Adaptive average BM3D filter for reconstruction of images with combined noise

Isidora Stanković[1,2], Igor Djurović[1], Miloš Daković[1]
[1] Faculty of Electrical Engineering, University of Montenegro, Podgorica, Montenegro
[2] GIPSA Lab, University of Grenoble Alpes, Grenoble, France
{isidoras,igordj,milos}@ac.me

***Abstract*—A method for reconstruction of images corrupted by mixed, salt-and-pepper and Gaussian, noise is proposed. The adaptive average BM3D filter is a combination of two algorithms for the reconstruction of images corrupted with mixed noise. The two-stage adaptive average filter showed good results in the denoising of images corrupted by a salt-and-pepper noise, whereas the BM3D algorithm showed good results in denoising images corrupted by a Gaussian noise. The algorithm was compared to a state-of-the-art algorithm.**

***Keywords*— Adaptive average filter, BM3D, Denoising, Gaussian, Image processing, salt-and-pepper**

I. Introduction

In denoising of images, the main goal is to find a compromise between the complexity and the precision of the algorithm, for a successful reconstruction of corrupted pixels. The noise in images can be caused by various reasons, like having dead pixels in the images, obtained during transmission or some hardware problems of image sensors. It also can happen that some pixels are not available in the procedure of sampling. Denoising of images is still a hot topic with numerous denoising algorithms available, some of them described in [1]–[11].

The basic denoising techniques are the mean and the median filtering of images. They are based on windowing an image in smaller blocks and replacing the corrupted pixel with a mean/median value of the windowed block. The advantage of these methods is that they represent very simple techniques of image filtering. However, these methods are not robust for highly corrupted images.

In this paper, we present two state-of-the-art methods for denoising and their combination in order to successfully reconstruct images corrupted by a combined noise. The considered algorithms are the two-stage adaptive average filter from [7] and the block-matching 3D (BM3D) algorithms from [8]. The two-stage adaptive average filter is a modified mean filtering technique for denoising images corrupted by a salt-and-pepper noise. The BM3D is based on matching by using the collaborative filtering procedure and an inverse 3D transform of the filtered coefficients. It has been efficiently used for the reconstruction of images corrupted by a Gaussian noise.

The paper is organized as follows. In Section II, the problem formulation is presented. In Section III, the two denoising methods are briefly explained, along with the proposed combination. The comparison with the state-of-the art technique, also based on the BM3D, with respect to the structural similarity index and the peak-to-noise ratio, is analyzed in Section IV. Conclusions are given in Section V.

II. Problem formulation

The original, non-noisy, image will be denoted by $x(m,n)$, $m = 0,1,\dots,M-1, n = 0,1,\dots,N-1$. A noisy image will be defined as

$$y(m,n) = x(m,n) + \eta(m,n) \tag{1}$$

where $\eta(m,n)$ is a noise.

Here, we will focus on the two most common types of noise, which are the salt-and-pepper and the Gaussian noise. The density of salt-and-pepper noise depends on the percentage of affected pixels in the image, while the Gaussian noise is spread over the whole image and its intensity is usually described by the noise variance.

An 8-bit image pixel will have values between $\min\{x(m,n)\} = 0$ (black pixels) and $\max\{x(m,n)\} = 255$ (white pixels). An image affected by salt-and-pepper noise is then defined as

$$y(m,n) = \begin{cases} x(m,n) & \text{for } (m,n) \in \mathbb{N}_o \\ 0 \text{ or } 255, & \text{for } (m,n) \in \mathbb{N}_n \end{cases} \tag{2}$$

where $\mathbb{N}_o$ denotes the set of noncorrupted pixel positions and $\mathbb{N}_n$ is the set of noise positions in the image. Note that $\mathbb{N}_o$ and $\mathbb{N}_n$ are complementary sets, meaning that $\mathbb{N}_o \cup \mathbb{N}_n = \mathbb{N}$ and $\mathbb{N}_o \cap \mathbb{N}_n = \emptyset$, where $\mathbb{N} = \{(0,0),\dots,(M-1,N-1)\}$ is the complete set of pixel coordinates in the image.

An image affected by Gaussian noise is

$$y(m,n) = x(m,n) + \eta(m,n), \tag{3}$$

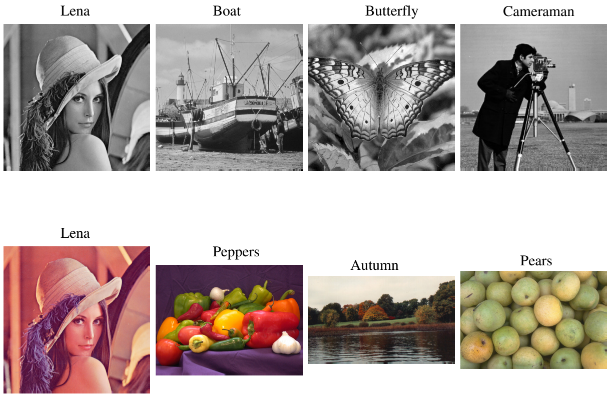

Fig. 1. Images used for analysis. Grayscale images (top row); Color images (bottom row)

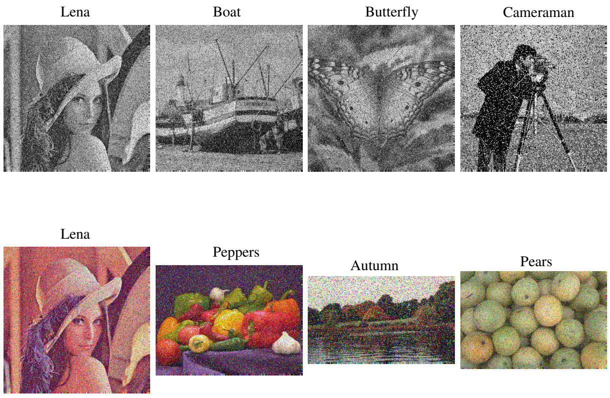

Fig. 2. Images affected by 20% salt-and-pepper and zero-mean Gaussian noise with $\sigma = 20$. Grayscale images (top row); Colour images (bottom row)

where $\eta(m,n) \sim \mathcal{N}(\mu,\sigma^2)$, with mean μ and variance σ^2 being a squared value of the standard deviation σ.

The test images are presented in Fig. 1. Four of them are grayscale images and the remaining four are color images. Images affected by 20% of the salt-and-pepper noise and a zero-mean Gaussian noise with $\sigma = 20$ are shown in Fig. 2.

III. Reconstruction Algorithm

The two-stage adaptive average filter is used for filtering pixels affected by the salt-and-pepper noise. The BM3D showed good results in reconstruction of image corrupted by a Gaussian noise. The idea is first to 'clean' the image from the salt-and-pepper noise and after that to reconstruct the image from Gaussian noise. The two-stage algorithm can be described as

1. $x_p(m,n) = \text{ 2SAA } \{y(m,n)\}$
2. $\hat{x}(m,n) = \text{BM3D}\{x_p(m,n)\}$,

where 2SAA$\{.\}$ presents the averaging filter. These two algorithms will be presented in the next two subsections.

A. Two-stage adaptive average filter

The two-stage adaptive average filter is a state-of-the-art mean filter for image denoising using adaptive averaging filters. It has been proposed in [7].

The algorithm is divided into two stages: the first one is in detecting the noisy pixels, and the second one is in filtering the images. Let us consider an image $y(m,n)$ which is corrupted by a salt-and-pepper noise. The noisy pixels are then of the values 0 or 255. When we detect the positions of the noisy pixels, a new matrix is made of the size as the original image, i.e. $M \times N$. The positions of the noisy pixels are zero-valued. For the consistency, the number of noisy pixels will be declared as N_n.

The level of noise is simply calculated as $D = N_n/(M \times N)$. The window size B which will be used depends on the noise level. It is calculated as

$$B = 2\left\lfloor\sqrt{\frac{2.2}{1-D}}\right\rfloor + 1 \quad (4)$$

where $\lfloor . \rfloor$ is the floor value. A matrix with local sums of each $B \times B$ of the non-noisy pixels, by summing them up using the convolution, is added. In each window, we find the non-noisy pixels and calculate their local averages. The noisy pixels are then the values of the rounded local averages.

B. Block-matching and 3D filtering (BM3D) algorithm

The BM3D algorithm has been introduced in [8]. The algorithm is based on two main steps: grouping and collaborative filtering.

The grouping step is done by using the d-dimensional blocks of image with a similar content. The blocks are grouped into $d+1$-dimensional arrays. For an image, as a 2D signal, we will group the blocks which are similar to a 3D array. Note that the sizes of blocks in a group are fixed. We take a reference block Y_R and search for blocks with similar contents. This is the block-matching (BM) part of the algorithm.

When similar blocks are found and grouped, they are filtered using a collaborative filtering procedure. The collaborative filtering means that each part collaborates for the filtering of all others, and vice versa. The collaborative filtering techniques, such as hard-thresholding or Wiener filtering, can be used. In this method, the filtering by shrinkage in the transform domain is used. After the filtering is done, the blocks are brought back to the 2D signal forms, at their original positions.

For the application to the color images, the images are converted to the YCbCr color space. Only the Y component is

Fig. 3. Images reconstructed with the proposed combined algorithm. Grayscale images (top row); Colour images (bottom row)

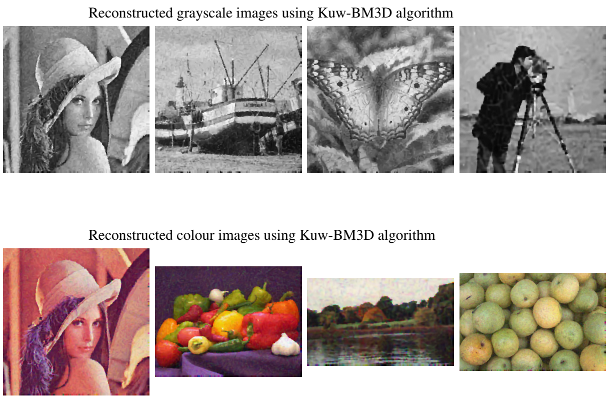

Fig. 4. Images reconstructed using the combination Kuwahara-BM3D algorithms. Grayscale images (top row); Colour images (bottom row)

used for the block-matching. It is assumed (and experimentally checked in [8], [11]) that it is highly probable that similar blocks in Y component will be similar in other Cb and Cr components. Filtering is then applied on all three components.

IV. Comparison

The comparison is done between the proposed algorithm (abbreviated "2SAA" in this section) and the Kuwahara-BM3D algorithm (abbreviated "Kuw") presented in [10], since they are both based on the BM3D filtering procedure and can be used for mixed noise. We consider the noisy images presented in Fig. 2. Their reconstructions by the proposed method are presented in Fig. 3. The images reconstructed using the Kuwahara-BM3D algorithm is presented in Fig. 4.

For the comparison, we use the structural similarity (SSIM) index and the peak-to-noise ratio (PSNR). We will denote the original image as $\mathbf{x}_o$, and the denoised image as $\mathbf{x}_r$.

The SSIM index is introduced in [12] and it is defined as

$$\mathrm{SSIM}(\mathbf{x}_o,\mathbf{x}_r)=\frac{(2\mu_{\mathbf{x}_o}\mu_{\mathbf{x}_r}+c_1)(2\sigma_{\mathbf{x}_o\mathbf{x}_r}+c_2)}{(\mu^2_{\mathbf{x}_o}+\mu^2_{\mathbf{x}_r}+c_1)(\sigma^2_{\mathbf{x}_o}+\sigma^2_{\mathbf{x}_r}+c_2)}, \quad (5)$$

where $\mu_{\mathbf{x}_o}\mu_{\mathbf{x}_r}$ are the mean values of the images, $\sigma_{\mathbf{x}_o\mathbf{x}_r}$ is the covariance, $\sigma^2_{\mathbf{x}_o}$ and $\sigma^2_{\mathbf{x}_r}$ are the variances, and c_1 and c_2 are stabilization constants. The SSIM value close to 1 represents very similar images and 0 is for not similar.

The PSNR for an 8-bit image is calculated as

$$\mathrm{PSNR}(\mathbf{x}_o,\mathbf{x}_r)=10\log_{10}\left(\frac{255^2}{\mathrm{MSE}(\mathbf{x}_o,\mathbf{x}_r)}\right), \quad (6)$$

where 255 is maximal pixel value and $\mathrm{MSE}(\mathbf{x}_o,\mathbf{x}_r)=\mathrm{mean}(|\mathbf{x}_o-\mathbf{x}_r|^2)$ is the mean square error of the image.

The comparison of considered algorithms using seven different grayscale images with fixed noise parameters (20% of the salt-and-pepper noise and a Gaussian noise with standard deviation $\sigma=20$) is shown in Tables I and II. Table I shows the SSIM values and Table II shows the PSNR values for these images. The numbers present these images in the following order: 1–Lena, 2–Boat, 3–Butterfly, 4–Pout, 5–Cameraman, 6–Lifting body, 7–Pirate. All images are from the standard MATLAB database. The bold values present the algorithm with better performance.

TABLE I
SSIM comparison for of different images when 20% salt-and-pepper noise and Gaussian with $\sigma=20$

Image	1	2	3	4	5	6	7
Kuw	0.76	0.75	0.76	0.75	0.65	0.74	0.67
2SAA	**0.92**	**0.91**	**0.92**	**0.88**	**0.82**	**0.91**	**0.80**

TABLE II
PSNR comparison for of different images when 20% salt-and-pepper noise and Gaussian with $\sigma=20$

Image	1	2	3	4	5	6	7
Kuw	26.6	25.1	25.4	28.9	22.4	28.3	23.4
2SAA	**32.1**	**30.5**	**30.0**	**34.7**	**27.3**	**34.0**	**26.5**

The SSIM comparison between the two algorithms with various noise levels is shown in Table III and the PSNR comparison is presented in Table IV. Note that the percentage represents the percentage of salt-and-pepper noise in the image and σ is the standard variation used for Gaussian noise.

Also note that in Tables III and IV the value of the standard deviation σ is known to the BM3D algorithm, and it reconstructs the images accordingly. In Tables V and VI, the standard deviation of noise σ is unknown to the BM3D algorithm. Therefore, it uses a predefined value for σ (for each reconstruction). In our calculation, the value is $\sigma=25$.

Few remarks about the performances of the two methods:

TABLE III
SSIM VALUES FOR GRAYSCALE IMAGE "LENA" FOR DIFFERENT NOISE LEVELS (BM3D ASSUMES KNOWN STANDARD DEVIATION)

	$\sigma = 10$		20		30		40	
%	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**
5	0.92	**0.97**	0.89	**0.94**	0.86	**0.90**	0.81	**0.86**
10	0.89	**0.97**	0.87	**0.93**	0.84	**0.89**	0.80	**0.86**
15	0.83	**0.96**	0.84	**0.93**	0.81	**0.89**	0.76	**0.85**
20	0.74	**0.96**	0.76	**0.92**	0.76	**0.88**	0.73	**0.84**
30	0.54	**0.96**	0.61	**0.91**	0.60	**0.87**	0.59	**0.81**
40	0.41	**0.95**	0.47	**0.89**	0.48	**0.84**	0.48	**0.77**

TABLE IV
PSNR VALUES FOR GRAYSCALE IMAGE "LENA" FOR DIFFERENT NOISE LEVELS (BM3D ASSUMES KNOWN STANDARD DEVIATION)

	$\sigma = 10$		20		30		40	
%	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**
5	29.4	**35.6**	29.4	**32.8**	28.6	**30.7**	27.7	**28.3**
10	28.2	**35.4**	28.6	**32.5**	28.0	**30.5**	27.3	**28.3**
15	26.8	**35.1**	27.6	**32.4**	27.3	**30.2**	26.3	**28.1**
20	25.2	**34.7**	26.6	**32.1**	26.3	**30.0**	25.9	**27.8**
30	21.8	**34.1**	23.5	**31.6**	23.8	**29.6**	23.4	**27.4**
40	19.3	**33.0**	20.9	**30.9**	21.6	**29.0**	21.6	**26.9**

1. The two-stage adaptive average filter denoises only the salt-and-pepper pixels, while the adaptive Kuwahara method filters some of the Gaussian noise as well. This is the reason why, with a fixed and predefined value of σ in the BM3D, the Kuwahara-BM3D method performs better. The noise after Kuwahara filtering is not the same as at the beginning of the filtering procedure. Since the two-stage adaptive average filter denoises impulses only, better results are obtained when the level of Gaussian noise is known in the BM3D algorithm.

2. In both cases of noise, the two-stage adaptive average filter, combined with the BM3D, produces better results in most of the analyzed images.

3. The Kuwahara-BM3D produces better results in the case when the image is corrupted with a high Gaussian noise and a low percentage of salt-and-pepper noise (for example, $\sigma = 40$ and 5% up to 20% of salt-and-pepper noise), and when a predefined standard deviation is used. The reason is explained in Remark 1.

V. CONCLUSIONS

A method for combining two algorithms for denoising images corrupted with a mixed, salt-and-pepper and Gaussian, noise is proposed. The method is a combination of an averaging filter with the BM3D algorithm. It is shown that this combination gives good results in denoising of noisy images.

TABLE V
SSIM VALUES FOR GRAYSCALE IMAGE "LENA" FOR DIFFERENT NOISE LEVELS (BM3D ASSUMES STANDARD DEVIATION OF $\sigma = 25$)

	$\sigma = 10$		20		30		40	
%	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**
5	0.91	**0.95**	0.89	**0.93**	0.85	**0.89**	**0.79**	0.71
10	0.90	**0.95**	0.87	**0.93**	0.83	**0.88**	**0.76**	0.71
15	0.87	**0.95**	0.84	**0.93**	0.80	**0.87**	**0.71**	0.71
20	0.83	**0.95**	0.81	**0.93**	0.74	**0.86**	0.67	**0.70**
30	0.69	**0.94**	0.64	**0.92**	0.58	**0.84**	0.54	**0.67**
40	0.51	**0.94**	0.49	**0.91**	0.46	**0.80**	0.46	**0.64**

TABLE VI
PSNR VALUES FOR GRAYSCALE IMAGE "LENA" FOR DIFFERENT NOISE LEVELS (BM3D ASSUMES STANDARD DEVIATION OF $\sigma = 25$)

	$\sigma = 10$		20		30		40	
%	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**	**Kuw**	**2SAA**
5	30.0	**33.4**	29.4	**32.4**	28.6	**30.2**	**27.4**	24.5
10	29.3	**33.3**	28.7	**32.3**	27.9	**30.2**	**26.7**	24.9
15	28.3	**33.1**	27.7	**32.1**	27.1	**30.0**	**25.9**	25.2
20	27.1	**32.8**	26.9	**31.9**	26.0	**29.8**	25.1	**25.3**
30	24.5	**32.5**	24.2	**31.4**	23.4	**29.1**	22.4	**25.4**
40	22.0	**32.1**	21.6	**31.0**	21.0	**28.4**	20.4	**24.9**

REFERENCES

[1] P. Chatterje, and P. Milanfar, "Is denoising dead?," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, 2010.

[2] R.C. Gonzalez, and R.E. Woods, *Digital Image Processing.* 2nd edition, Prentice Hall, 2002.

[3] A.C. Bovik, *Handbook of Image and Video Processing.* Academic Press, 2000.

[4] A. Pižurica, "Image Denoising Algorithms: From Wavelet Shrinkage to Non-local Collaborative Filtering," *Wiley Encyclopedia of Electrical and Electronics Engineering*, February 2017.

[5] M. Elad, and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transasctions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, December 2006.

[6] I. Stanković, I. Orović, M. Daković, and S. Stanković, "Denoising of Sparse Images in Impulsive Disturbance Environment," *Multimedia Tools and Applications*, 2017, https://doi.org/10.1007/s11042-017-4502-7

[7] Z. Ramadan, "Efficient Restoration Method for Images Corrupted with Impulse Noise" *Circuits, Systems, and Signal Processing*, vol. 31, no. 4, pp. 1397–1406, 2012.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, August 2007.

[9] K. Bartyzel, "Adaptive Kuwahara filter," *Signal, Image and Video Processing*, July 2015, DOI: 10.1007/s11760-015-0791-3

[10] I. Djurović, "Combination of the adaptive Kuwahara and BM3D filters for filtering mixed Gaussian and impulsive noise," *Signal, Image and Video Processing*, vol. 11, no. 4, pp. 753–760, May 2017.

[11] M. Lebrun, "An Analysis and Implementation of the BM3D Image Denoising Method," Image Processing On Line, 2012.

[12] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.