for a 1-D signal.

### 2.1 Bilateral Filter

For a start, it is important to understand how the Gaussian filter works. It sums the values in the neighbourhood by weighing them using a kernel  $G()$  as seen in Figure 1(a). This results in the output:

$$I_{out}(q) = \sum_{p \in S} I_{in}(q) G_{\sigma}(\|p - q\|) \quad (1)$$

where  $G(x)$ :

$$G_{\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$

And  $S$  represents a set of points in the kernel window, considered as neighbours. Usually, the size of the kernel with radius  $1.5 * \sigma$  is used.

The Bilateral filter takes it a step further, by taking into account the variation of intensities to preserve edges. Two points are considered close to each other if both are nearby in the spatial domain and are similar in intensities. The range kernel is introduced to represent the similarity in intensity seen in Figure 1(b).

$$BF(q) = \frac{1}{k(q)} \sum_{p \in S} I_{in}(q) G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_{in}(q) - I_{in}(p)\|) \quad (3)$$

where  $k$  is the normalization factor:

$$k(q) = \sum_{p \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_{in}(q) - I_{in}(p)\|) \quad (4)$$

Variances of spatial and range kernels are represented by  $\sigma_s$  and  $\sigma_r$  respectively. These parameters affect the blurring behaviour of the Bilateral filter.

## 2.2 Trilateral Filter

Proposed by Choudhury and Tumblin, the Trilateral filter [2] extends upon the Bilateral filter. It modifies the range kernel by "tilting" the range kernel in order to account for gradients present in the image. Instead of directly comparing intensities of the neighbouring points, it approximates a plane through the neighbourhood and compares the neighbouring point intensities to the plane as seen in Figure 1(c). This allows for smoothing areas where neighbouring points are not similar to each other (such as high-gradient areas). The use of a Bilateral filter is suggested in order to approximate the tilting angle of the plane:

$$G_{\theta}(q) = \frac{1}{k(q)} \sum_{p \in S} \nabla I(q) G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|\nabla I(p) - \nabla I(q)\|) \quad (5)$$

where  $\nabla I_{in}(x)$  is the gradient of the point computed as:

$$\nabla I(x, y) \approx (I(x+1, y) - I(x, y), I(x, y+1) - I(x, y)) \quad (6)$$

The forward difference is used as opposed to central to minimize the smoothing effect.

The plane is then computed as:

$$P(q, p) = I_{in}(q) + G_{\theta}(q) * (p - q) \quad (7)$$

Deviation from the plane is defined as:

$$I_{\delta}(p, q) = I_{in}(q) - P(p, q) \quad (8)$$

The final value is then computed as:

$$I_{out}(q) = I_{in}(q) + \frac{1}{k_{\delta}} \sum_{p \in S} I_{\delta}(p, q) G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_{\delta}(p, q)) \quad (9)$$

with normalization factor  $k_{\delta}$ :

$$k_{\delta}(q) = \sum_{p \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_{\delta}(p, q)) \quad (10)$$

## 3 Quadrilateral Filter

The Quadrilateral filter presented in this paper aims to enhance the accuracy of image reconstruction by considering the non-linear distribution of pixel values from neighbouring pixels. The main contributions of this approach are:

- *Quadratic Surface Mapping*: Using a quadratic surface approximation for the local geometry.
- *Uncertainty Estimation*: A measure of how well the plane is mapped onto the neighbourhood pixels, used to blend filtering results adaptively.

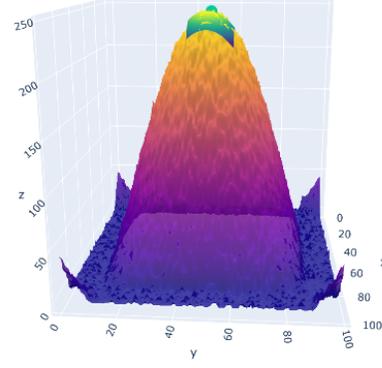


Figure 2: A quadratic plane reconstruction of the local geometry of a noisy 2-D signal at a point.

## 3.1 Quadratic Surface Mapping

The quadratic surface allows us to represent the desired curvature of the pixel value distribution as shown in Figures 1(d) and 2. A second-order Taylor series expansion is employed to construct the quadratic surface approximation for each pixel. To find plane approximation at point  $(x, y)$ , from original  $(x_0, y_0)$

$$P_{quad}(x, y) = \underbrace{f(x_0, y_0)}_{\text{Order 0 part}} + \underbrace{f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)}_{\text{Order 1 part}} + \underbrace{\left\{ \begin{array}{l} \frac{1}{2} f_{xx}(x_0, y_0)(x - x_0)^2 \\ + f_{xy}(x_0, y_0)(x - x_0)(y - y_0) \\ + \frac{1}{2} f_{yy}(x_0, y_0)(y - y_0)^2 \end{array} \right\}}_{\text{Order 2 part}} \quad (11)$$

For this, it is necessary to compute partial derivatives of the input image. Just as in the case of the trilateral filter, which can be thought of as a first-order Taylor series approximation, a bilateral filter is used to average gradients of an image, rejecting gradients that are very different from the current pixel. The partial derivatives for  $x$  and  $xx$  are shown below, and the rest are calculated similarly.

$$f_x(q) = \frac{1}{k(q)} \sum_{p \in S} \nabla_x I(q) G_{\sigma_s}(p_x - q_x) G_{\sigma_r}(\|\nabla_x I(p) - \nabla_x I(q)\|) \quad (12)$$

$$f_{xx}(q) = \frac{1}{k(q)} \sum_{p \in S} \nabla_x f_x(q) G_{\sigma_s}(p_x - q_x) G_{\sigma_r}(\|\nabla_x f_x(p) - \nabla_x f_x(q)\|) \quad (13)$$

Following the methodology proposed in the trilateral filter [2], forward differences are used to calculate the gradient at a point. The gradient  $\nabla I$  is computed as:

$$\begin{aligned}\nabla_x I(x, y) &\approx I(x + 1, y) - I(x, y) \\ \nabla_y I(x, y) &\approx I(x, y + 1) - I(x, y)\end{aligned}\quad (14)$$

The quadratic plane is used to find the deviation of the neighbouring pixels from the approximated geometry:

$$I_\delta(p, q) = I_{in}(q) - P_{quad}(p, q) \quad (15)$$

$$I_{out}(p) = I_{in}(q) + \frac{1}{k_\delta} \sum_{p \in S} I_\delta(p, q) G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_\delta(p, q)) \quad (16)$$

$$k_\delta(p) = \sum_{p \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_\delta(p, q)) \quad (17)$$

This approach was chosen as an extension of the concepts introduced in the Trilateral filter paper due to its robustness and efficiency in approximating linear geometry. This easily extends to Taylor's second-order series approximation to capture the curvature in the local geometry. This extension comes at a computational cost described in Section 4.5.

### 3.2 Uncertainty Estimation

The quadratic approximation may not always accurately represent the input signal. In such instances, the data may not be appropriately filtered, making it preferable to adopt a simpler geometric approximation, such as the Bilateral filter. To identify these specific areas, the concept of uncertainty is introduced. This concept was first explored in the Fast Bilateral Filtering paper [3], where it serves to identify anomalous pixels in images. In the context of the Quadrilateral filter, high uncertainty values would indicate pixels where the plane approximation does not closely match the actual intensities of the adjacent pixels. As detailed in [3], the normalization factor  $k$  from Equation 17 is employed as an uncertainty estimator.

The normalization factor must be mapped to a meaningful [0,1] range for interpolation with the bilateral filter. Uncertainty should be higher for outliers where the points are farther away from the reconstruction plane. To find areas with lower than average normalization values  $k$ , a sigmoid function is used:

$$U(x) = \frac{1}{1 + e^{a*(k-\mu_k)/\sigma_k} + b} \quad (18)$$

Here, the  $\mu_k$  is the average normalization factor, and  $\sigma_k$  is the standard deviation of  $k$  values. There are 2 constant present in the formula, the higher the value of  $a$  the more of an outlier the point needs to be for high uncertainty value. The higher the second constant,  $b$ , the smaller uncertainty values are in general, so less interpolation is involved.

High uncertainty values indicate regions where the quadratic approximation performs poorly, such as at sharp edges. In these cases, interpolation between the output of the Quadrilateral filter  $Q$  and the Bilateral filter  $B$  is calculated as follows:

$$I_{final}(q) = (1 - u) \cdot Q(q) + u \cdot B(q), \quad (19)$$

## 4 Results

This section evaluates the Bilateral, Trilateral, and Quadrilateral filters on different inputs. It highlights improvements and artifacts.

### 4.1 Signal input

Figure 3 demonstrates the distortion of the curved signal inputs by both Bilateral and Trilateral filters. The Quadrilateral filter, by design, preserves the quadratic functions ideally due to the use of Taylor's second-order approximation. It also performs nearly perfectly on the cubic function input, suggesting that the second-order approximation is sufficient and that higher-order terms of the Taylor series are not necessary.

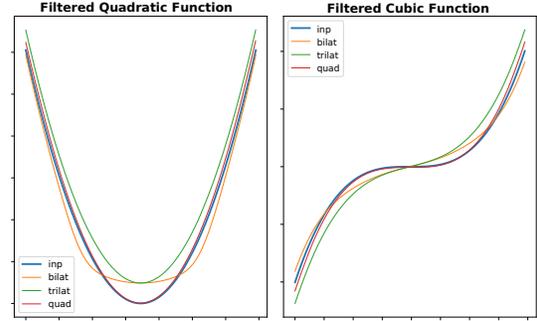


Figure 3: Quadratic and cubic functions as input signals. Quadrilateral preserves both shapes while the Trilateral and Bilateral distort them.

The Bilateral filter performs poorly at removing noise from the signal, as seen in Figures 4 and 5. This is particularly evident in high-gradient regions Figure 4(2), where the neighbouring values are too different in intensity and therefore discarded by the range kernel.

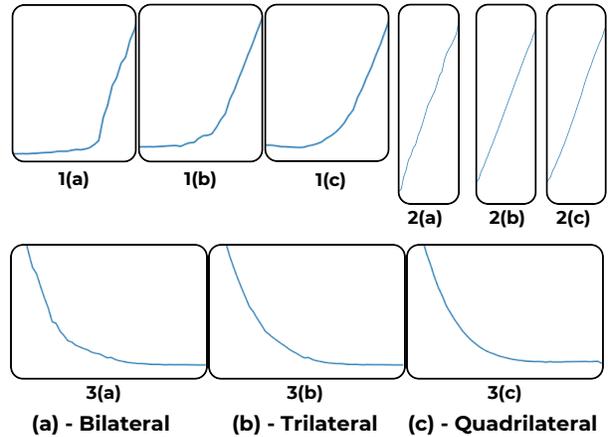


Figure 4: Zoomed in details from Appendix A. (1) - the Bilateral filter maintains a sharper edge. (2) - linear gradient area smoothed by Quadrilateral and Trilateral filters. (3) - Quadrilateral filter produces a smoother curve.

Both the Trilateral and Quadrilateral filters perform better at smoothing the signal. However, the Quadrilateral filter

shows an improvement, especially in regions where the gradient changes more rapidly, as shown in Figures 4(3) and 5(1).

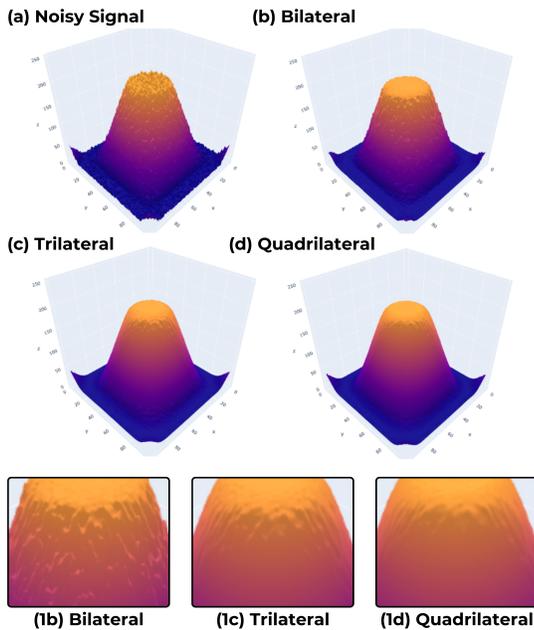


Figure 5: Filter comparison on 2-D signal, with a zoom-in on the curved region.

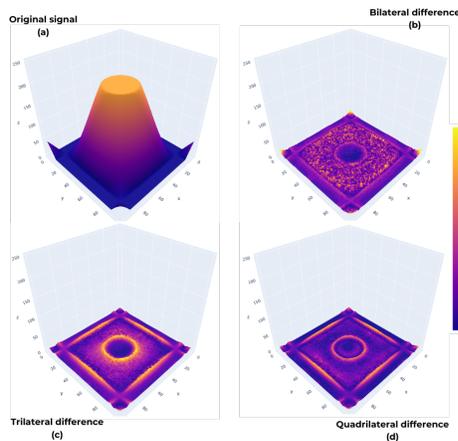


Figure 6: Difference between filtered noised signal and the original signal

Figure 6 highlights the patterns of the filters by comparing the filtered results with the reference signal prior to the applied Gaussian noise. The Bilateral filter successfully smoothes areas with constant intensity but fails to remove noise from high-gradient areas.

The Trilateral and Quadrilateral filters exhibit similar behaviour in this example, both effectively smoothing high-gradient areas, with the Quadrilateral filter achieving a result closer to the original signal. However, both Trilateral and Quadrilateral filters produce an undesired halo effect on the

perimeter of the graph and at the sharp cut-off at the top, Figure 6. The same artifact is observed in 4(1) where both filters are smoothing in sharp edge areas.

## 4.2 Images

In this section, a visual comparison is made of various images to demonstrate the properties of the filters. The following values of  $\sigma_s = 8$  and  $\sigma_r = 30$  are used to highlight the difference between the filters since they perform more similarly under smaller parameters.

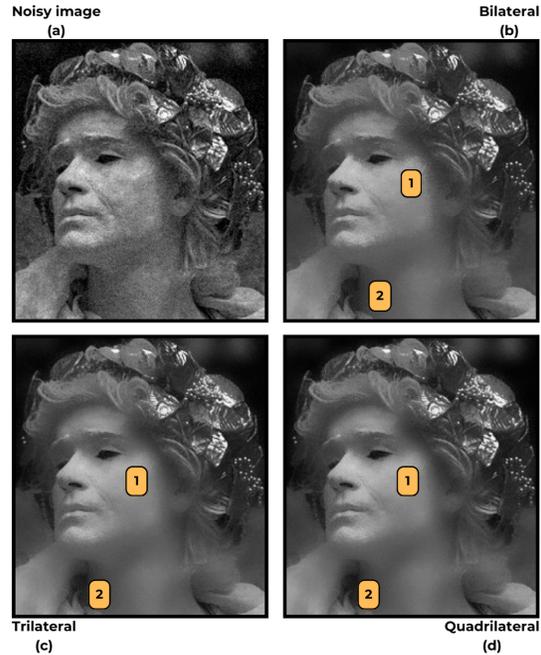


Figure 7: Applying filters on an image of a sculpture.

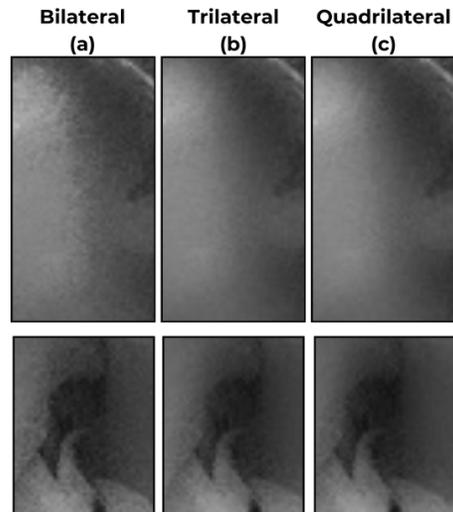


Figure 8: Highlights from Figure 7

All three filters preserve the details in the highly detailed

area in Figure 7, such as the hair and the leaves. However, in this case, both Trilateral and Quadriateral filters produce a smoother result. The main difference can be seen in the highlighted areas in Figure 8, where the Bilateral filter does not smooth these areas and leaves them noisy. The detail layer also shows regions where the Bilateral filter did not remove the noise as opposed to the other two filters, Figure 11(1).

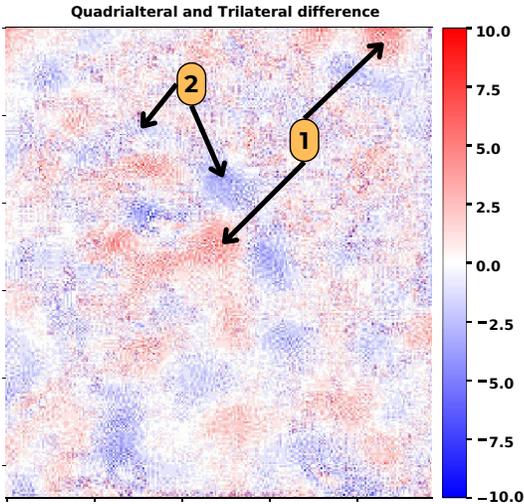


Figure 9: Difference between the Quadriateral and the Trilateral filters. (1) - the Quadriateral keeps originally bright areas in the input image brighter than the Trilateral. (2) - The Quadriateral filter keeps originally dark areas darker than the Trilateral filter.

However, as seen in Figure 10, the Quadriateral filter produces stronger halos on sharper edges than the Bilateral filter. This is caused by the attempt of the Quadriateral filter to approximate sharp edges with a smoother geometry. While these artifacts are not very visible in the filtered images, they are still undesirable.

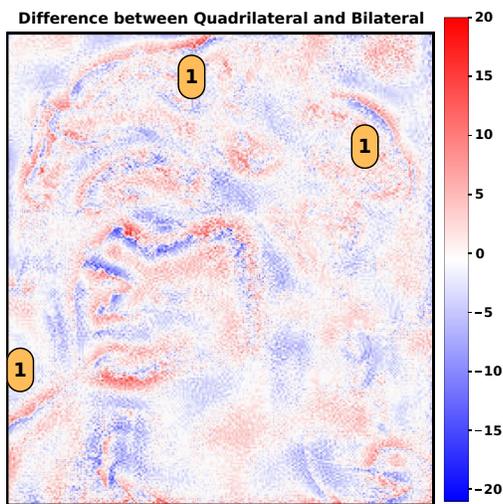


Figure 10: Difference between the Quadriateral and the Bilateral filters, highlighting halos.

The main difference between the Trilateral and the Quadriateral filters, as demonstrated in Figure 9, is the contrast in the produced image. The Quadriateral filter has darker, dark areas Figure 9(2) and brighter, bright areas 9(1). This is also visible in the detail layer, Figure 11, where the Trilateral filter darkens originally bright areas more than the Quadriateral filter. This effect is further confirmed by Figure 12, showing the increase in the image's contrast as the range sigma increases.

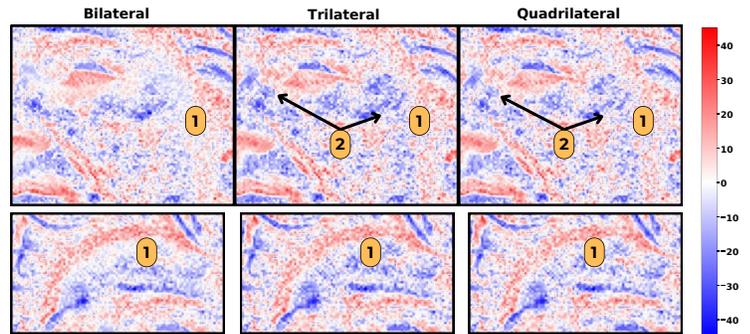


Figure 11: Highlights of a detail layer of the filters produced by subtracting filtered image from the noisy image. (1) - areas smoothed better by Quadriateral and Trilateral filters. (2) - originally bright areas that are darkened by the Trilateral filter more than by the Quadriateral. Appendix B shows the full detail layer.

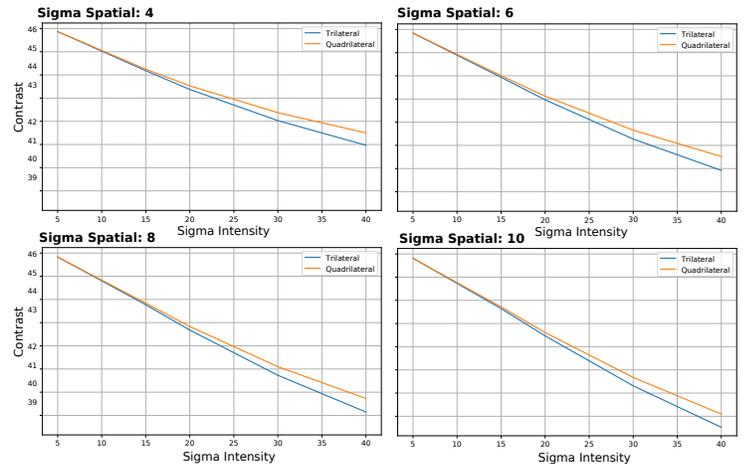


Figure 12: Image contrast of Trilateral and Quadriateral filters.

### 4.3 Interpolation

Uncertainty, discussed in Section 3.2, helps to identify regions where the quadratic surface approximation is not accurate to the neighbouring pixel intensity distribution. Figure 13 shows a noisy step function and an artifact produced by the Quadrilateral filter next to the edge caused by the quadratic surface approximation. Interpolation with the Bilateral filter resolves this issue.

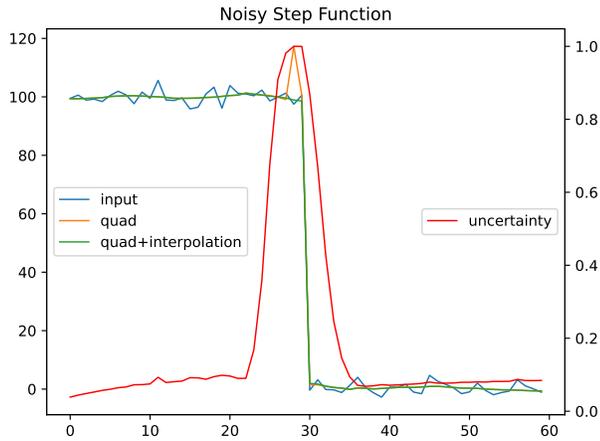


Figure 13: Contrast comparison of Trilateral and Quadrilateral filters.

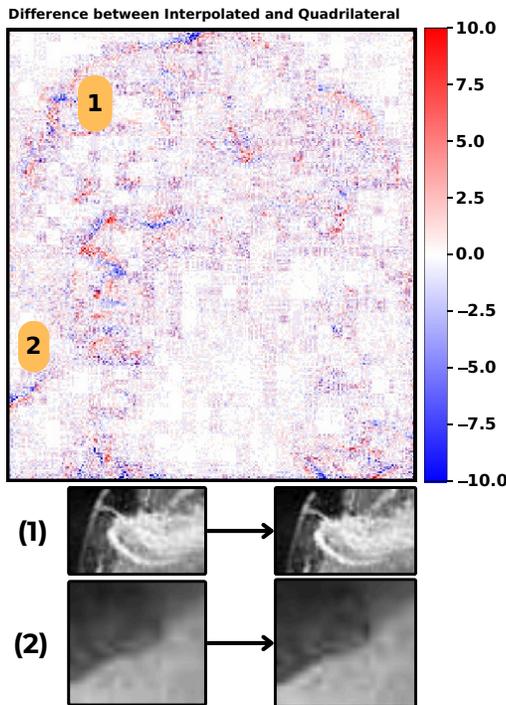


Figure 14: Difference between interpolated and Quadrilateral results and corresponding highlights from Appendix D.

Using the uncertainty map shown in Appendix C and interpolating with the Bilateral filter, the method successfully reduces the halo effect on an image, producing sharper edges, as seen in Figure 14.

### 4.4 HDR tone-mapping

This section discusses the tone reduction method proposed by Durand and Dorsey [3], which relies on a two-scale decomposition of an image into base and detail layers. In this approach, contrast reduction is applied only to the base layer, preserving the details.

The base layer is obtained by applying a de-noising filter to the intensity map of the input HDR image. After the contrast in the base layer is reduced, the detail layer, representing the removed "noise," is added back to the image.

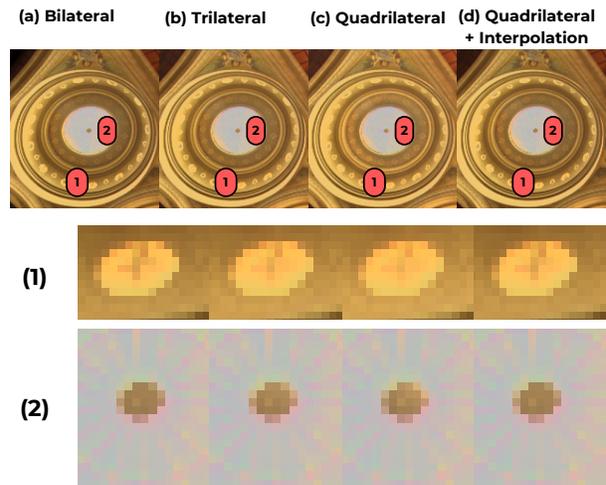


Figure 15: Highlights of a tone-mapped image. The overall brightness of the quadrilateral filter is higher than that of the other filters. (1) - preservation of small details. (2) - halos. The full image is in the Appendix E.

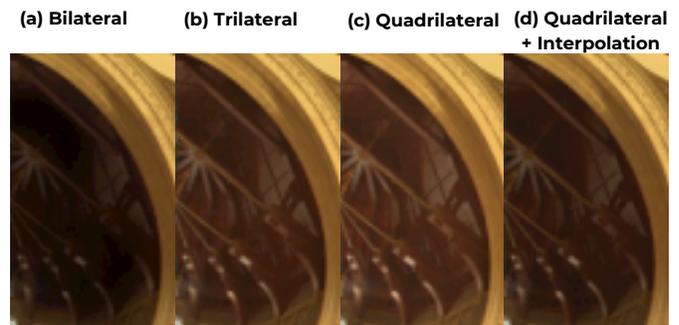


Figure 16: Highlight of a dark area of a tone-mapped image. The full image is in the Appendix E.

The Quadrilateral filter results in an overall bright and clear image, as seen in Figures 15 and 16. This can be explained by the closer alignment of the range kernel to the neighbouring pixel intensities distribution, resulting in dark areas less affected after filtering as seen in the detail layer, Figure 17.

However, as seen in Figure 15(c)(1), some details are lost due to the incorrect surface approximation. The Quadrilateral filter also produces halos, as seen in Figure 15(c)(2). Both issues are fixed by utilizing interpolation with the Bilateral filter. The interpolated result is the middle ground between the Bilateral filter’s detail sharpness and the Quadrilateral filter’s brightness in dark regions.

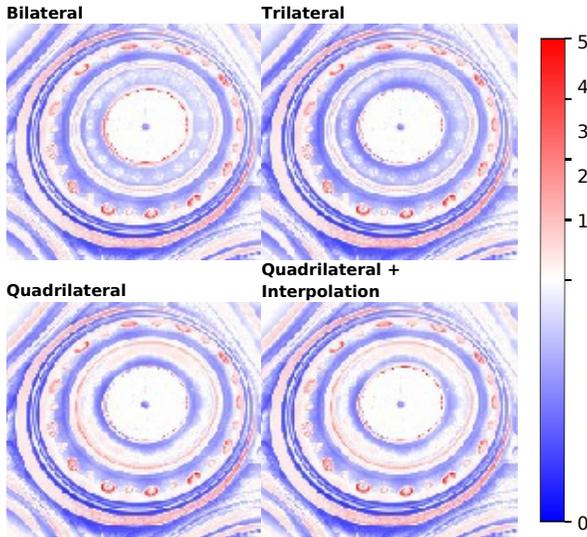


Figure 17: Detail layer extracted from a tone-mapped image, Appendix E.

#### 4.5 Computational performance

Table 1 shows the run-times of the Bilateral, Trilateral and Quadrilateral filters. Varying image size and kernel diameters are used. Computation of each derivative is essentially the pass of a bilateral filter, therefore, theoretically the Quadrilateral filter take around 6 times longer than the standard Bilateral filter. In practice, the Quadrilateral filter is around 3-4 times slower than the Bilateral filter.

These time samples measure the performance of non-optimized code and can be improved with techniques such as Fourier transform or sub-sampling mentioned in [3] and hardware-based acceleration methods [5], [1].

Image Size	Kernel size	Bilateral Time(s)	Trilateral Time(s)	Quadrilateral Time(s)
243 × 269	11	3.2	5.6	10.8
243 × 269	25	3.0	7.2	12.9
640 × 427	11	13.5	24.3	45.4
640 × 427	25	13.6	31.7	60.0
1024 × 768	11	41.53	70.21	131.26
1024 × 768	25	41.9	91.1	174.4

Table 1: Running time of each filter on different images with different kernel diameters.

## 5 Responsible Research

The study incorporates 1-D and 2-D signal examples featuring surfaces with constant intensities, as well as linear and non-linear gradients. This diverse set of signal characteristics ensures that the input is not tailored to the curvature-based filter, allowing for an unbiased evaluation of the filters’ performance. These examples serve as simplified yet effective representations to illustrate the filters’ capabilities.

The images chosen for testing show a wide range of properties, including high-detail areas, medium-detail areas, and areas with no detail. This selection provides a comprehensive evaluation of the filter’s performance across different types of content. Additionally, these images are commonly used in other filter-related research papers, allowing comparisons and validations of results.

To ensure the reproducibility of results, the implementation methods for both the Quadrilateral and Trilateral filters are thoroughly documented. The OpenCV implementation of the Bilateral filter is used for comparison. Documented code is provided, encompassing all relevant filters and visualization scripts, Appendix F. The parameters used in the experiments are also detailed, allowing for the replication of results and verification of findings. This approach not only validates the results but also contributes to the broader research community by providing a solid foundation for further studies.

## 6 Conclusions and Future Work

The Quadrilateral filter introduces an innovative edge-preserving smoothing technique that considers the curvature of pixel intensity distributions. This filter effectively processes N-dimensional input data, offering excellent noise reduction while preserving details. The Quadrilateral filter can smooth an image in high-gradient areas while preserving the original distribution of pixel intensities. The concept of uncertainty and interpolation with the Bilateral filter offers an even greater sharp-edge preservation. Its potential applications extend beyond image de-noising and tone mapping.

Future research could explore several ways to enhance the Quadrilateral filter. For instance, implementing a multi-stage interpolation process involving the Trilateral and Bilateral filters could achieve improved results. Additionally, employing a small-kernel Gaussian filter for interpolating individual anomalous pixels, as discussed in [3], could further refine the smoothing capabilities of the Quadrilateral filter.

Investigating the parameters involved in uncertainty calculation is another important area for future work. Developing methods for automatic parameter deduction from the input context, rather than relying on user-defined parameters, could greatly enhance the filter’s usability and effectiveness. Moreover, addressing the current computational efficiency challenges will be essential for achieving faster implementations, making the Quadrilateral filter more practical for real-time applications.

In summary, the Quadrilateral filter represents a significant advancement in edge-preserving smoothing techniques, with a potential for future improvements and applications.

## A Appendix: Test 1D signal

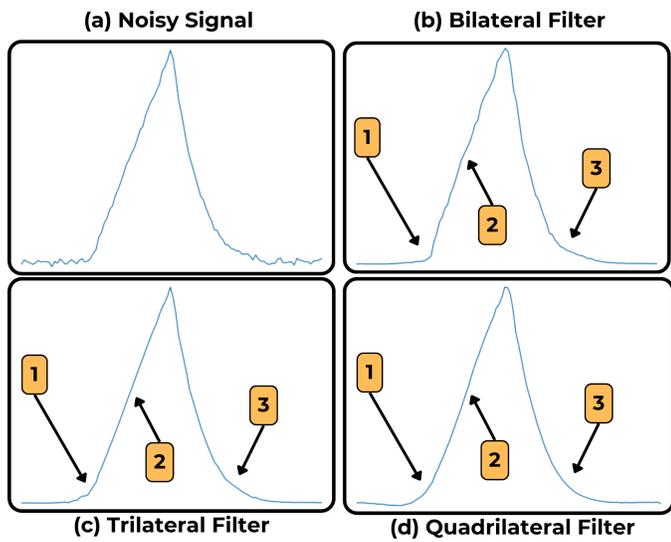


Figure 18: Filter comparison on 1-D signal containing constant intensity, linear gradient and a non-linear gradient.

## B Appendix: Detail layer

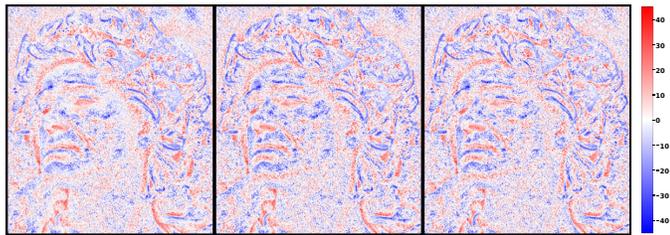


Figure 19: Detail layers of the Bilateral, Trilateral and Quadrilateral filters.

## C Appendix: Uncertainty Map

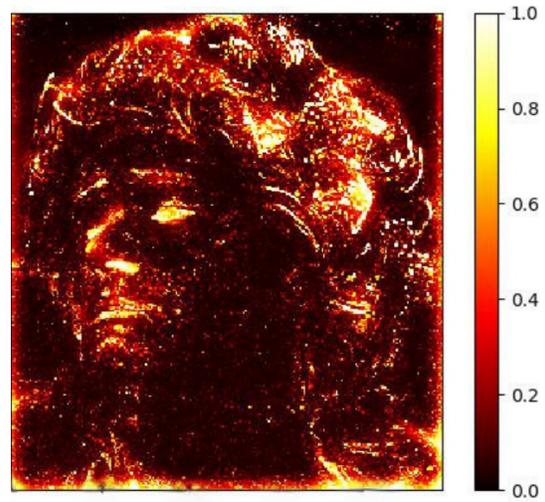


Figure 20: Uncertainty map of a sculpture image.

## D Appendix: Interpolated and non-Interpolated Quadrilateral results

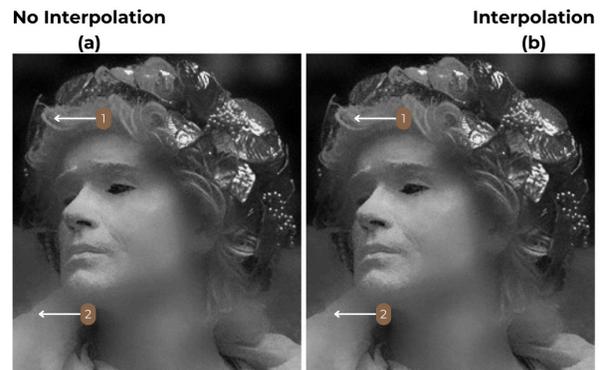


Figure 21: Original Quadrilateral and the one using Interpolation with the Bilateral filter.

## E Appendix: Memorial church tone-mapped

**Bilateral**

**Trilateral**



**Quadrilateral**

**Quadrilateral +  
Interpolation**

Figure 22: Tone-mapped image of the memorial church. Parameters used: Spatial sigma - 8, Range sigma - 0.5.

## F Appendix: Code repository

GitHub repository - <https://github.com/dmytroMaksymchuk/QuadrilateralFilter>

## References

- [1] E Bethel. Exploration of optimization options for increasing performance of a gpu implementation of a three-dimensional bilateral filter. 2012.
- [2] Prasun Choudhury and Jack Tumblin. The trilateral filter for high contrast images and meshes. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, page 5–es, New York, NY, USA, 2005. Association for Computing Machinery.
- [3] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high - dynamic - range images. *ACM Trans. Graph.*, 21:257–266, 07 2002.
- [4] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics (Proceedings of Siggraph Conference)*, 23, 08 2004.
- [5] Shraddha Oza and Kalyani R. Joshi. Cuda based fast bilateral filter for medical imaging. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 930–935, 2018.
- [6] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, 1998.
- [7] Jin Wang, Zhensen Wu, and Jiaji Wu. Efficient adaptive deinterlacing algorithm using bilateral filter. volume 61, page 02021, 01 2016.
- [8] Holger Winnemoeller, Sven Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Trans. Graph.*, 25:1221–1226, 07 2006.
- [9] Lei Yang, Pedro Sander, and Jason Lawrence. Geometry-aware framebuffer level of detail. *Comput. Graph. Forum*, 27:1183–1188, 06 2008.