JDSSV

Journal of Data Science, Statistics, and Visualisation

*April 2024, Volume IV, Issue III.* *doi: 10.52933/jdssv.v4i3.74*

# An Edge Preserving Median Filter For Images Based On Level-sets

**Jean-Pierre Stander**
University of Pretoria

**Inger Fabris-Rotelli**
University of Pretoria

**Theodor Loots**
University of Pretoria

**Johan M van Niekerk**
University of Twente

**Alfred Stein**
University of Twente

## Abstract

Edges in images define boundaries of objects which are important for various tasks; thus, edges are crucial components in an image. Noise can be introduced to images through various sources which can compromise the integrity of the edges and other component in the image. We propose an edge preserving median filter, called the level-set adaptive median filter, for noise removal in images. This filter uses connected sets of pixels with the same value, namely level-sets, as flexible regions which contour to edges in the image. The filter determines whether a set is noise or signal and it smooths the noise. These set regions are flexible in terms of shape since they are created based on their values, and being data-driven therefore provide the mechanism for the filter to preserve edges in the image. To validate our approach we used metrics such as Pratt's Figure of Merit and Peak-Signal-to-Noise Ratio on the labelled faces in the wild data set,[*] which is a widely used collection of photographs of faces usually used for studying facial recognition, to assess the algorithm's performance. We concluded that the proposed level-set adaptive median filter does remove noise while preserving the edges in the image better than the traditional adaptive median filter.

*Keywords*: noise removal, image filter, median filter, adaptive median filter, level-sets.

[*] see https://vis-www.cs.umass.edu/lfw/

# 1. Introduction

Noise in an image is the occurrence of random variation of the pixel intensities and obscures important information in the image that is necessary for image understanding. This results in pixels having a different pixel intensity than the true intensity. Noise can be caused by various factors such as lighting, dust particles on the camera lens, or the digitisation process itself (Verma and Ali 2013). Different types of noise that can occur in images include impulse, additive and multiplicative noise (Verma and Ali 2013; Patidar et al. 2010). Therefore, noise removal plays a significant role in image processing since it aims to recapture the true image and should take place prior to further modelling (Peters 1995).

Jain et al. (1995) defined a region as a set of connected pixels that share similar properties. Objects in an image consist of one or more regions, and regions are typically separated by edges. An edge in an image can thus be defined as a significant local change. It is important that a noise removal filter applied to an image preserves the edges while removing noise (Patidar et al. 2010). Preserving the edges in an image is important because the loss of edges can lead to an incorrect representation of the true image.

Numerous noise removal filters have been developed. These algorithms can be classified into linear and non-linear filters. An example of a simple linear filter is the mean filter (Al-Amri et al. 2010). Non-linear filters have also been developed (Hwang and Haddad 1995; Wang et al. 2010). These types of filters are more robust i.e., the output is less affected by small changes in the input. Non-linear filters also preserve edges in images better than linear filters (Patidar et al. 2010). These filters along with two-dimensional structuring elements have been used to construct filters to smooth and remove noise from images (Dougherty and Astola 1994).

The median filter is a non-linear filter that was introduced by Tukey in 1977 for smoothing time series data (Tukey 1977). Since then median filtering has been used to smooth and filter noise from digital images (Verma et al. 2015; Lin and Willson 1988). The simplicity of the median filter is attractive. The median filter uses a neighbourhood around the pixel of interest and replaces it with the median pixel value of that neighbourhood if the pixel is noise-contaminated (Justusson 1981). The median of the set is a better measure of pixel intensity of the set around the pixel of interest since it is more robust to outliers, namely other neighbouring pixels. For this reason, the median filter is a popular technique for noise removal. In literature, the median filter is used extensively as it outperforms linear filtering when denoising images where the object in the image has edges (Verma et al. 2015; Arias-Castro and Donoho 2009).

The traditional median filter although powerful, does have some shortcomings. It does not provide sufficient smoothing of non-impulsive noise and struggles with removing impulse noise which occurs with high probability (Hwang and Haddad 1995). Hwang and Haddad (1995) extended the median filter to make use of windows of variable size. This improved the algorithm's capability to smooth non-impulse noise and impulse noise occurring with high probability. In this adaptive median filter, a square window of increasing size is defined for each pixel of the image which is then used as the neighbourhood. Albeit the windows can vary in size, the shape of the window is rigid. This extension further includes a detection step in which the algorithm determines whether the pixel is noise-contaminated or not, and then replaces this value with the median value if the pixel is noise-contaminated (Hwang and Haddad 1995).

Much work has since gone into advancing adaptive median filters such as combining it with partition clustering (Pang et al. 2018), fuzzy switching median (Singh and Maheswari 2017) and hybrid versions (Mustafa et al. 2012). We investigate an alternative involving level-sets.

Level-sets have been used in applications such as image segmentation with colour, shape movement, and texture (Cremers et al. 2007). The reason for using level-sets are, to name some, their flexibility, ease of use, and a priori information (Wang et al. 2021).

We propose a level-set adaptive median filter with flexible, data-driven window shapes. Level-sets in an image are a collection of pixels that have the same pixel intensity and each pixel in the collection is the neighbour of at least one other pixel. The proposed filter uses the idea of level-sets to choose more flexible neighbourhood areas. These sets of pixels are naturally closely related to each other and therefore it provides more relevant data to determine if a specific pixel or set of pixels is noise-contaminated. It also replaces contaminated pixels with more informative values. As level-sets are sets of connected pixels with the same pixel intensity, it can be said that if one of these pixels is classified as noise the rest of the set is also noise. These level-sets are flexible since a pixel only has to be connected to one pixel in the set to be included in the set. The flexibility of level-sets forms a basis on which a strong and adaptive filter can be built. This new filter also allows for edge preservation.

In Section 2, the proposed filter is presented and some of its properties are discussed. In Section 3, edge preservation and its importance is discussed. In Section 4, an implementation of the algorithm on the labelled faces in the wild data set is shown and compared to existing filters. In Section 5 the results of the application are discussed.

# 2. Level-set adaptive median filter

Consider an image $f$ with pixel intensities $\{f(x_{ij})\}$ on $\mathcal{A}(\mathbb{Z}^2)$ where $\mathcal{A}(\cdot)$ is a vector lattice and $\mathbb{Z}$ is the set of integers. The pixel intensities $f(\cdot)$ are defined on a range $[0, n]$ for some $n$[1]. The connectivities of pixels in the image $f$ are based on the definition for a morphological connection as defined in Definition 1 (Serra 2006). A morphological connection implies the elements (pixels) of a set are connected if the elements touches or overlaps; in image processing this is typically quantified by pixel adjacency such as 4- or 8-connectivity. For the purpose of this work, the connected pixels are determined by using 8-connectivity. In 8-connectivity, a pixel is connected to its immediate neighbours, which include the four pixels that are directly adjacent (above, below, left, right) as well as the four diagonal pixels (for instance, top-right or bottom-left). This results in eight connected pixels surrounding each central pixel. The reason for using 8-connectivity is that, since it is using the eight pixels around it, it closely aligns with the variable square-shaped window used in the adaptive median filter. In general, any connectivity can be used provided it satisfies Definition 1.

[1] This is usually integers between 0 and 255 for grayscale. It can also be defined as real numbers ranging between 0 and 1.

**Definition 1.** *A connection (Serra 2006)*
*Let $B$ be a non-empty set. A family $\mathcal{C}$ of subsets of $B$ is a connection if:*

1. $\emptyset \in \mathcal{C}$,
2. $\{x\} \in \mathcal{C}$ *for each* $x \in B$ *and*
3. *for any family* $\{C_i, i \in I\} \subseteq \mathcal{C}$ *we have*

$$\bigcap_{i \in I} C_i \neq \emptyset \implies \bigcup_{i \in I} C_i \in \mathcal{C}.$$

*Then any $C_i \in \mathcal{C}$ is called a connected set.*

Based on Definition 1, any set $C$ of connected class $\mathcal{C}$ is connected. In the context of any image, the pixels make up the non-empty set in this case. Definition 1 (1) states that any empty set is connected i.e., a set of no pixels is seen as being connected. Definition 1 (2) state that a set containing only one element is connected, i.e., a set containing only 1 pixel is a connected set. Definition 1 (3) states that if groups that are by themselves connected share an element, they can be combined to create a larger group which will still be connected. For example, if in a image we have a group of pixels $\mathcal{C}_1$ and $\mathcal{C}_2$ and the element $x_i$ is in both these connected groups, $\mathcal{C}_1$ and $\mathcal{C}_2$ can be combined into a one single group which will still be a connected set. A set of pixels which satisfy the criteria of Definition 1 is called a morphological connection. This paper is focusing on the image we will be referencing to a set of connected pixel, since the set we are considering here is one of pixels. The set of connected pixels of size $n+1$ that includes the pixel $x$ for any $x \in \mathbb{Z}^2$ and $n \in \mathbb{N}$ is defined as $\mathcal{N}_n(x)$ (Anguelov and Fabris-Rotelli 2010). This is also known as the neighbourhood of pixel $x$.

Level-sets in a gray-scale image are sets of connected pixels, using the connection definition from Definition 2 of constant value.

**Definition 2.** *A level-set (Braga-Neto and Goutsias 2004)*
*Consider an image defined by $f$ on $A(\mathbb{Z}^2)$ and a non-empty set of pixels $C \in \mathcal{C}$ where $\mathcal{C}$ is a connection. If,*

1. *$C$ is a connected set of pixels as defined in Definition 1.*
2. $f(x) = k \;\forall\; x \in C$ *for some constant* $k$,

*then $C$ is a level-set.*

The level-sets of size $p$, for our proposed filter, are defined as,

$$J^{(p)} = \{V : V \in \mathcal{C}, f(x) = f(y) \forall x, y \in V, card(V) = p\},$$

where $\mathcal{C}$ is a connectivity class (Serra 2006). Therefore $J^{(p)}$ are sets of size $p$ where all the elements in the set have the same value and are all connected based on the chosen connectivity $\mathcal{C}$.

The level-set adaptive median filter is applied to each level-set size sequentially in the image i.e., $J^{(p)}$ for $p = 1, \ldots, p_{\max}$. The algorithm has two hyperparameters namely, $p_{\max}$ and $n_{\max}$, where $p_{\max}$ is the maximum size of the level-sets that will be investigated, and $n$ is the size of the neighbourhood which is increased at each step if the level-set is determined to be signal rather than noise, up to a maximum of $n_{\max}$. The parameter $n_{\max}$ plays the same role as the increasing window size in the original adaptive median filer.

With the adaptive median filter, we define a window $S_{ij}$ centred at $x_{ij}$ as $S_{ij} = \{x_{i-u,j-v} : -L \leq u \leq L, -L \leq v \leq L\}$ resulting in a window of size $(2L+1) \times (2L+1)$, where $(2L+1) \leq w_{\max}$ and where $w_{\max}$ is the hyperparameter defining the maximum window size to be considered. Herein, the proposed level-set adaptive median filter defines this window as the neighbours of the level-sets, $V \in J^{(p)}$, instead of the usual square window, $S_{ij}$, around pixel $x$. This data-driven window is defined as

$$B_V^{(n)} = \bigcup_{(i,j) \in V \in J^{(p)}} \bigcup_{K \in \mathcal{N}_n(i,j)} K, \text{ for } n = 1, ..., n_{\max}.$$

Therefore, $B_V^{(n)}$ is the union of the level-sets $V \in J^{(p)}$ under consideration and the neighbourhood of size $n$ of each pixel $x_{ij}$ in $V$. An example of this set is shown in Figure 1(b) for $B_V^{(1)}$ and Figure 1(c) for $B_V^{(2)}$. It can be seen that the window adapts to the shape of the level-set which distinguishes it from the traditional adaptive median filter that uses a square window. By using these flexible windows, the filter is able to adapt to the underlying structures in the image. The shape of the region $B_V^{(n)}$ is dependent on the definition of connectivity. Since the pixel intensities of image $f$ are defined as $\{f(x_{ij})\}$, $f_{B_V^{(n)}}$ is the set of pixel intensities of the set $B_V^{(n)}$. Therefore, $f_{B_V^{(n)}}$ is a subset of the original image with the level-set and all its neighbours.

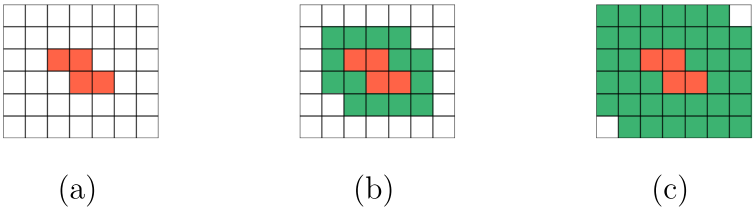


Figure 1: (a) $V \in J^{(3)}$: Level-set of size 3. (b) $B_V^{(1)}$: The neighbourhood of $V$ for $n = 1$. (c) $B_V^{(2)}$: The neighbourhood of $V$ for $n = 2$.

The algorithm for the level-set adaptive median filter is provided in Algorithm 1. The algorithm starts by determining all the level-sets in the image. Level-set size 1 up to $p_{\max}$ are investigated. It is determined whether the level-set is noise-contaminated or not by looking at a neighbourhood of size 1 to $n_{\max}$ of the level-set of interest. If a set is determined to be noise-contaminated, it is smoothed.

To decide if the set of interest is noise that should be smoothed or not, as well as the value to smooth it to, follows the same logic as the traditional adaptive median filter. This follows a two-step approach namely noise detection and noise reduction. In the first step, it checks if the set is likely to be noise based on the range of its neighbourhood. If it falls outside this range the set is more likely to be corrupted. In

**Algorithm 1** Level-set adaptive median filter.

```
Input: $f$: The images to be smoothed
       $n_{\max}$: The maximum window size
       $p_{\max}$: The maximum level-set size to be smoothed
Output: $f$: The smoothed image
Procedure LevelSetAdaptiveMedian($f$, $n_{\max}$, $p_{\max}$):
    for $p = 1, \ldots, p_{\max}$ do
        Obtain $J^{(p)}$
        if $J^{(p)} \neq \emptyset$ then
            for each $V \in J^{(p)}$ do
                set $n = 1$
                while $n < n_{\max}$ do
                    Obtain $B_V^{(n)}$
                    $f^{smth}$, $n$ = SmoothNoise($f_{B_V^{(n)}}$, $V$, $n$, $n_{\max}$)
                    $f_{ij} = f^{smth} \; \forall \, (i,j) \in V$
    return $f$
```

the second step, if the set is determined to be noise, it is corrected by replacing this value with the median of the set's neighbourhood. This is shown in Algorithm 2.

## 2.1. Edge Strength

We next define a measure of strength for the local change required for an edge definition. An edge is defined as a boundary between two regions of different constant gray level (Davis 1975). In the example below, a simple edge is shown. For this simple example we compare the regions on either side of the edge. Later herein, we make use of a metric which measures the preservation/strength of all edges in an image rather than for a single edge. Consider Figure 2. The original image has a definite, strong edge. The image is divided into two regions, one on each side of the edge.

Let the within-region variation be represented by $\delta$ and the between-region variation by $\gamma$. Then $\delta$ and $\gamma$ are defined for regions $R_1$ and $R_2$ on either side of an edge are as follows,

$$\begin{aligned}\delta &= \sum_{i=1}^{2} \sum_{x \in R_i} (x - \bar{x}_{R_i})^2, \\ \gamma &= \sum_{i=1}^{2} n_i (\bar{x}_{R_i} - \bar{x})^2,\end{aligned}$$

where $\bar{x}_{R_1}$ and $\bar{x}_{R_2}$ denote the mean pixel values in the regions $R_1$ and $R_2$ respectively, $n_i$ denotes the number of elements in region $R_i$ and $\bar{x}$ the overall mean. In general, regions $R_1$ and $R_2$ are chosen as the neighbourhoods of similar pixels on either side of the edge, determined appropriately by the application under consideration.

An edge is clear when $\delta$ is small enough and $\gamma$ is large enough. We make use of the ratio of within and between region variation,

$$\beta = \frac{\delta}{\gamma}. \tag{1}$$

**Algorithm 2** Detect and Reduce noise.

```
Input: $f_{B_V^{(n)}} = \{f_{ij} : (i,j) \in B_V^{(n)}\}$: The set of pixels to be smoothed and its neighbours.
       $V$: The pixel intensity value of the set.
       $n$: The current window size.
       $n_{\max}$: The maximum window size.
Output: $f^{smth}$: The smoothed set.
        $n$: The adapted windows size.
Function SmoothNoise($f_{B_V^{(n)}}$, $V$, $n$, $n_{\max}$):
    Determine $\min(f_{B_V^{(n)}})$, $\text{med}(f_{B_V^{(n)}})$ and $\max(f_{B_V^{(n)}})$

    if $\min(f_{B_V^{(n)}}) < med(f_{B_V^{(n)}}) < \max(f_{B_V^{(n)}})$ then
        if $\min(f_{B_V^{(n)}}) < V < \max(f_{B_V^{(n)}})$ then
            $f^{smth} = V$
        else
            $f^{smth} = \text{med}(f_{B_V^{(n)}})$
        $n = n_{\max}$
    else
        $f^{smth} = \text{med}(f_{B_V^{(n)}})$
        $n = n + 1$
    return $f^{smth}$, $n$
```

We refer to $\beta$ as the edge strength. In Figure 2, the within-region variation $\delta$ increases from the left to the right. The between-region variation $\gamma$ decreases from the top to the bottom. It is visible that as the edge strength, $\beta$ decreases, the clearer the edge becomes.

To show that the algorithm proposed herein will at worst keep the edge strength the same and at best result in stronger edges, the within- and between-region variation will be considered. The greater the between variation and smaller the within variation of two regions on either side of an edge, the stronger the edge is, as seen in Figure 2.

When the filter is applied to an image the within-region variation will remain the same or decrease. Similarly, the between-region variation will remain the same or increase. This is shown in Proposition 1. For the proof of Proposition 1, see the Appendix.

**Proposition 1.** *Consider and image $f$. Let $L : \mathcal{A}(\mathbb{Z}^2) \to \mathcal{A}(\mathbb{Z}^2)$ be the level-sets adaptive median filter such that*

$$f_{smoothed} = L(f)$$

*The level-set adaptive median filter results in*

$$\delta(f_{smoothed}) \leq \delta(f)$$

*and,*

$$\gamma(f_{smoothed}) \geq \gamma(f)$$

*and,*

$$\beta(f_{smoothed}) \leq \beta(f)$$

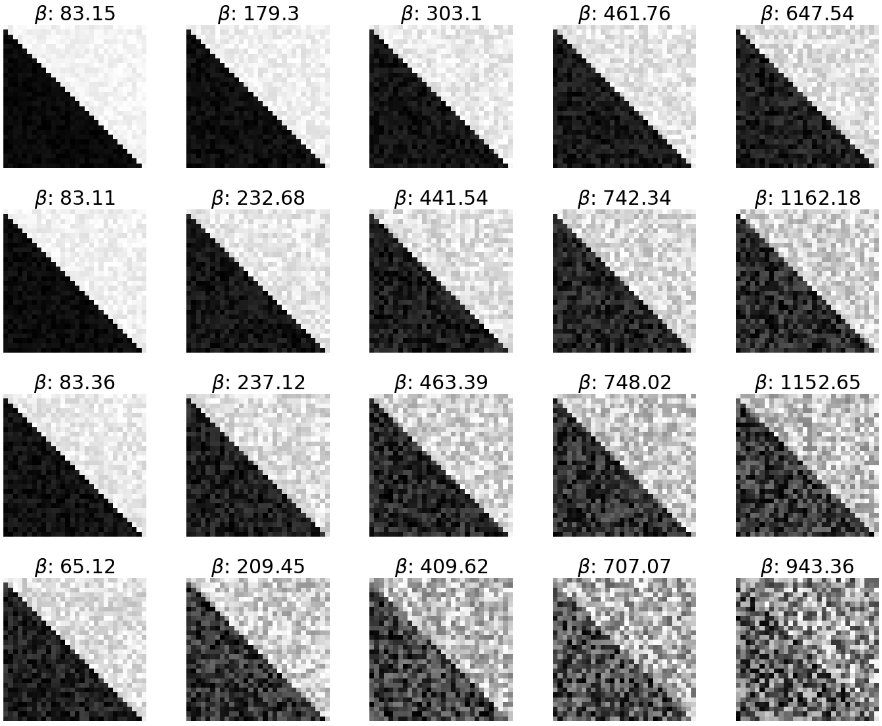


Figure 2: Examples of edges for different combinations of high and low between and within region variation. The within-region variation increases from the left to the right. The between-region variation decreases from the top to the bottom. Here $\beta$ is as in Equation 1

*where $f_{smoothed}$ is the resulting image after applying the level-set adaptive median filter to image $f$.*

# 3. Edge preservation

Pratt's Figure of Merit (Yu and Acton 2002) is used to evaluate the preservation of edges after the image is smoothed. This index is calculated as

$$FOM = \frac{1}{\max(N_{original}, N_{smoothed})} \sum_{i=1}^{N_{smoothed}} \frac{1}{1+\frac{1}{9}d_i^2},$$

where $N_{original}$ is the number of edge points detected in the original noise-free image, $N_{smoothed}$ the number of edge points detected after the noisy image is smoothed, $d_i^2$ is the Euclidean distance between the $i^{th}$ edge point in the original image and its closest edge point in the smoothed image. This index is bounded between 0 and 1, where 1 is perfect edge preservation and 0 is no edge preservation.

To investigate the edge-preserving ability of the level-set adaptive median filter compared to the adaptive median filter, we simulate an edge, see Figure 3(a). Salt and pepper noise is randomly added to 10%, 20% and 30% of the image. Both the adaptive median and level-set adaptive median filters are then applied to this noisy image and

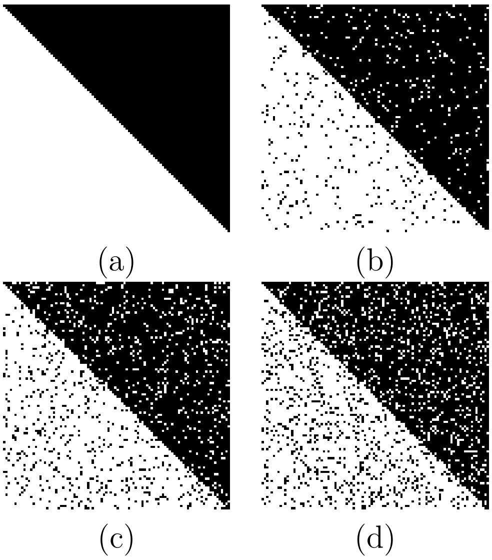


Figure 3: (a) Simple example of an edge, contaminated with (b) 10% (c) 20% (d) 30% salt and pepper noise.

the FOM is calculated to evaluate the edge preservation under various speckle noise strengths.

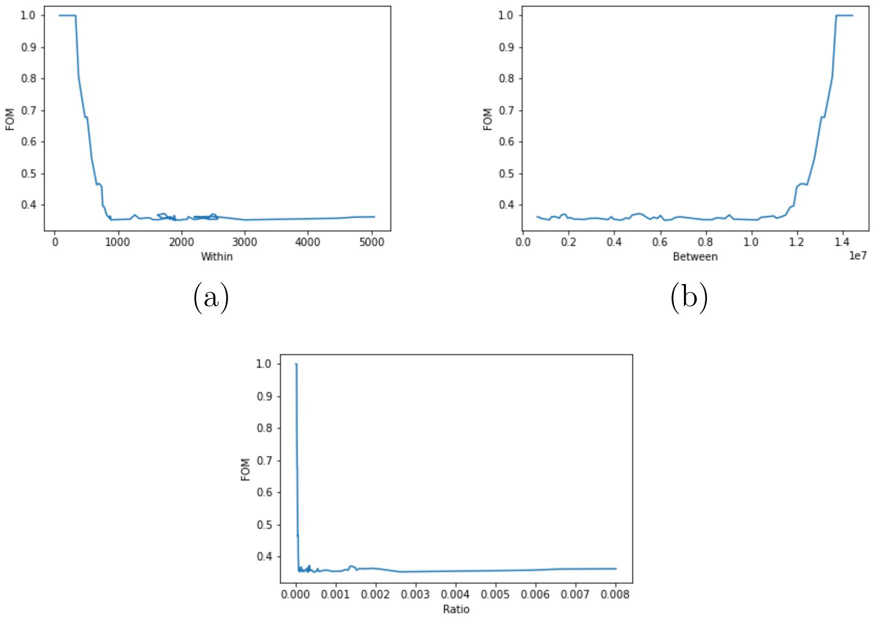
The simulation is repeated 100 times. The means of the FOM values for the 100 repetitions are shown in Table 1. The results are shown for the level-set adaptive median filter for $p_{\max} = 2, 3, \ldots, 10$, and for the original adaptive median filter for $w_{\max} = 2, 3, \ldots, 10$.

Table 1: The FOM values comparing the performance of the adaptive median (AM) filter against the level-set adaptive median (LS-AM) filter for different hyperparameters. This is done for 10%, 20% and 30% simulated Salt and Pepper noise.

| $p_{\max}$/ $w_{\max}$ | $\text{AM}_{10\%}$ | LS $\text{AM}_{10\%}$ | $\text{AM}_{20\%}$ | LS $\text{AM}_{20\%}$ | $\text{AM}_{30\%}$ | LS $\text{AM}_{30\%}$ |
|---|---|---|---|---|---|---|
| 2 | **0.94** | 0.87 | **0.75** | 0.51 | **0.46** | 0.29 |
| 3 | **0.94** | 0.91 | **0.75** | 0.74 | **0.45** | 0.43 |
| 4 | **0.94** | 0.92 | 0.75 | **0.84** | 0.45 | **0.58** |
| 5 | **0.94** | 0.91 | 0.80 | **0.85** | 0.67 | **0.69** |
| 6 | **0.94** | 0.91 | 0.80 | **0.86** | 0.66 | **0.76** |
| 7 | 0.91 | **0.92** | 0.73 | **0.85** | 0.58 | **0.80** |
| 8 | **0.91** | **0.91** | 0.73 | **0.86** | 0.58 | **0.81** |
| 9 | 0.89 | **0.91** | 0.64 | **0.85** | 0.52 | **0.81** |
| 10 | 0.88 | **0.91** | 0.65 | **0.86** | 0.52 | **0.81** |

From Table 1 it can be seen that the level-set adaptive median filter overall performs better than the adaptive median filter in terms of edge preservation. It can also be noted that the level-set adaptive median filter performs better for larger values of $p_{\max}$.

To further investigate the relationship between FOM and $\gamma$, $\delta$ and $\beta$, a simulated an image with a strong edge is presented with the noise increased systematically, similar to Figure 2. After the image is smoothed, the within (Figure 4(a)) and between (Figure 4(b)) region variation as well as the ratio (Figure 4(c)) of within and between region variation are calculated and plotted against the FOM of the smoothed image. It can be seen that the smaller $\delta$ and $\beta$ are and the larger $\gamma$ is the stronger the edge.



(a)

(b)

(c)

Figure 4: The (a) within- and (b) between-region variation and (c) the ratio plotted against the FOM of an edge where noise was added systematically (see Figure 2)

## 3.1. Hyperparameter effect on edge preservation

The level-set adaptive median filter has two hyperparameters namely $p_{\max}$ and $n_{\max}$. To investigate the effect of these two parameters, we simulate images of different sizes with a clear edge as in Figure 2. The images were then contaminated with three types of noise namely 20% salt and pepper noise, noise from a Gaussian distribution with $\mu = 0$ and $\sigma^2 = 20$ and noise from a Gumbel distribution with $\mu = 0$ and $\beta = 20$. We use salt and pepper noise as this type of noise commonly occurs in images. Gaussian and Gumbel noise are also used to see the performance on symmetrical as well as asymmetrical noise. The level-set adaptive median filter is applied to the simulated images using various combinations of $n_{\max}$ and $p_{\max}$ values. Table 2 contains the combination of hyperparameter values that resulted in the highest FOM value for each image size.

During the investigations, it was found that choosing $n_{\max}$ as 2 and $p_{\max}$ as 3 generally yielded better results. From Table 2 it seems that the larger the image the larger $p_{\max}$ should be for better results. Although in larger images, a larger value for $p_{\max}$ can be used, using $n_{\max}$ as 2 and $p_{\max}$ as 3 are good starting parameters. In larger images, sometimes a higher value for $n_{\max}$ yielded better results but more often than not 2 was a better choice regardless of image size. The filter smooths based on the underlying structures in the image, and therefore it makes sense that the parameters increase since the underlying structure in a larger image will be made up of larger level-sets as well

Table 2: The FOM values for three different noise types, Salt and Pepper (S&P), Gaussian, and Gumbel noise. The results of the level-set adaptive median filter are shown for images of various sizes with the best-performing hyperparameter combination.

| | S&P noise | | | Gaussian noise | | | Gumbel noise | | |
|---|---|---|---|---|---|---|---|---|---|
| Image size | $n_{\max}$ | $p_{\max}$ | FOM | $n_{\max}$ | $p_{\max}$ | FOM | $n_{\max}$ | $p_{\max}$ | FOM |
| 10×10 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 4 | 0.85 |
| 20×20 | 3 | 4 | 0.92 | 3 | 1 | 0.58 | 3 | 3 | 0.56 |
| 128×128 | 2 | 6 | 0.90 | 4 | 5 | 0.12 | 2 | 8 | 0.13 |
| 256×256 | 4 | 6 | 0.89 | 3 | 7 | 0.06 | 4 | 8 | 0.07 |
| 512×512 | 2 | 8 | 0.87 | 2 | 3 | 0.03 | 2 | 2 | 0.03 |

as more level-sets. Therefore, we recommend when using this algorithm to start with $n_{\max}$ as 2 and $p_{\max}$ as 3.

## 4. Application

The proposed level-set adaptive median filter (LS-AM) and the adaptive median filter (AM) are compared on natural images. We make use of the Labelled Faces in the Wild (LFW) dataset (Huang et al. 2007). Edges play a significant role in the features of faces. The fact that edges are important in images of faces is the reason why this dataset was chosen. This dataset consists of 13,233 images each of size 250×250 of faces collected from the internet in an unconstrained manner (Huang et al. 2008).

Three types of noise are randomly added to each of the images at two levels of intensity, 10% and 20% salt and pepper noise (impulsive noise), noise from a Gaussian (symmetrical noise) with $\mu = 0$ and $\sigma^2$=10 and 20 and noise from a Gumbel (asymmetrical noise) with $\mu = 0$ and $\beta$=10 and 20. The AM filter and proposed LS-AM filter are then applied to these noise-contaminated images. An example of an image from the dataset as well as an example of the noise added and the result of the AM and LS-AM can be seen in Figures 11 - 13 in the Appendix.

The performance of the filters is compared by using the peak signal-to-noise ratio (PSNR) (Huynh-Thu and Ghanbari 2008), defined as,

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{NM}\sum_i\sum_j(o_{ij}-s_{ij})^2}\right), \tag{2}$$

where $o_{ij}$ is the pixel intensity of the original images (before the noise was added) and $s_{ij}$ is the pixel intensity of the smoothed images (after noise was added and the filter applied) at pixel $(i, j)$. The PSNR values are calculated before and after the noisy image is smoothed. An increase (decrease) in the PSNR value indicates that the number of pixels containing signal increased (decreased) relative to the number of pixels containing noise.

In Table 3, various PSNR values are calculated for comparison. The noisy PSNR average is the average PSNR value calculated on all images after being contaminated

Table 3: The average PSNR values before and after smoothing the noisy images from the LFW dataset with both the LS-AM and AM filters.

| **Noise type** | **Noise level** | **Noisy PSNR Average** | **Smooth PSNR Average** | |
|---|---|---|---|---|
| | | | **AM** | **LS-AM** |
| Gaussian | $N(0, 10^2)$ | 28.38 | 30.55 | **31.19** |
| Gaussian | $N(0, 20^2)$ | 22.54 | 25.23 | **26.20** |
| Gumbel | $\text{Gum}_{(0,10)}(\mu, \beta)$ | 25.36 | 28.03 | **28.69** |
| Gumbel | $\text{Gum}_{(0,20)}(\mu, \beta)$ | 19.56 | 22.55 | **23.36** |
| Salt & Pepper | 10% | 14.76 | 38.87 | **40.33** |
| Salt & Pepper | 20% | 11.75 | 33.82 | **36.82** |

with noise. The smooth PSNR average is the average PSNR values calculated on the images after being smoothed. If the average PSNR value increases from before to after smoothing, it indicates that the number of pixels containing signal is increased relative to the number of pixels containing noise.

Table 3 presents the results on the labelled faces in the wild dataset. Figure 5 - 7 shows the distributions of the PSNR simulation results for various smoothed images. A higher PSNR value indicates better smoothing.

From Table 3 it can be seen that the proposed level-set adaptive median filter definitively outperforms the adaptive median filter in all cases. From Figures 5 - 7, it is clear that the level-set adaptive median filter outperforms the adaptive median filter in the case of symmetrical noise (Gaussian noise) and non-symmetrical noise (Gumbel noise) as well as impulsive noise (Salt and Pepper noise). The distributions of the PSNR results of the level-set adaptive median filter (green and red lines) are concentrated higher in general than that of the adaptive median filter (blue and yellow lines).

In Figures 8 - 10 the difference between the PSNR for each image is calculated and plotted. Since the PSNR is the ratio of peak signal-to-noise, a higher value is desired. Therefore, to compare the performance of the algorithms, we can subtract the PSNR results on the same images for the two different algorithms from each other. If we subtract the PSNR of the level-set adaptive median filter from that of the adaptive median filter, the smaller the value would indicate the LS-AM is performing better than the AM. If this difference is negative, it means the PSNR of the LS-AM filter is higher than that of the AM since a higher PSNR value is an indication of better performance. It can be seen that the distribution of the difference of the PSNR for both levels of noise contamination where Gaussian and Gumbel noise was added is in the negative values, which supports the above-mentioned statement that the level-set adaptive median filter does perform better than the adaptive median filter.

Figures 11 - 13 provide a snapshot of an image evaluated in the experiment discussed above. Figures 11 - 13 show some examples of the proposed algorithm applied to images where noise was added. In each figure, the image (a) shows the original image. Images (b) and (e) show the same image with various types of noise added at different intensities (Figure 11 has Gumbel distributed noise added, Figure 12 has Gaussian distributed noise added, and Figure 13 has Salt and Pepper noise added). Images (c) and (f) show the results of level-set adaptive median filter. Images (d) and (g) show the result of the adaptive median filter on the various noises and intensities. For Figures 11 and Figure 12, it can be difficult to see the noise added to the image due to the type of noise added to it. For Figure 13, we can see the noise and the resulting much easier since it is impulsive noise that is added to the images.

Except for Figure 13, the results are difficult to see and compare visually, therefore we make use of metrics to measure and compare the results. One of these metrics, PSNR, is summarised in Table 3 where we can see that although the difference in the results of the adaptive median filter(AM) and the level-sets adaptive median filter(LS-AM) are small the LS-AM does perform better than the standard AM. When looking at Figure 13, we can see the differences visually a bit better, and we can see both the algorithms smooth the noise quite well overall. However, when looking at the edges of the images we can see that the AM struggles to smooth the noise that is observed at the edge of the image whereas the LS-AM is able to smooth this noise as well.

# 5. Conclusion

The proposed level-set adaptive median filter takes advantage of the underlying structure and detail in the image by allowing for variable size as well as variable shape windows which adapt to the image content, namely the level-sets. In doing so it performs well in terms of edge preservation, an important characteristic of a filter, and specifically better than the traditional adaptive median filter. The filter performs well on impulsive, skewed, and symmetrical noise.

The filter examines all level-sets of size $n_{\max}$ and smaller. This, along with the fact that the level-sets are investigated sequentially, can lead to computational intensity, especially in larger images. The same shortcoming does however apply to the traditional adaptive median filter.

An added benefit of the proposed filter is that similarly to the traditional adaptive median filter, the level-sets are investigated to determine whether it is noise-contaminated or not. Therefore, if a set is determined to be noise and subsequently smoothed, the reasoning as to why this happened can be explained.

In this paper, a level-set adaptive median filter to remove noise in images while preserving the edges in the image is proposed. This filter was compared to the adaptive median filter. Three different types of noise were considered namely salt and pepper, Gaussian and Gumbel noise. The performance of these filters were compared using the Pratt's Figure of Merit and Peak Signal-to-Noise Ratio. The proposed filter outperformed the adaptive median filter for all types of noise albeit the difference are small for some of the scenarios. Both the AM and LS-AM were coded in Python and optimised as far as possible, but further optimisation may be possible in a low-level programming language like C++.

Future work could investigate optimisation of $p_{\max}$ choice using total variation convergence criteria or investigating the difference of total variation in the image is less than some predefined value, $\delta$. The filter proposed herein provides a useful addition to the pool of robust median filters, with the added robustness of edge preservation and adaptiveness to the image content.

# References

Al-Amri, S., Kalyankar, N., and Khamitkar, S. (2010). A comparative study of removal noise from remote sensing image. *International Journal of Computer Science Issues*, 7(1).

Anguelov, R. and Fabris-Rotelli, I. (2010). LULU operators and Discrete Pulse Transform for multidimensional arrays. *IEEE Transactions on Image Processing*, 19(11):3012–3023, DOI: 10.1109/tip.2010.2050639.

Arias-Castro, E. and Donoho, D. L. (2009). Does median filtering truly preserve edges better than linear filtering? *The Annals of Statistics*, 37(3), ISSN: 0090-5364, DOI: 10.1214/08-aos604.

Braga-Neto, U. and Goutsias, J. (2004). Grayscale level connectivity: theory and applications. *IEEE Transactions on Image Processing*, 13(12):1567–1580, DOI: 10.1109/TIP.2004.837514.

Cremers, D., Rousson, M., and Deriche, R. (2007). A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, DOI: 10.1007/s11263-006-8711-1.

Davis, L. S. (1975). A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4(3):248–270, DOI: 10.1016/0146-664x(75)90012-x.

Dougherty, E. R. and Astola, J. (1994). *An Introduction to Nonlinear Image Processing*, volume 16. SPIE Press.

Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. (2008). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

Huynh-Thu, Q. and Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, DOI: 10.1049/el:20080522.

Hwang, H. and Haddad, R. A. (1995). Adaptive median filters: new algorithms and results. *IEEE Transactions on Image Processing*, 4(4):499–502, DOI: 10.1109/83.370679.

Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*, volume 5. McGraw-hill New York.

Justusson, B. (1981). Median filtering: Statistical properties. In *Two-Dimensional Digital Signal Prcessing II*, pages 161–196. Springer, DOI: 10.1007/bfb0057597.

Lin, H.-M. and Willson, A. N. (1988). Median filters with adaptive length. *IEEE Transactions on Circuits and Systems*, 35(6):675–690, DOI: 10.1109/31.1805.

Mustafa, Z. A., Abrahim, B. A., and Kadah, Y. M. (2012). K11. modified hybrid median filter for image denoising. In *2012 29th National Radio Science Conference (NRSC)*, pages 705–712. IEEE, DOI: 10.1109/nrsc.2012.6208586.

Pang, K., Shi, Z., Xu, J., and Yao, S. (2018). Adaptive partition-cluster-based median filter for random-valued impulse noise removal. *Journal of Circuits, Systems and Computers*, 27(07):1850110, DOI: 10.1142/s0218126618501104.

Patidar, P., Gupta, M., Srivastava, S., and Nagawat, A. (2010). Image de-noising by various filters for different noise. *International Journal of Computer Applications*, 9(4):45–50, DOI: 10.5120/1370-1846.

Peters, R. (1995). A new algorithm for image noise reduction using mathematical morphology. *IEEE Transactions on Image Processing*, 4(5):554–568, DOI: 10.1109/83.382491.

Serra, J. (2006). A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision*, 24(1):83–130, DOI: 10.1007/s10851-005-3616-0.

Singh, N. and Maheswari, O. U. (2017). A new denoising algorithm for random valued impulse noise in images using measures of dispersion. In *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, pages 1–6. IEEE, DOI: 10.1109/icscn.2017.8085737.

Tukey, J. W. (1977). *Exploratory Data Analysis*, volume 2. Reading, Mass.

Verma, K., Singh, B. K., and Thoke, A. (2015). An enhancement in adaptive median filter for edge preservation. *Procedia Computer Science*, 48:29–36, DOI: 10.1016/j.procs.2015.04.106.

Verma, R. and Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(10).

Wang, C., Chen, T., and Qu, Z. (2010). A novel improved median filter for salt-and-pepper noise from highly corrupted images. In *2010 3rd International Symposium on Systems and Control in Aeronautics and Astronautics*, pages 718–722. IEEE, DOI: 10.1109/isscaa.2010.5633074.

Wang, Z., Ma, B., and Zhu, Y. (2021). Review of level set in image segmentation. *Archives of Computational Methods in Engineering*, 28(4):2429–2446, DOI: 10.1007/s11831-020-09463-9.

Yu, Y. and Acton, S. T. (2002). Speckle reducing anisotropic diffusion. *IEEE Transactions on Image Processing*, 11(11):1260–1270, DOI: 10.1109/tip.2002.804276.

# A. Proposition proof

**Proposition 2.** *Consider and image $f$. Let $L : \mathcal{A}(\mathbb{Z}^2) \to \mathcal{A}(\mathbb{Z}^2)$ be the level-sets adaptive median filter such that*

$$f_{smoothed} = L(f)$$

*The level-set adaptive median filter results in*

$$\delta(f_{smoothed}) \leq \delta(f),$$

$$\gamma(f_{smoothed}) \geq \gamma(f)$$

*and*

$$\beta(f_{smoothed}) \leq \beta(f)$$

*where $f_{smoothed}$ is the resulting image after applying the level-set adaptive median filter to image $f$.*

*Proof.* Consider and image $f$ and the set $V$ and $f(B_V^{(n)})$ being the set $V$ with its $n$-order neighbours. Let $L : \mathcal{A}(\mathbb{Z}^2) \to \mathcal{A}(\mathbb{Z}^2)$ be the level-sets adaptive median filter such that

$$f_{smoothed} = L(f).$$

From Algorithm 1 it is clear that there are four scenarios that can occur when the level-set adaptive median filter is applied to image $f$ with two possible outcomes, namely $f_{smoothed}(V) = \text{med}(f(B_V^{(n)}))$ or $f_{smoothed}(V) = f(V)$.

Now consider the four scenarios:

Scenario 1
$\text{med}(f(B_V^{(n)})) = \min(f(B_V^{(n)}))$ and $f(V) \neq \text{med}(f(B_V^{(n)}))$.

This will occur if at least half of the values in $f(B_V^{(n)})$ are equal to the $\min(f(B_V^{(n)}))$, resulting in, $f_{smoothed}(V) = \text{med}(f(B_V^{(n)}))$.

This will result in more values in set $V$ being equal to one another, so that $\delta(f_{smoothed}) < \delta(f)$, $\gamma(f_{smoothed} > \gamma(f)$ and $\beta(f_{smoothed} < \beta(f)$.

Scenario 2
$\text{med}(f(B_V^{(n)})) = \max(f(B_V^{(n)}))$ and $f(V) \neq \text{med}(f(B_V^{(n)}))$.

This will occur if at least half of the values in $f(B_V^{(n)})$ are equal to the $\max(f(B_V^{(n)}))$, resulting in $f_{smoothed}(V) = \text{med}(f(B_V^{(n)}))$. This will result in more values in set $V$ being equal to one another, so that $\delta(f_{smoothed}) < \delta(f)$, $\gamma(f_{smoothed}) > \gamma(f)$ and $\beta(f_{smoothed}) < \beta(f)$.

Scenario 3
$\min(f(B_V^{(n)})) < \text{med}(f(B_V^{(n)})) < \max(f(B_V^{(n)}))$ and either $f(V) = \min(f(B_V^{(n)}))$ or $f(V) = \max(f(B_V^{(n)}))$.

Then we have $f_{smoothed}(V) = \text{med}(f(B_V^{(n)}))$; in other words, one of the extremes was replaced with the median, so that $\delta(f_{smoothed}) < \delta(f)$, $\gamma(f_{smoothed}) > \gamma(f)$ and $\beta(f_{smoothed}) < \beta(f)$.

Scenario 4
$\min(f(B_V^{(n)})) < \text{med}(f(B_V^{(n)})) < \max(f(B_V^{(n)}))$ and $\min(f(B_V^{(n)})) < f(V) < \max(f(B_V^{(n)}))$.

In this scenario, the value of the set remains unchanged, so that $\delta(f_{smoothed}) = \delta(f)$, $\gamma(f_{smoothed}) = \gamma(f)$ and $\beta(f_{smoothed}) = \beta(f)$.

Therefore, we can conclude that when the level-set adaptive median filter is applied to an image, $\delta(f_{smoothed}) \leq \delta(f)$, $\gamma(f_{smoothed}) \geq \gamma(f)$ and $\beta(f_{smoothed}) \leq \beta(f)$.

Thus this filter will at worst result in the same strength edge or result in a stronger edge.

□

# B. Peak Signal to Noise Comparison

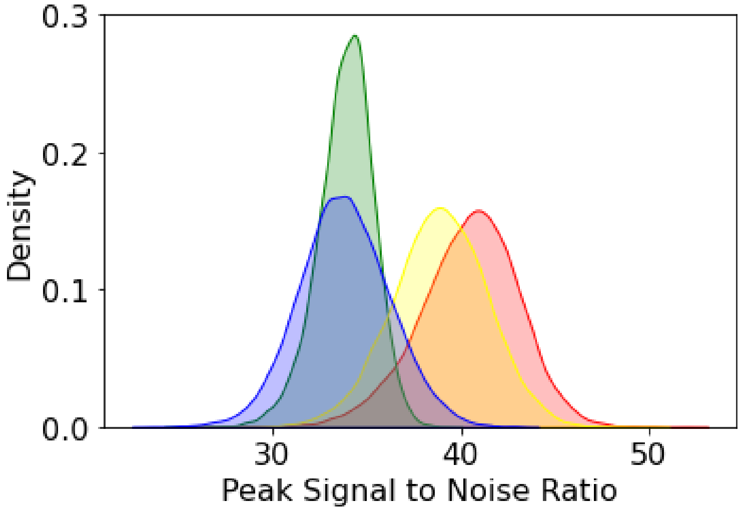


Figure 5: Distribution of simulation results for images with salt-and-pepper noise (red: sevel-set AM, contamination = 10%; yellow:AM, contamination = 10%; green: level-set AM, contamination = 20%; blue:AM, contamination = 20%).

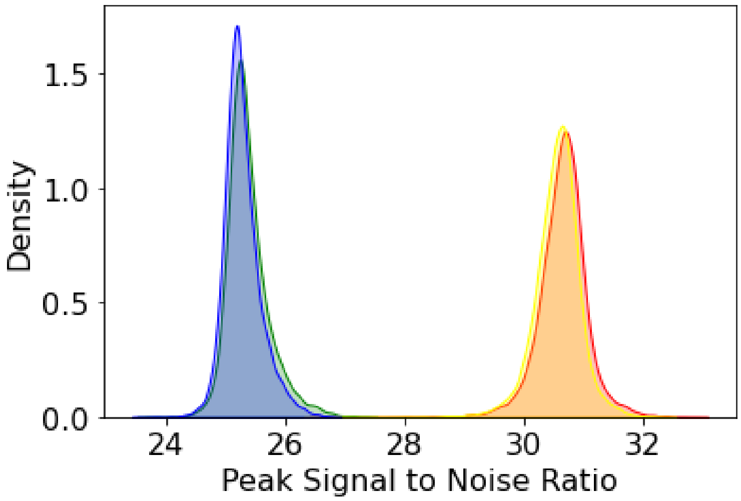


Figure 6: Distribution of simulation results for images with noise from the Gaussian distribution (red: level-set AM, std. dev=10; yellow:AM, std. dev=10; green: level-set AM, std. dev=20; blue:AM, std. dev=20).

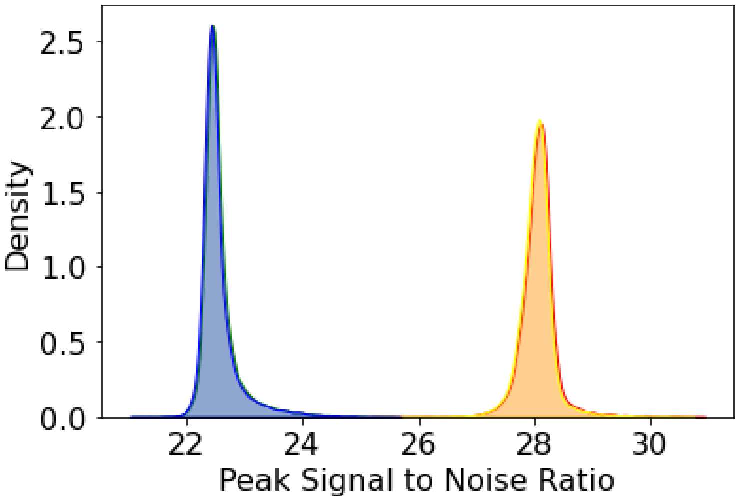


Figure 7: Distribution of simulation results for images with noise from the Gumbel distribution (red: level-set AM, std. dev=10; yellow:AM, std. dev=10; green: level-set AM, std. dev=20; blue:AM, std. dev=20).

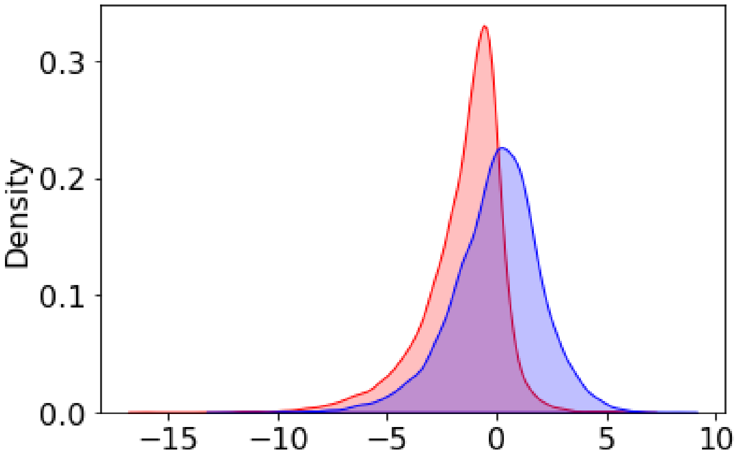


Figure 8: Distribution of image-specific difference between PSNR after smoothing images with salt-and-pepper noise (red: 10% contamination; blue: 20% contamination).

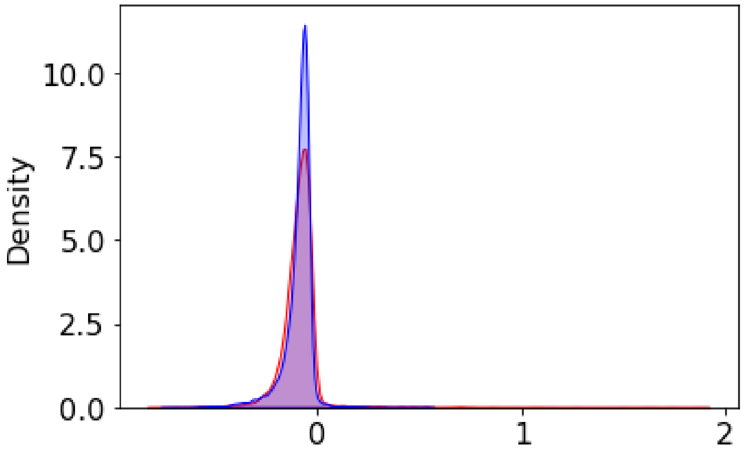


Figure 9: Distribution of image-specific difference between PSNR after smoothing images with noise from the Gaussian distribution (red: std. dev=10; blue: std. dev=20).

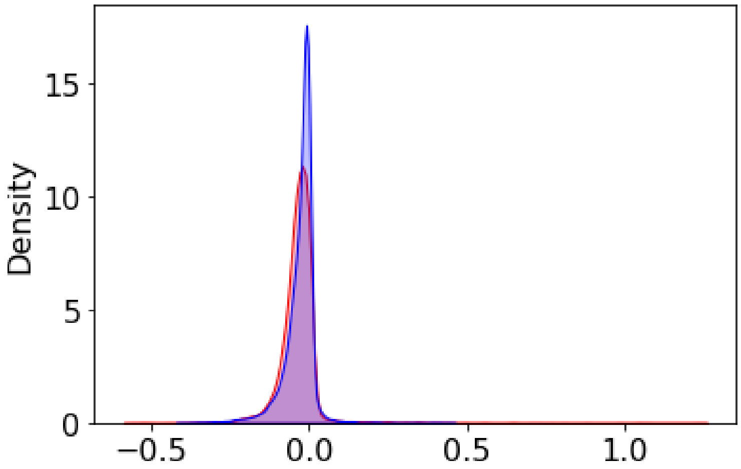
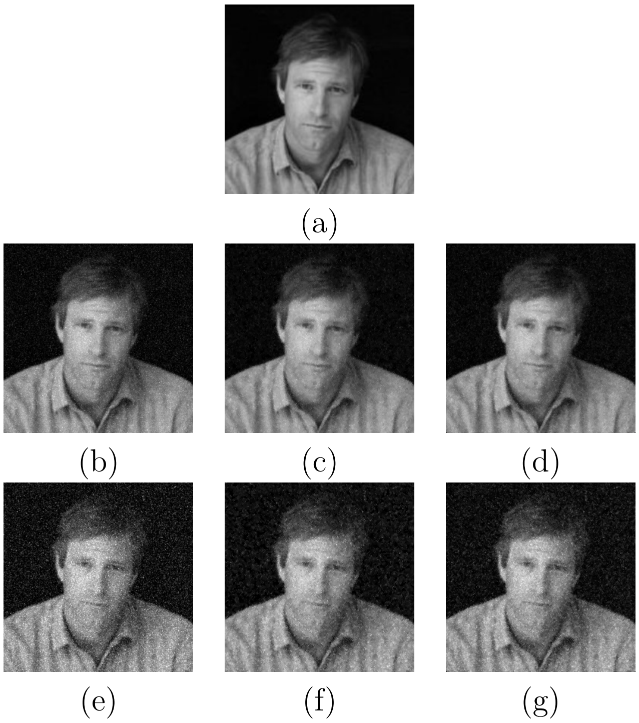


Figure 10: Distribution of image-specific difference between PSNR after smoothing images with noise from the Gumbel distribution (red: std. dev=10; blue: std. dev=20).

# C. Results



(a)

(b) (c) (d)

(e) (f) (g)

Figure 11: (a) The original image. (b) The images with Gumbel(0,10) noise. (c) Results of level-set filter on images with Gumbel(0,10) noise. (d) Results of adaptive median filter on images with Gumbel(0,10) noise. (e) The images with Gumbel(0,20) noise added. (f) Results of level-set filter on images with Gumbel(0,20) noise. (g) Results of adaptive median filter on images with Gumbel(0,20) noise.

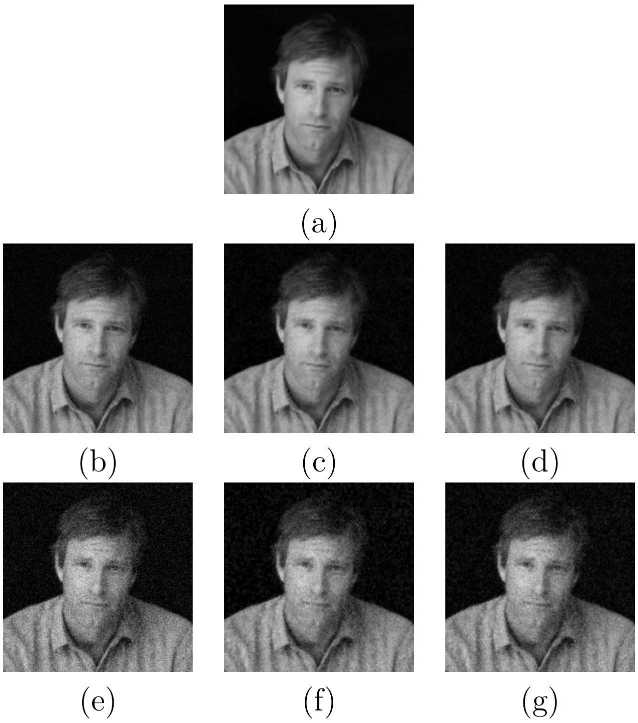


Figure 12: (a) The original image. (b) The images with Gaussian (0,10) noise. (c) Results of level-set filter on images with Gaussian (0,10) noise. (d) Results of adaptive median filter on images with Gaussian(0,10) noise. (e) The images with Gaussian(0,20) noise added. (f) Results of level-set filter on images with Gaussian(0,20) noise. (g) Results of adaptive median filter on images with Gaussian(0,20) noise.

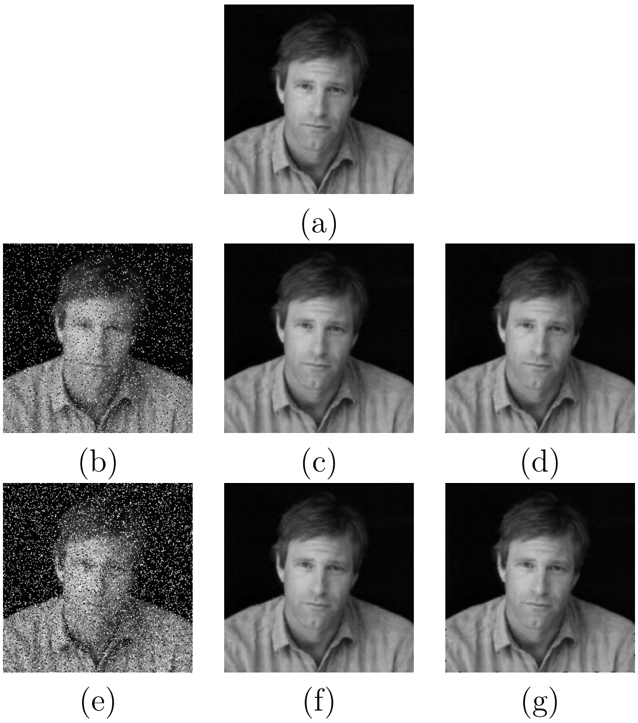


Figure 13: (a) The original image. (b) The images with 10% salt and pepper noise added. (c) Results of level-set filter on images with 10% salt and pepper noise. (d) Results of adaptive median filter on images with 10% salt and pepper noise. (e) The images with 20% salt and pepper noise added. (f) Results of level-set filter on images with 20% salt and pepper noise. (g) Results of adaptive median filter on images with 20% salt and pepper noise.

## Affiliation:

Jean-Pierre Stander
Department of Statistics
University of Pretoria
Lynnwood Road, Hatfield, Pretoria, South-Africa
E-mail: u15002536@tuks.co.za

Inger Fabris-Rotelli
Department of Statistics
University of Pretoria
Lynnwood Road, Hatfield, Pretoria, South-Africa
E-mail: inger.fabris-rotelli@up.ac.za

Theodor Loots
Department of Statistics
University of Pretoria
Lynnwood Road, Hatfield, Pretoria, South-Africa
E-mail: theodor.loots@up.ac.za

Johan van Niekerk
University of Twente
Drienerlolaan 5, 7522 NB Enschede, Netherlands
E-mail: xionvann@gmail.com

Alfred Stein
University of Twente
Drienerlolaan 5, 7522 NB Enschede, Netherlands
E-mail: astein@utwente.nl

*Journal of Data Science, Statistics, and Visualisation* https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/
April 2024, Volume IV, Issue III *Submitted:* 2022-09-04
doi:10.52933/jdssv.v4i3.74 *Accepted:* 2023-11-21