Seddik *EURASIP Journal on Advances in Signal Processing* 2014, **2014**:25
http://asp.eurasipjournals.com/content/2014/1/25

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH **Open Access**

A new family of Gaussian filters with adaptive lobe location and smoothing strength for efficient image restoration

Hassene Seddik

Abstract

Noise can occur during image capture, transmission, or processing phases. Image de-noising is a very important step in image processing, and many approaches are developed in order to achieve this goal such as the Gaussian filter which is efficient in noise removal. Its smoothing efficiency depends on the value of its standard deviation. The mask representing the filter presents generally static weights with invariant lobe. In this paper, an adaptive de-noising approach is proposed. The proposed approach uses a Gaussian kernel with variable width and direction called adaptive Gaussian kernel (AGK). In each processed window of the image, the smoothing strength changes according to the image content, noise kind, and intensity. In addition, the location of its lobe changes in eight different directions over the processed window. This directional variability avoids averaging details by the highest mask weights in order to preserve the edges and the borders. The recovered data is de-noised efficiently without introducing blur or losing details. A comparative study with the static Gaussian filter and other recent techniques is presented to prove the efficiency of the proposed approach.

Keywords: Efficient image de-noising; Variable Gaussian core; Neural network; Directional core; Edge preserving

1. Introduction

The image de-noising remains an important goal in image or video pre-processing as a preliminary task for data transmitting, pattern recognition, etc. In the case of high distortions, efficient noise-removing techniques may introduce artifacts or blur the image. Image de-noising techniques using the Gaussian filter has been widely used in many fields for its ability to efficiently restore degraded data. In [1], the authors combined the following three techniques: wavelet transform, curvelet transform, and the Gaussian filter to recover the distorted image. The authors in [2] exploited the relationship between linear diffusion and Gaussian scale space to estimate optimal variances and window size of the Gaussian. An efficient technique based on the Gaussian filter with dynamic structure that targets noise is introduced in [3,4]. Selecting the optimal value of the standard deviation in a Gaussian filter depending on few properties of the signal knowledge is proposed in [5]. In [6], an adaptive Gaussian filtering algorithm, in which the filter variance is adapted to both noise characteristics and the local variance of the signal, is studied.

The basic premise of the Gaussian technique is that different parts of an image have varying degrees of noisiness and types of edges. Therefore, each part of the image needs to be smoothed differently. For this reason, we propose to create an adaptive filter having a Gaussian core with a variable structure for each processed window. The location of the Gaussian lobe and its smoothing strength are optimized iteratively according to the noise intensity and image characteristics. These features are optimized to efficiently clean noise and preserve the image content. The paper is arranged as follows: a brief description of the Gaussian kernel in Section 2, the conception of the proposed filter in The proposed adaptive Gaussian filter and Experimental results, comparative study in Comparative study, and a summary and conclusion in Conclusions.

Correspondence: hassene.seddik@esstt.rnu.tn
ENSIT, University of Tunis, 05 Avenue Taha Hussein, Montfleury, Tunis 1005, Tunisia

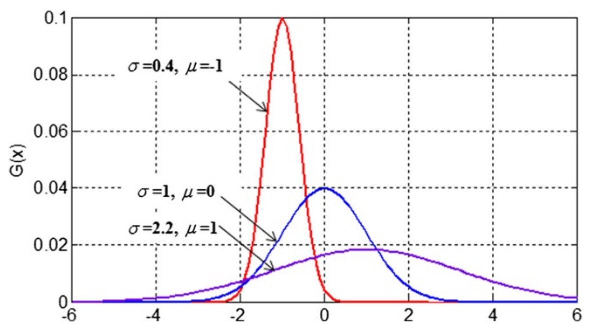


Figure 1 Variation of the Gaussian function.

2. Recall of the conventional Gaussian filter properties

The Gaussian filter is widely used in the literature as a low-pass filter for image de-noising. It can provide image smoothing and noise reduction, but it blurs edges and details. The Gaussian smoothing filter is efficient for reducing noise drawn from a normal distribution presented as follows:

$$G(x,y,\mu,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left[\frac{-(x-\mu_1)^2-(y-\mu_2)^2}{2\sigma^2}\right] \quad (1)$$

where σ is the spread parameter (the width distribution) and the couple (μ_1, μ_2) are the means (location of the peak) (Figure 1). In this paper, we propose a smart Gaussian filter with dynamic structure. In this new filter, the variation of the standard deviation is related to the nature and the characteristic of the image areas and zones. This variation is supervised by the neural network. However, changing the couple of means (μ_1 and μ_2) will vary the lobe position and the gradient magnitude in order to preserve the edge and borders.

The main properties of the Gaussian filter are described as follows:

- Gaussian smoothing is very effective for removing Gaussian noise.
- The weights give higher signification to pixels near the edge (reduce edge blurring).
- It is a static and linear low-pass filter.
- Separability into two one-dimensional (1D) filters.
- Rotationally symmetric (performs the same in all directions).
- The degree of smoothing is controlled by the standard deviation σ (larger σ for more intensive smoothing).

Figure 2 shows the result of applying Gaussian filters with different values of σ on the Lena image. It is clear that when we increase the width parameter (σ), the borders and the details are removed.

In order to overcome this problem, we study a smart Gaussian filter with dynamic structure. In this new filter,

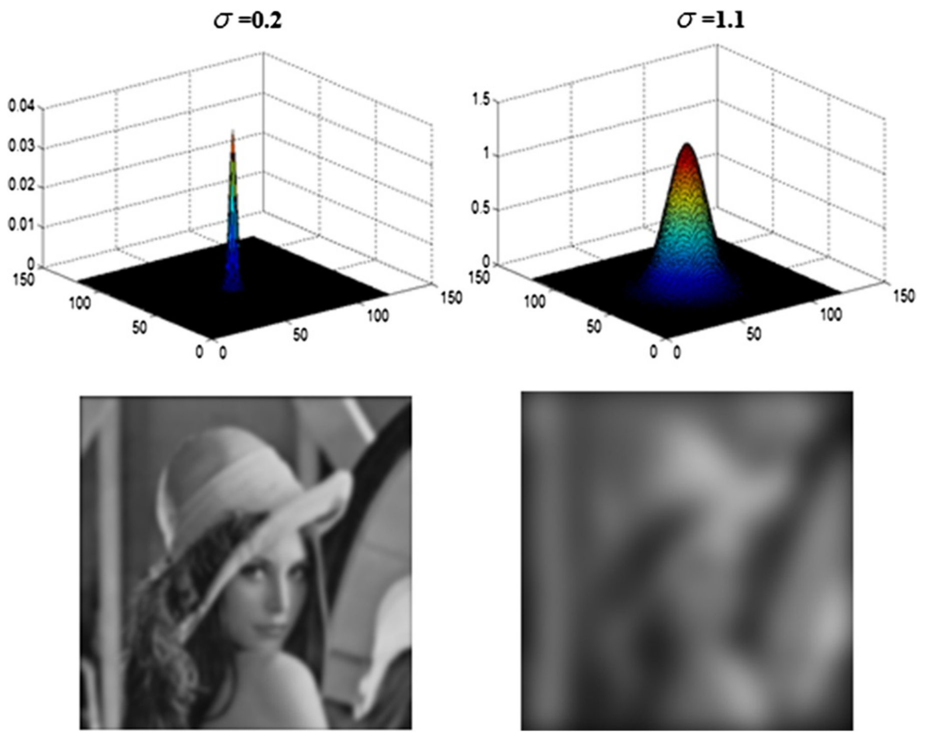


Figure 2 An image filtered with different variances.

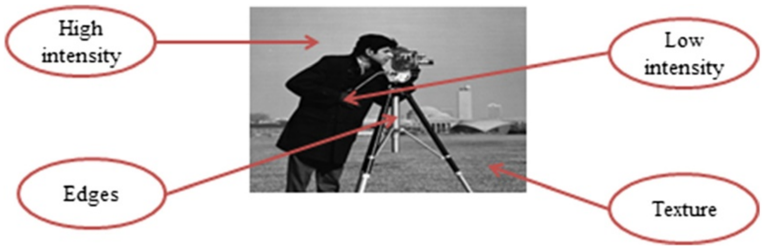


Figure 3 Pattern selection.

the variation of the standard deviation is done according to the nature and characteristic of the image areas and zones. This variation is supervised by a neural network, whereas changing the couple of means (μ_1 and μ_2) will vary the position of the filter lobe in order to preserve edges and borders.

3. The proposed adaptive Gaussian filter

3.1. Estimation of the adaptive smoothing strength

As we mentioned, the smoothing efficiency of the Gaussian filter depends on the value of its standard deviation. The more this parameter is increased, the more the image is blurred and the details and borders are removed. That is why every part of the image must be filtered with an appropriate kernel. So, we propose to vary the standard deviation using the neural network which responds to the non-linear variation of the smoothing strength. The procedure is described as follows.

From the processed image, we select patterns from different zones such as the edges, the dark zone, the high intensities, and the textures (Figure 3).

In order to estimate the optimal standard deviation which is the desired output of the neural network, the following stages will be undertaken. Each selected window from the image is distorted by different kinds of noise and filtered iteratively with a Gaussian filter increasing in each step the value of its standard deviation by a step $\Delta\sigma = 0.01$, and we compute the corresponding peak signal-to-noise ratio (PSNR). For each pattern of the selected windows, a curve illustrating the PSNR variation with regard to the standard deviation increase is plotted (see Figure 4). Once the allure of the function $\text{PSNR}_\sigma = p(\sigma)$ is determined, we compute its derivative to find the optimal value of the standard deviation called σ_{opt}. The segment *D* illustrated in red is the tangent to the curve at the index σ_{opt} where the derivative is null $P'(\sigma_{\text{opt}}) = 0$ and *D* is defined by the following equation:

$$D = P'(\sigma_{\text{opt}})(d\sigma - \sigma_{\text{opt}}) + P(\sigma_{\text{opt}}) \tag{2}$$

The optimal value of the standard deviation is manually adjusted around the computed value in the range of $[-2\Delta\sigma, 2\Delta\sigma]$. In fact, σ_{opt} is located between the transitory and stable zone of the PSNR_σ function that are separated by the computed tangent. To validate our selection, we compute the normalized cross-correlation between the filtered and original image called *C*. A segment noted *D*1 is drawn as a tangent on this curve with parallel direction to *D*. The index representing the value of the standard deviation found confirms the computed σ_{opt}. The range of σ_{opt} is constrained by a minimum threshold imposed to the PSNR called PSNR_{min} that must be maintained over 32 dB.

The selected distorted patterns of the image are introduced to a multilayer perceptron (MLP) neural network which is composed of three layers ‘input, hidden layer,

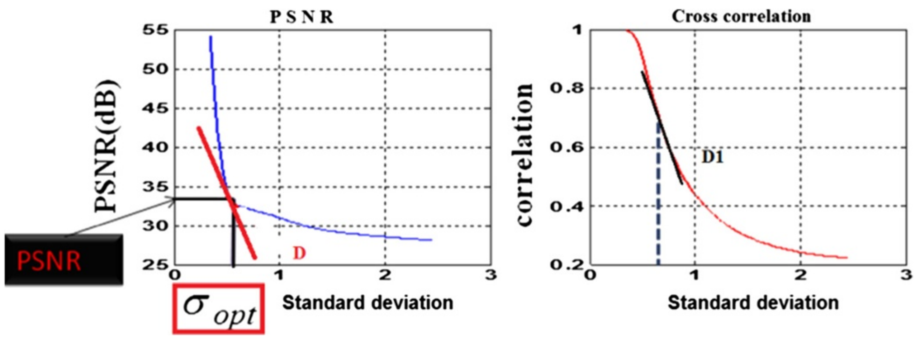


Figure 4 PSNR results and normalized cross-correlation used to determine the optimal standard deviation.

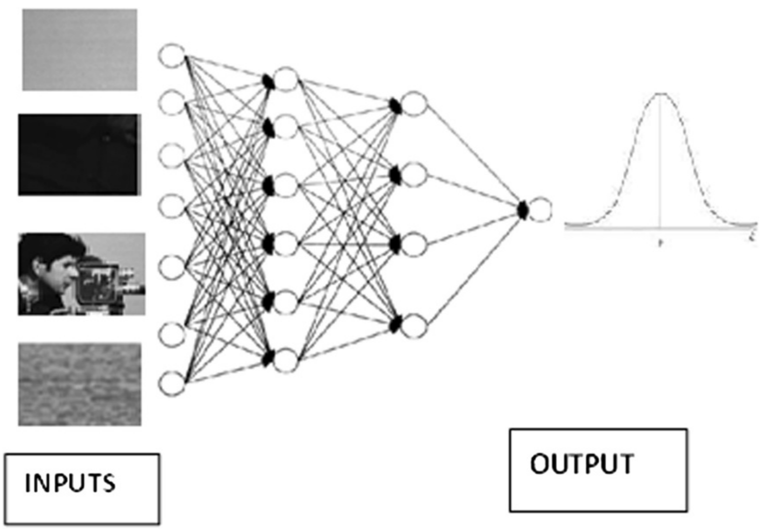


Figure 5 Neural network training.

and output layer'. In the test phase, the neural network generates different values of standard deviations for all the introduced distorted windows. The network generates the appropriate outputs according to the noise density and kind (Figure 5).

The neural weights are adjusted to adapt the neural output to the desired ones corresponding to the distorted inputs. When filtering the image, the neuronal outputs supervise real time the smoothing strength of the proposed adaptive kernel that changes iteratively for each processed window.

$$E_\sigma = \frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{K}\left[\sigma_{(\mathrm{opt})p,k} - \sigma_{\mathrm{RN}(p,k)}\right]^2 \tag{3}$$

If we consider the window size is $(m \times n)$ and the image size is $(M \times N)$, then the set of estimated standard deviations is as follows: $[\sigma_{\mathrm{RN}}] = \sigma_{\mathrm{RN}}(i)$ such as $i \in \left[1, \frac{M\times N}{m\times n}\right]$, where σ_{RN} represents the standard deviation output by the trained network and i is the iterative number of the processed windows. The filter size is fixed equal to $(6 \times \sigma + 1)$ in order to cover more than 99.99% of the processed data. The new filter structure is illustrated by the following equation:

$$G_i(x, y, \sigma_{\mathrm{RN}}) = \sum_{i=1}^{\frac{M\times N}{m\times n}} \frac{1}{2\pi\cdot\sigma_{\mathrm{RN}}^2(i)} \times \exp\frac{-\left(X^2_{[(6\sigma_{\mathrm{RN}}(i)+1)\times(6\sigma_{\mathrm{RN}}(i)+1)]} + Y^2_{[(6\sigma_{\mathrm{RN}}(i)+1)\times(6\sigma_{\mathrm{RN}}(i)+1)]}\right)}{2\sigma_{\mathrm{RN}}^2(i)} \tag{4}$$

3.2. The adaptive kernel location

The (μ_1, μ_2) represent the positions of the Gaussian core (location of the peak). In this work, we apply the Gaussian filter only on noise to avoid blurring details and borders. The steps of the kernel location variability are presented as follows:

First step: Edge detection using the canny high-boost filter operator presented by the following equation:

$$f(x) = a\exp(-\alpha x)\cdot\sin(wx) \tag{5}$$

Second step: Filter the noisy image based on a decision computed from 8 to 25 neighborhoods' comparison:

- If $I(x, y)$ 'the processed pixel' belongs to an edge, compute the difference between this pixel and its

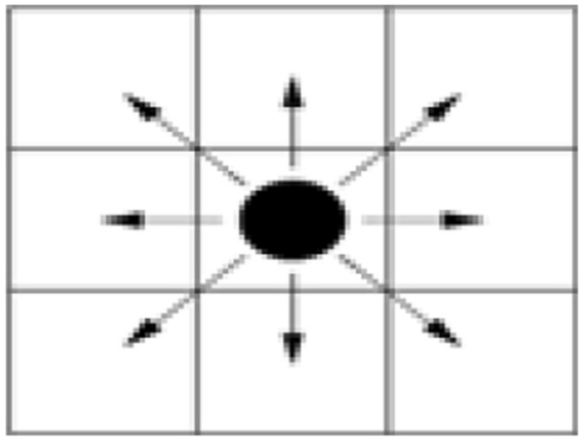


Figure 6 Pixel neighboring comparison.

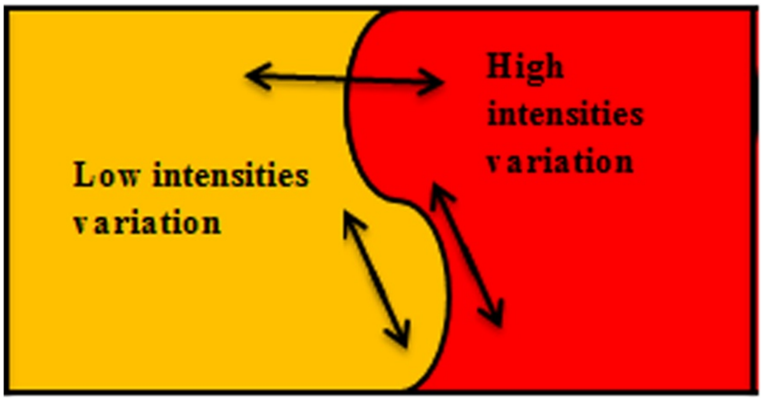


Figure 7 Area of significant variation in the image intensity.

eight neighbors called $P(x, y)$ (Figure 6 and Equation 6); in this case, the values of the mean (μ_1 and μ_2) are determined according to the maximum variation (gradient).

- Elsewhere, we process the selected window using a filter with support size equal to $(6 \times \sigma_{opt} + 1) \times (6 \times \sigma_{opt} + 1)$ and the appropriate smoothing strength.

$$P(x,y) = \begin{cases} I(x,y)-I(x-1,y-1) \\ I(x,y)-I(x-1,y) \\ I(x,y)-I(x-1,y+1) \\ I(x,y)-I(x,y-1) \\ I(x,y)-I(x,y+1) \\ I(x,y)-I(x+1,y-1) \\ I(x,y)-I(x+1,y) \\ I(x,y)-I(x+1,y+1) \end{cases} \quad (6)$$

For 1 to 8 (number of neighbors for each processed pixel in a window of size '3 × 3', the maximum variation max[$P(x,y)$] is computed to determine automatically the values of the means (μ_1 and μ_2) and define the location of the Gaussian lobe.

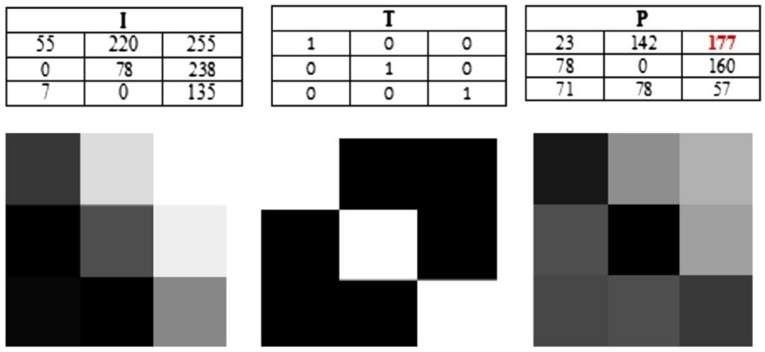
Sharp changes in an image can be associated to edges, or noise and such changes correspond to higher gradients. To consider that a pixel $I(x, y)$ belongs to an edge and not as noise, we must satisfy two conditions:

- High gradient variation between this pixel and its 8 or 25 neighbors for (3 × 3) or (5 × 5) window size
- Connection continuity between different pixels considered as edges as presented in Figure 7

The gradient has the advantage of detecting the orientation of an edge which can be positive or negative. The gradient of an image measures how it is changing. It provides two essential information. The magnitude of the gradient tells us how quickly the image content is changing, while the direction of the gradient tells us the orientation of these rapid changes. The gradient magnitude of an image and its approximation is given by the following equations:

$$\vec{\nabla} G[I(x,y)] = \begin{bmatrix} \mathrm{Gr}\,\overrightarrow{x} \\ \mathrm{Gr}\,\overrightarrow{y} \end{bmatrix} = \begin{bmatrix} {}^{\partial I}/_{\partial x} \\ {}^{\partial I}/_{\partial y} \end{bmatrix} \quad (7)$$

$$\mathrm{Mag}\ (\mathrm{Gr}) = \sqrt{\mathrm{Gr}_x^2 + \mathrm{Gr}_y^2} \approx |\mathrm{Gr}_x| + |\mathrm{Gr}_y| \quad (8)$$



I		
55	220	255
0	78	238
7	0	135

T		
1	0	0
0	1	0
0	0	1

P		
23	142	177
78	0	160
71	78	57

Figure 8 Local edge detection and maximum gradient variation.

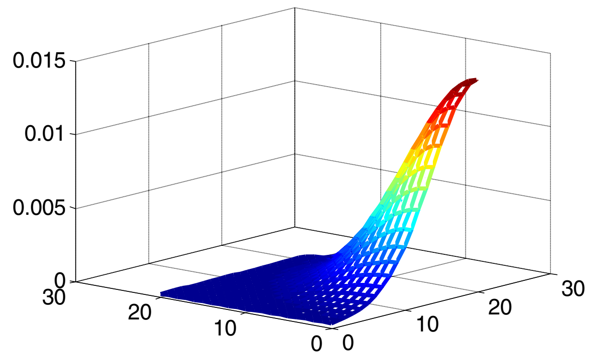


Figure 9 Filter lobe location displacement to the upper right direction of the window (Figure 8).

where the partial derivatives of the image representing its variation for both x and y directions are given as follows:

$$\vec{\tau}_x = \frac{\partial I(x,y)}{\partial x} = \lim_{\Delta x \to 0} \frac{I(x+\Delta x, y) - I(x,y)}{\Delta x} = \lim_{\Delta x \to 0} \frac{I(x,y) - I(x-\Delta x, y)}{\Delta x} \tag{9}$$

$$\vec{\tau}_y = \frac{\partial I(x,y)}{\partial y} = \lim_{\Delta y \to 0} \frac{I(x, y+\Delta y) - I(x,y)}{\Delta y} = \lim_{\Delta y \to 0} \frac{I(x,y) - I(x, y-\Delta y)}{\Delta y} \tag{10}$$

The gradient orientation is described by the following equation:

$$\theta = \arg\left(\vec{\nabla} I\right) = \tan^{-1}\left[\frac{\mathrm{Gr}_y}{\mathrm{Gr}_x}\right] \tag{11}$$

The edge orientation is contained in a window of size (3×3) or (5×5). This direction of the edge is used to determine the Gaussian lobe displacement. This direction is constrained by two factors as follows:

1. $\vec{\tau}_{x,y}$ is always orthogonal to the tangent of the image edge.
2. $\vec{\tau}_{x,y} = \max\left\|\vec{P}(x,y)\right\|$: The lobe displacement follows the maximum gradient of the image windows where the central pixel of this window belongs to the detected edge.

If the continuity condition between pixel neighbors is satisfied, we have the confirmation that the processed pixels belong to an edge. The Gaussian lobe moves in the chosen direction $\vec{\tau}_{x,y}$ with a step of magnitude defined by the following equations respectively to each direction:

$$\left\|\vec{d\tau}_\theta(dx,dy)\right\| = \begin{cases} \left\|\vec{d\tau}_\theta(dx,dy)_1\right\| = \dfrac{|I(x-\Delta x, y-\Delta y) - I(x,y)|}{\sqrt{\Delta x^2 + \Delta y^2}} \\ \left\|\vec{d\tau}_\theta(dx,dy)_2\right\| = \dfrac{|I(x-\Delta x, y) - I(x,y)|}{\Delta x} \\ \left\|\vec{d\tau}_\theta(dx,dy)_3\right\| = \dfrac{|I(x-\Delta x, y+\Delta y) - I(x,y)|}{\sqrt{\Delta x^2 + \Delta y^2}} \\ \left\|\vec{d\tau}_\theta(dx,dy)_4\right\| = \dfrac{|I(x, y-\Delta y) - I(x,y)|}{\Delta y} \\ \left\|\vec{d\tau}_\theta(dx,dy)_5\right\| = 0 \\ \left\|\vec{d\tau}_\theta(dx,dy)_6\right\| = \dfrac{|I(x, y+\Delta y) - I(x,y)|}{\Delta y} \\ \left\|\vec{d\tau}_\theta(dx,dy)_7\right\| = \dfrac{|I(x+\Delta x, y-\Delta y) - I(x,y)|}{\sqrt{\Delta x^2 + \Delta y^2}} \\ \left\|\vec{d\tau}_\theta(dx,dy)_8\right\| = \dfrac{|I(x+\Delta x, y) - I(x,y)|}{\Delta x} \\ \left\|\vec{d\tau}_\theta(dx,dy)_9\right\| = \dfrac{|I(x+\Delta x, y+\Delta y) - I(x,y)|}{\sqrt{\Delta x^2 + \Delta y^2}} \end{cases} \tag{12}$$

Figure 8 illustrates the highest gradient variation toward which the Gaussian kernel will move. I is the selected

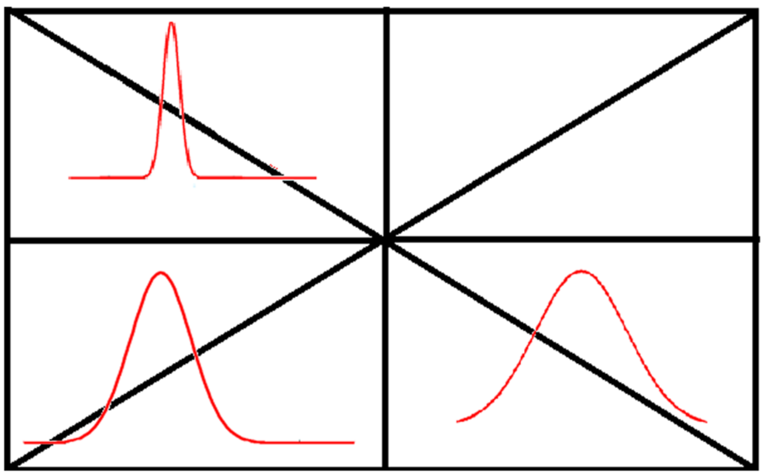


Figure 10 Displacement of the Gaussian core.

Table 1 Comparison between the AGK and the static Gaussian filter

Noise densities	AGK PSNR (dB)	Static PSNR (dB)	ΔPSNR (dB)
0.01	36.84	34.48	2.35
0.02	37.13	33.56	3.54
0.03	37.9	33.2	4.88
0.04	37.8	32.6	5.2
0.05	38.24	32.13	6.11
0.06	38.84	31.78	7.05
0.07	38.97	31.46	7.51
0.08	39.49	31.23	8.26
0.09	39.83	31.06	8.77

noisy window, T is the same window illustrating the detected edge, and P is the computed gradient magnitude.

The Gaussian core is orientated as follows: $\{\mu_1, \mu_2\} = \{-1, 1\}$. Figure 9 shows the highest weight of the Gaussian kernel moving to the upper right direction of the window following the highest magnitude. Doing this, the edge will be preserved and averaged only by the lower filter coefficient. The lobe displacement is always orthogonal to the edge tangent which means that scalar between them is null. The first constraint cited above can express mathematically the Gaussian movements by the following equations:

$$\left\langle \vec{\tau}_{x,y}, \vec{f'}(x,y) \right\rangle = 0 \tag{13}$$

$$\text{where } \vec{\tau}_{x,y} = \begin{bmatrix} \vec{\tau}_x \\ \vec{\tau}_y \end{bmatrix} \text{ and } f'(x,y) = \begin{bmatrix} \dfrac{\partial I(x,y)}{\partial x} \\ \dfrac{\partial I(x,y)}{\partial y} \end{bmatrix}$$

$$\vec{\tau}_x \cdot \frac{\partial I(x,y)}{\partial x} = 0 \text{ and } \vec{\tau}_y \cdot \frac{\partial I(x,y)}{\partial y} = 0 \tag{14}$$

Since the direction of the Gaussian is defined by $\vec{\tau}_{x,y}$, the center of the lobe follows the highest gradient in a direction orthogonal to the edge tangent. Finally, Equation 1 becomes:

$$\vec{G}\left(x, y, \sigma, \left(\mu_1 \cdot \vec{\tau}_x\right), \left(\mu_2 \cdot \vec{\tau}_y\right)\right) = \frac{1}{2\pi\sigma_{\text{opt}}{}^2} \exp\left[\frac{-\left(x-\mu_1 \cdot \vec{\tau}_x\right)^2 - \left(y-\mu_2 \cdot \vec{\tau}_y\right)^2}{2\sigma_{\text{opt}}{}^2}\right] \tag{15}$$

Mathematically, this can be written as the scalar between the Gaussian and the image edge is always null as presented by these equations:

$$\left\langle G(x,y), f'(x,y) \right\rangle = \int_{-\infty}^{+\infty} G(x,y).f'^{*}(x,y)dxdy \tag{16}$$

where * is the conjugate. Since the two functions are discrete and real, it follows:

$$\left\langle G(x,y), f'(x,y) \right\rangle = \sum_{0}^{N}\sum_{0}^{M} G(x,y).f'(x,y) = 0 \tag{17}$$

where the couple (N, M) represents the image size.

On the other hand, the second constraint of $\vec{\tau}_{x,y}$ means that the Gaussian lobe displacement can be in the left or right side of the tangent following the maximum gradient variation (Figure 10). As a result, we can conclude that the proposed adaptive kernel with variable Gaussian core targets noise in each window over the whole image with a variable smoothing strength and kernel location. If it detects edges or borders, the lobe change its location following the highest gradient in the processed

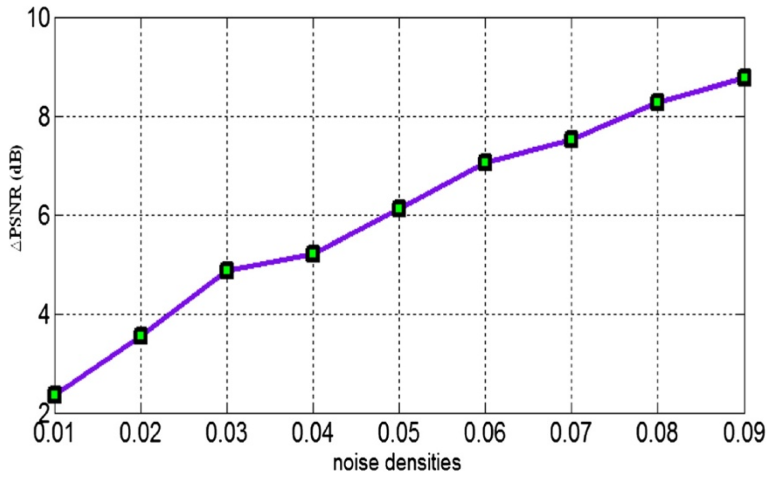


Figure 11 **PSNR difference 'ΔPSNR' between the AGK and the static Gaussian filter for different noise densities.**

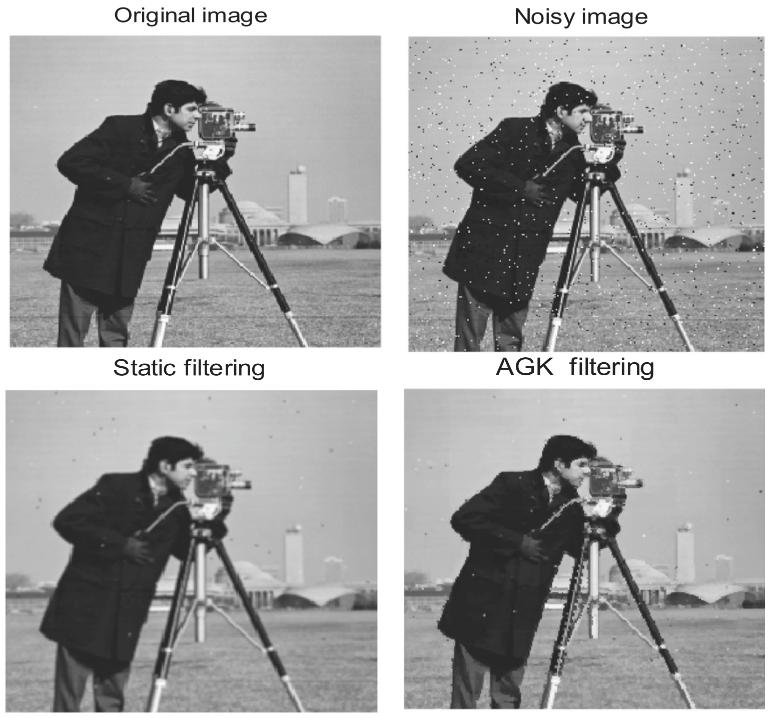


Figure 12 Filtering results.

window in a direction orthogonal to the tangent of this edge. Doing this, we avoid to weight the details and borders by the highest filter weight and introduce blur to the image. The image edges are even preserved and the noise cleaned.

4. Experimental results

We conducted a number of experiments to verify the efficiency of the proposed approach. This section discusses the de-noising results obtained by the proposed adaptive filter.

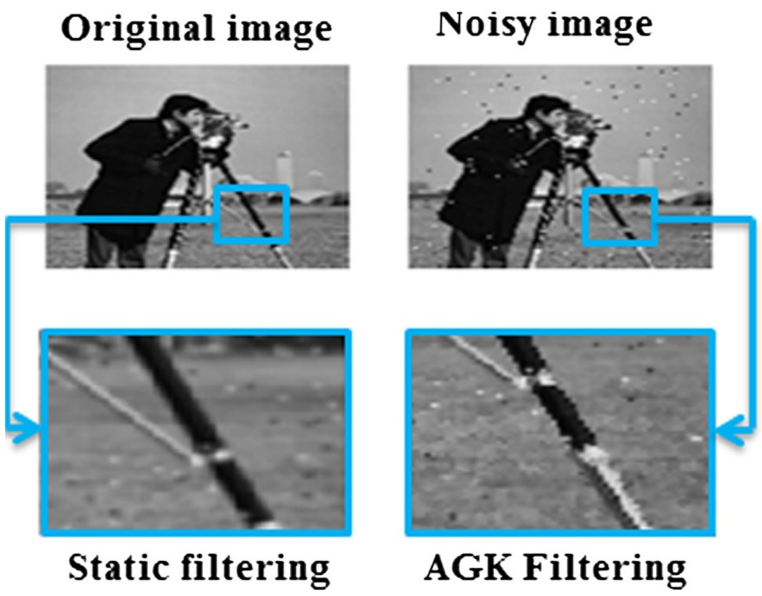


Figure 13 Zoomed views.

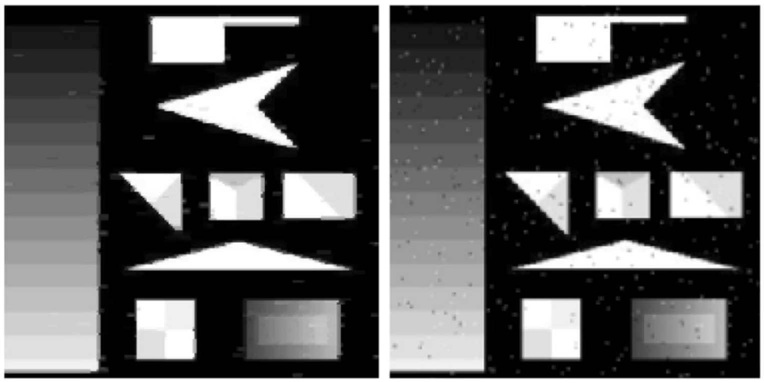


Figure 14 Example of a gray-level image filtered by the AGK and the conventional Gaussian filter.

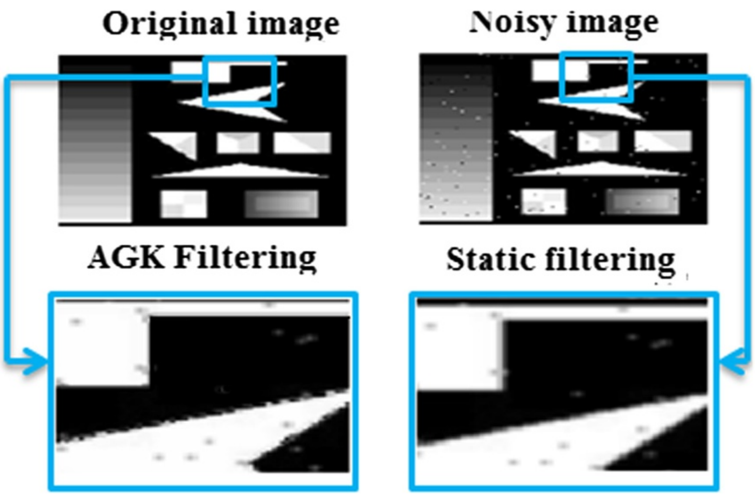


Figure 15 Comparison between the conventional and the AGK filter.

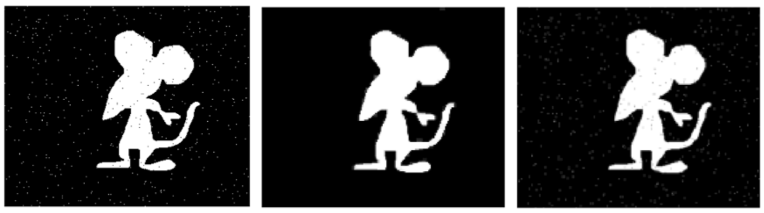


Figure 16 Example of binary image filtered respectively by the AGK and the static filter.

Table 2 PSNR variation

S&P densities	AGK PSNR	Static PSNR	ΔPSNR
0.05	38.06	34.77	3.29
0.06	38.19	34.2	3.99
0.07	38.20	33.87	4.33
0.08	38.28	33.32	4.96
0.09	38.12	33.18	4.94
0.1	38.05	33.03	5.02
0.2	37.397	31.35	6.04
0.3	35.6	3.58	5.01
0.4	33.94	30.24	3.7
0.5	32.8	30	2.79

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{d^2}{\mathrm{MSE}}\right) \tag{18}$$

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(r_{i,j}-f_{i,j}\right)^2}{NM} \tag{19}$$

MSE is the mean square error, d is the maximal coded image intensity, n and m are the image sizes, and f and r are the original and the filtered image.

4.1. Filtering salt and pepper noise

In the salt and pepper noise model, only two possible values are present. For an 8-bit coded image, the typical intensity for pepper noise is close to 0 and for salt noise is close to 255. Our goal is to efficiently remove noise while preserving edges and borders. The proposed method is compared to the static Gaussian filtering using the same standard deviation σ_{opt} and a support size equal to $(6 \times \sigma_{\mathrm{opt}} + 1) \times (6 \times \sigma_{\mathrm{opt}} + 1)$ (Table 1).

$$\Delta\mathrm{PSNR} = \text{AGK-based PSNR} - \text{Static PSNR} \tag{20}$$

According to Table 1, the adaptive Gaussian kernel (AGK) gives better PSNR results than the static filter. The difference is conspicuous and significant. Figure 11 shows the variation of the ΔPSNR with respect to the noise density increase.

Figure 12 represents an original image 'cameraman' followed by the noisy image and the filtering results respectively by the static and the AGK Gaussian filter. The used salt and pepper noise has a density of 10%.

As presented in Figure 12, zoomed in Figures 13, 14, 15 and 16, using the proposed AGK filter, the noise is efficiently removed and the image content, edges, and borders are well preserved.

Table 2 and Figures 15 and 16 show filtering results concerning a synthetic image belonging to the database. In the following table, we illustrate respectively the PSNR of the filtered image using the AGK filter and the static filter.

Processing real, binary, or synthetic gray-level images, Figure 16 demonstrates the efficiency of the proposed adaptive filter to clean noise and preserve sharp details.

4.2. Speckle noise

The speckle is a multiplicative noise described as follows:

$$Y(x,y) = S(x,y) \times N(x,y) \tag{21}$$

As presented in Figure 17, even against a multiplicative noise, the proposed method generates excellent results when compared to the static filtering and essentially save undamaged edges and borders.

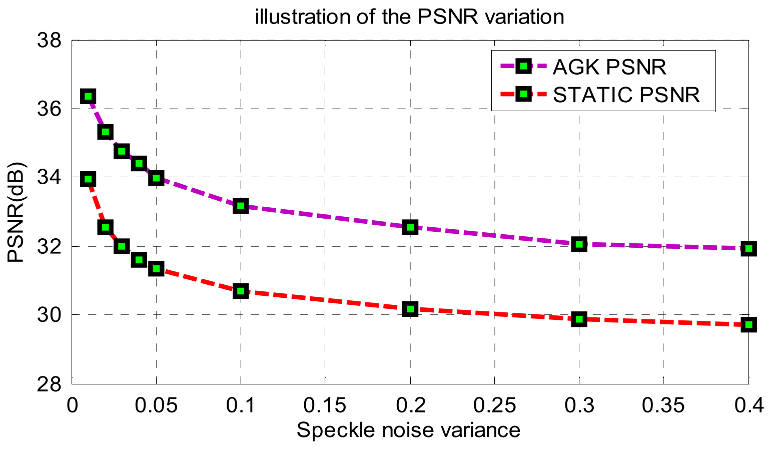


Figure 17 Comparative study between the proposed AGK and the static Gaussian filtering.

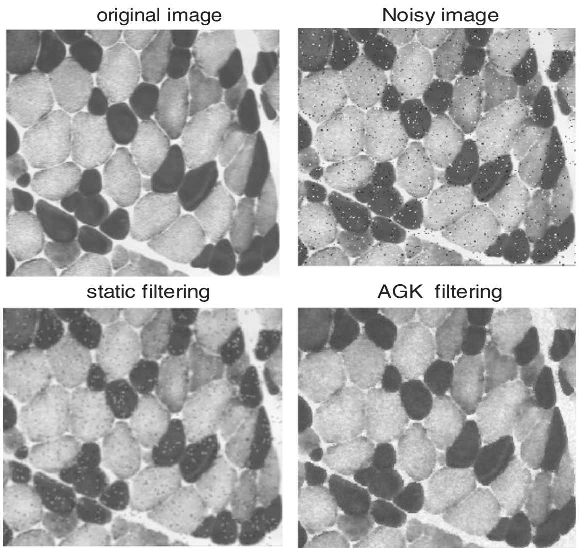


Figure 18 Comparison between the conventional and the AGK Gaussian filter.

Figure 18 represents a third example which illustrates the efficiency of the AGK filter.

The proposed AGK filter generates better de-noising results than the conventional filter. This efficiency was illustrated by different examples of filtered image. All the content and details were preserved and well perceptible after the filtering process.

5. Comparative study

As shown, experimentally, the proposed adaptive Gaussian kernel is efficient compared to the conventional filter. To prove the superiority of the proposed AGK filter, we conducted a comparative study between the AGK filter and other recent Gaussian techniques cited in the literature. For example, in [2], Eva and Aishy used the Gaussian filter for image de-noising where they studied the relation between Gaussian scale space and linear diffusion, and they derived an estimation of the Gaussian variance and window size; this technique is called AGSS. Table 3 represents the comparison between the proposed approach and this technique.

Table 3 Comparison results between the proposed technique, the conventional filter, and the AGSS filtering process

Images	AGSS technique	Static Gaussian filter	Proposed technique (AGK)	Δ*P*1	Δ*P*2
Boat	31.9	32.818	40.11	8.21	7.29
Lena	23.1	33.0982	37.957	14.86	4.86
House	28.7	33.2332	37.891	9.19	4.66
Barbara	35.5	32.7839	38.331	2.83	5.55
Peppers	20.3	33.6212	40.274	19.97	6.65

Gaussian noise (variance = 0.01). Δ*P*1 represents the PSNR difference computed between the AGK and the AGSS approach, whereas Δ*P*2 is the difference between the proposed approach and the static Gaussian filter. The computed PSNR of the filtered image using the proposed adaptive Gaussian filter is more important than those of the AGSS and the static Gaussian filter.

6. Conclusions

In this paper, a new low-pass filter with Gaussian core is presented. An adaptive Gaussian kernel based on dynamic core with variable structure is shaped. This new kernel conserves all the mathematical characteristics of the static Gaussian filter. The smoothing strength and the support size are supervised in each processed window of the image by a neural network to achieve the best filtering results. At the same time, the Gaussian lobe moves continuously in eight directions with the appropriate magnitude to avoid averaging of higher filter weights to preserve borders and edges. A comparative study is conducted to prove the efficiency of the proposed approach. Different image tests are shown with zoomed zones to validate the efficiency of this filter.

Competing interests
The author has no competing interests.

Received: 2 October 2013 Accepted: 18 February 2014
Published: 1 March 2014

References

1. SN Kota, G Umamaheswara Reddy, Fusion based Gaussian noise removal in the images using curvelets and wavelets with Gaussian filter. Int. J. Image Process (IJIP) **5**(4), 456–468 (2011)
2. A Amer, E Rifkah, Automated Gaussian filtering via Gaussian scaled and linear diffusion, in *20th European Signal Processing Conference (EUSIPCO)* (IEEE, Piscataway, 2012), pp. 1539–1542
3. H Seddik, Efficient noise removing based optimized smart dynamic Gaussian filter. Int. J. Comput. Appl. **51**, 5 (2012)
4. T Sondes, S Hassene, M Zouhair, BB Ezzedine, RGB Image De-Noising using New Low-Pas Filter with Variable Gaussian Core Real Time Optimized by Neural Networks, in *2013 International Conference on Electrical Engineering of Software Applications, ICEESA 2013* (Hammamet, 2013)
5. SK Kopparapu, M Satish, Identifying optimal Gaussian filter for Gaussian noise removal, in *Third IEEE Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2011, pp. 126–129
6. G Deng, LW Cahill, An adaptive Gaussian filter for noise reduction and edge detection. IEEE Conf. Rec. **3**, 1615–1619 (1993)

doi:10.1186/1687-6180-2014-25
Cite this article as: Seddik: A new family of Gaussian filters with adaptive lobe location and smoothing strength for efficient image restoration. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:25.