Noise Reduction Techniques for Processing of Medical Images

Luis Cadena, Alexander Zotin, Franklin Cadena, Anna Korneeva, Alexander Legalov, Byron Morales

***Abstract* — An application of different techniques for processing medical images is presented in this paper. The proposed of the techniques consists of application of digital spatial filters. Digital filters are used which in the last decades have taken great impetus for the treatment of images in different fields of science and in the case of the medical image processing better results are obtained in order to make better interpretations.**

Medical images can contain some noise therefore it makes sense to suppress noise on preprocessing stage. The algorithms of most often used filters have been considered, such as mean filter, Gaussian filter, median filter and 2d Cleaner. With the trend toward larger images and proportionally larger filter kernels, the need for a more efficient filtering algorithm becomes pressing. Conducted comparison of optimized and classical implementations of filters algorithms. It shows great speed improvement of optimized implementation.

***Index Terms*— medical image, image processing, denoising, mean filter, median filter, Gaussian 2D filter, 2DCleaner filter**

I. INTRODUCTION

Digital image processing consists of algorithmic processes that transform one image into another in which certain information of interest is highlighted, and/or the information that is irrelevant to the application is attenuated or eliminated. Thus, image processing tasks include noise suppression, contrast enhancements, removal of undesirable effects on capture such as blurring or distortion by optical or motion effects, color transformations, and so on.

Filtering is a technique for modifying and enhancing an image. Various filters are used for image preprocessing. The primary purpose of these filters is a noise reduction, but filter can also be used to emphasize certain features of an image or remove other features. In image processing, 2D filtering techniques are usually considered an extension of 1D signal processing theory.

Manuscript received February 26 2017; This work was supported by Universidad de las Fuerzas Armadas ESPE, Av. Gral Ruminahui s/n, Sangolqui Ecuador

L. Cadena is with Electric and Electronic Department, Universidad de las Fuerzas Armadas ESPE, Av. Gral Ruminahui s/n, Sangolqui Ecuador. (phone: +593997221212; e-mail: ecuadorx@gmail.com; lrcadena@espe.edu.ec).

A. Zotin is with Department of Informatics and Computer Techniques, Reshetnev Siberian State Aerospace University, 31 krasnoyarsky rabochy pr., Krasnoyarsk 660014, Russia Federation (e-mail: zotinkrs@gmail.com)

F. Cadena is with College Juan Suarez Chacon, Quito, Ecuador (e-mail: fcfc041@gmail.com)

A. Korneeva is Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia (e-mail: korneeva_ikit@mail.ru)

A. Legalov is Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia (e-mail: alexander.legalov@gmail.com)

B. Morales is Universidad de las Fuerzas Armadas ESPE, Av. Gral Ruminahui s/n, Sangolqui Ecuador. (e-mail: bomorales1@espe.edu.ec)

Almost all contemporary image processing involves discrete or sampled signal processing.

Most of image processing filters can be divided into two main categories: linear filters and nonlinear filters. Nonlinear filters include order statistic filters and adaptive filters. The choice of filter is often determined by the nature of the task and the type and behavior of the data. Noise, dynamic range, color accuracy, optical artifacts, and many more details affect the outcome of filter in image processing [13-17].

II. FILTERS

Filtering is a technique for modifying and enhancing an image. Various filters are used for image preprocessing. The primary purpose of these filters is a noise reduction, but filter can also be used to emphasize certain features of an image or remove other features. In image processing, 2D filtering techniques are usually considered an extension of 1D signal processing theory. Almost all contemporary image processing involves discrete or sampled signal processing. Most of image processing filters can be divided into two main categories [1-3]: linear filters and nonlinear filters. Nonlinear filters include order statistic filters and adaptive filters. The choice of filter is often determined by the nature of the task and the type and behavior of the data. Noise, dynamic range, color accuracy, optical artifacts, and many more details affect the outcome of filter in image processing.

One of the simplest linear filters is a filter which calculates arithmetic mean value of spectrum. The arithmetic mean filter is defined as the average of all pixels spectrum within a local region of an image. Pixels that are included in the averaging operation are specified by a mask (Kernel). Kernel size can be different and depends on task. An arithmetic mean filter operation on an image removes short tailed noise such as uniform and Gaussian type noise from the image at the cost of blurring the image. Mathematically mean filter can be described as follows:

$$\mathbf{C}_{new}(y,x) = \frac{1}{(RH \times 2+1)\times(RW \times 2+1)} \sum_{dy=-RH}^{RH} \sum_{dx=-RW}^{RW} \mathbf{C}_{old}(y+dy, x+dx)$$

where C_{new}, C_{old} – new and old values of the image pixels spectrum, respectively; RH, RW – constants defining the rank of the filter vertically and horizontally.

Pixels that are included in the averaging operation are specified by a kernel. The larger the filtering kernel becomes the more predominant the blurring becomes and less high spatial frequency detail that remains in the image.

In the case of using the descriptions in the form of convolution filter the computation takes the following form:

$$\mathbf{C}_{new}(y,x) = \sum_{dy=-RH}^{RH} \sum_{dx=-RW}^{RW} A_{dy+RH,dx+RW} \times \mathbf{C}_{old}(y+dy, x+dx)$$

where A – filter kernel.

Defining the K_{Hs} as vertical kernel size (K_{Hs}=RH×2+1) and K_{Ws} as horizontal kernel size (K_{Ws}=RW×2+1). The kernel coefficients of mean filter are calculated according to formula:

$$A_{i,j} = \frac{1}{K_{Hs} \times K_{Ws}}$$

Due to arithmetic mean filter property of using equal weights it can be implemented using a much simpler accumulation algorithm which is significantly faster than using a sliding window algorithm. Thus the accumulation of the neighborhood of pixel P(y,x), shares a lot of pixels in common with the accumulation for pixel P(y,x+1). This means that there is no need to compute the whole kernel for all pixels except only the first pixel in each row [4]. Successive pixel filter response values can be obtained with just an add and a subtract to the previous pixel filter response value. Thus, the filter computation can be considered the following way:

$$C_{new}(y,x) = \begin{cases} \frac{1}{K_{Hs} \times K_{Ws}} \sum_{dy=-RH}^{RH} \sum_{dx=-RW}^{RW} C_{old}(y+dy, x+dx), & \text{if } x = 0 \\ C_{new}(y,x-1) - \frac{1}{K_{Hs}} \sum_{dy=-RH}^{RH} C_{old}(y+dy, x-RW-1) + \frac{1}{K_{Hs}} \sum_{dy=-RH}^{RH} C_{old}(y+dy, x+RW), & \text{otherwise} \end{cases}$$

Figure 1 shows difference in processing time of classical and optimized variant of mean filter on image with size 512×512. Comparison of filters we conducted on following PC: Intel core i5 3.1GHz 8 GB RAM.

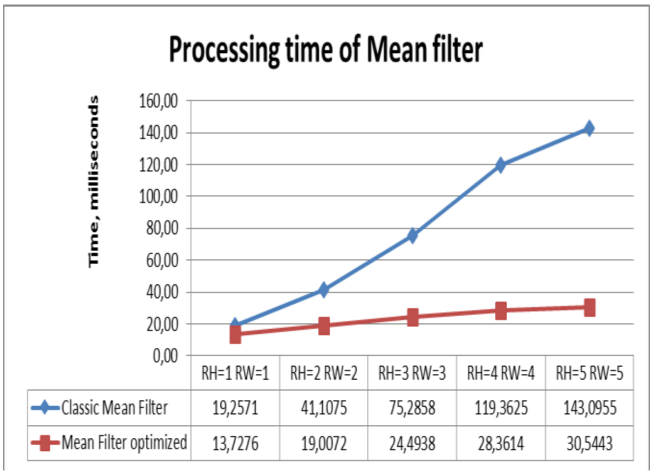


Fig. 1. Comparison of classical and optimized implementation of mean filter

The Gaussian filter (also known as Gaussian blur) is a smoothing filter, which used to blur images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel. The Gaussian filter uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. The equation of a Gaussian function in one dimension is

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

in two dimensions, it is the product of two such Gaussians, one in each dimension:

$$G(x,y) = \frac{1}{2\pi \cdot \sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

Since the image is represented as a collection of discrete pixels it is necessary to produce a discrete approximation to the Gaussian function before perform the convolution. Depends on kernel size and σ some of coefficients can be out range of kernel. Theoretically the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel. In practice it is effectively zero more than about three standard deviations from the mean. Thus it is possible to truncate the kernel size at this point. Sometimes kernel size truncated even more. Thus after computation of Gaussian Kernel, the coefficients must be corrected that way that the sum of all coefficients equals 1. Once a suitable kernel has been calculated, then the Gaussian smoothing can be performed using standard convolution methods. The convolution can in fact be performed fairly quickly since the equation for the 2-D isotropic Gaussian is separable into y and x components [5] (Figure 2). In some cases the approximation of Gaussian filter can be used instead of classic version [6,7].

0,004432
0,053991
0,241971
0,398942
0,241971
0,053991
0,004432

x

0,004432	0,053991	0,241971	0,398942	0,241971	0,053991	0,004432

=

0,00002	0,00024	0,00107	0,00177	0,00107	0,00024	0,00002
0,00024	0,00292	0,01306	0,02154	0,01306	0,00292	0,00024
0,00107	0,01306	0,05855	0,09653	0,05855	0,01306	0,00107
0,00177	0,02154	0,09653	0,15915	0,09653	0,02154	0,00177
0,00107	0,01306	0,05855	0,09653	0,05855	0,01306	0,00107
0,00024	0,00292	0,01306	0,02154	0,01306	0,00292	0,00024
0,00002	0,00024	0,00107	0,00177	0,00107	0,00024	0,00002

Fig. 2. Isotropic Gaussian is separable into y and x components

Difference in processing time of classical 2D and double 1D implementations of Gaussian filter shown on Figure 3.

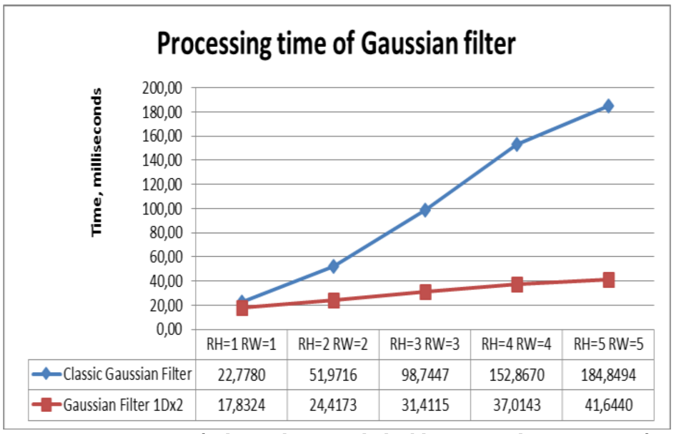


Fig. 3. Comparison of classical 2D and double 1D implementation of Gaussian filter

The best-known order-statistics filter is the median filter, which, as its name implies, replaces the value of a pixel spectrum by the median of the spectrum levels in the neighborhood of that pixel:

$$\mathbf{C}_{new}(y,x) = \underset{(ky,kx)\in K_{xy}}{\text{med}} \{\mathbf{C}_{old}(ky,kx)\}$$

where K – kernel window, with dimensions K_{Hs}×K_{Ws} centered at (x, y).

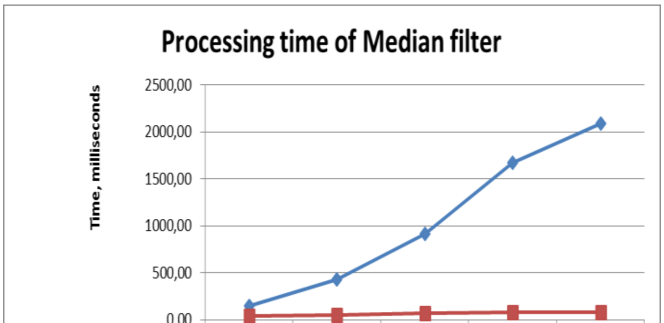
The original value of the pixel is included in the computation of the median. Median filters are quite popular

ISBN: 978-988-14047-4-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. It is particularly useful in removing speckle and salt and pepper noise. The pattern of neighbors is defined by kernel called the "window", which slides, entry by entry, over the entire signal. Usually kernel size of median filter has an odd number of entries, because it is simple to define: it is just the middle value after all the entries in the kernel are sorted numerically.

The majority of the computational effort and time is spent on calculating the median of kernel. Because the filter must process every pixel in the image, for large images, the efficiency of this median calculation is a critical factor in determining how fast the algorithm can run. The classic implementation involves sorting of every entry in the kernel to find the median. However since only the middle value in a list of numbers is required, for median filter can be used much more efficient selection algorithms [8]. Furthermore in image processing the histogram of spectrum for median calculation can be far more efficient because it is simple to update the histogram from window to window, and finding the median of a histogram is not particularly onerous [9-11].

Comparison of processing time of classical and histogram based (optimized) implementations of median filter is shown on Figure 4.



	RH=1 RW=1	RH=2 RW=2	RH=3 RW=3	RH=4 RW=4	RH=5 RW=5
Classic Median Filter	147,8318	427,1835	917,7736	1669,5701	2089,2057
Median Filter optimized	37,7357	51,7239	65,3639	76,5620	80,3026

Fig. 4. Comparison of classical and histogram based (optimized) implementations of median filter

One of adaptive filters for noise reduction is 2D Cleaner by Jim Casaburi [12]. It is often used in video processing. The main idea of filter is calculation of arithmetic mean value in each color channel if it's deviation from

$$C_{new}(s,y,x,Ts)=\frac{\sum_{dy=-RH}^{RH}\sum_{dx=-RW}^{RW}Sv_s(y+dy,x+dx,TS)}{\sum_{dy=-RH}^{RH}\sum_{dx=-RW}^{RW}Cc_s(y+dy,x+dx,TS)},\quad s\in\{Red,Green,Blue\}$$

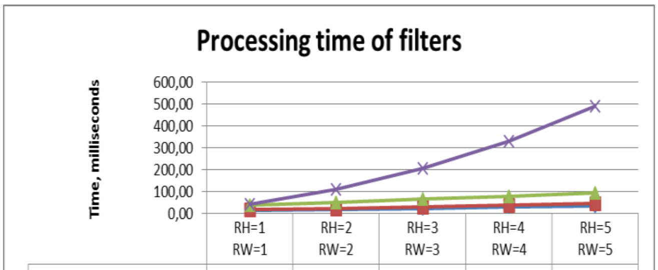
$Sv_s(i,j,TS)$ – spectrum cut-off function on the threshold value; $Cc_s(i,j,TS)$ – function indicates the suitability of spectrum according the threshold value.

$$Sv_s(i,j,TS)=\begin{cases}sv_s(i,j), & if\ |sv_s(i,j)-sv_s(0,0)|\le Ts\\ 0\ , & if\ |sv_s(i,j)-sv_s(0,0)|>Ts\end{cases}$$

$$Cc_s(i,j,TS)=\begin{cases}1, & if\ |sv_s(i,j)-sv_s(0,0)|\le Ts\\ 0\ , & if\ |sv_s(i,j)-sv_s(0,0)|>Ts\end{cases}$$

where $sv_s(i,j)$ – the value of spectrum considered a color channel; Ts – the threshold value.

Estimation of processing time of optimized filtering algorithms (Mean filter, Median filter, Gaussian Filter) and 2D cleaner, for different kernel size is shown on Figure 5. The comparison of time-consuming for processing image by filters using kernel size 5×5 (RH=2 RW=2) is shown on graph (Figure 6). There were taken images with different size (512×512 – 1920×1080) for experimental research.

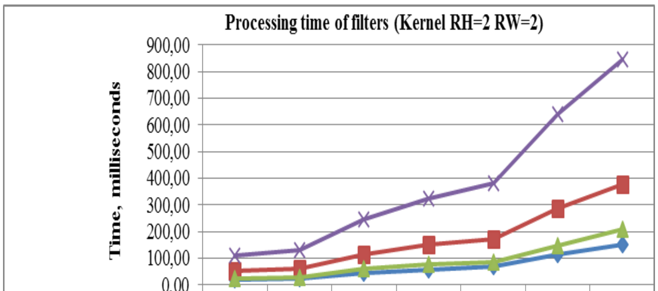


	RH=1 RW=1	RH=2 RW=2	RH=3 RW=3	RH=4 RW=4	RH=5 RW=5
Mean Filter optimized	13,7276	19,0072	24,4938	30,3614	36,4911
Gaussian Filter 1Dx2	17,8324	24,4173	31,4115	38,0143	45,0974
Median Filter optimized	37,7357	51,7239	65,3639	80,5620	94,3840
2D Cleaner Filter	43,6059	110,7796	205,6099	332,5104	491,8004

Fig. 5. Estimation of processing time of optimized filtering algorithms (Mean filter, Median filter, Gaussian Filter) and 2D cleaner, for different kernel size

III. EXPERIMENTAL RESULTS OF NOISE REDUCTION

A medical image was processed with the filters: 2D Cleaner, Gaussian 2D Filter, Mean Filter, Median Filter, with kernels: 3×3, 5×5, 7×7, 9×9, 11×11.



	512x512 (0,26 MP)	640x480 (0.31 MP)	768x768 (0.59 MP)	1024x768 (0.79 MP)	1280x720 (0.92 MP)	1440x1080 (1.56 MP)	1920x1080 (2.07 MP)
Mean Filter optimized	19,2838	22,4691	42,8924	56,9953	66,9512	111,9140	149,0202
Gaussian Filter 1Dx2	51,7341	60,0419	113,4086	149,7494	173,3290	287,8208	376,4876
Median Filter optimized	23,9961	29,6185	59,5255	77,1208	86,5259	146,6886	207,3144
2D Cleaner Filter	111,0410	128,6128	244,9082	324,6816	379,0070	637,9722	846,5816

Fig. 6. Comparison of time-consuming for processing image by filters using kernel size 5×5 (RH=2 RW=2)

In table 1 PSNR (dB) values for 5%, 10%, 15% and 20% of additive noise reduction with different filters are shown. To simulate the noise that may occur in the equipment, it was decided to use the following noise characteristic. The total noise map share of impulse noise is 20%, and the additive noise – 80%. The magnitude of the noise component of the additive is from 5 to 15% of the dynamic range of the data examined. Obtained results are shown in table 2. Figure 7 shows results of 10% additive noise reduction by 2D Cleaner filter with Threshold 10 for different kernel size.

ISBN: 978-988-14047-4-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I.
ADDITIVE NOISE REDUCTION PSNR (DB) VALUES FOR FILTERS: 2D CLEANER, GAUSS 2D, MEAN, MEDIAN FOR 3×3, 5×5, 7×7, 9×9 AND 11×11 KERNELS.

Noise level	Kernel size	Filter			
		Mean	Gaus	Median	2D Cleaner
5 %	3×3	53,492	55,259	54,46	77,901
	5×5	48,399	52,213	47,952	78,769
	7×7	46,6	51,847	46,735	79,452
	9×9	45,393	51,809	45,832	79,925
	11×11	44,457	51,808	45,101	80,231
10 %	3×3	53,452	55,208	54,386	74,275
	5×5	48,387	52,188	47,937	74,987
	7×7	46,593	51,823	46,721	75,575
	9×9	45,387	51,788	45,822	75,955
	11×11	44,45	51,786	45,099	76,184
15 %	3×3	53,404	55,147	54,327	71,852
	5×5	48,38	52,164	47,963	72,522
	7×7	46,59	51,8	46,727	73,019
	9×9	45,384	51,764	45,824	73,342
	11×11	44,448	51,762	45,094	73,527
20%	3×3	53,357	55,089	54,296	69,776
	5×5	48,365	52,134	47,953	70,4
	7×7	46,58	51,773	46,72	70,868
	9×9	45,377	51,737	45,821	71,157
	11×11	44,44	51,735	45,093	71,299

TABLE II.
COMPLEX NOISE REDUCTION PSNR (DB) VALUES FOR FILTERS: 2D CLEANER, GAUSS 2D, MEAN, MEDIAN FOR 3×3, 5×5, 7×7, 9×9 AND 11×11 KERNELS.

Noise level	Kernel size	Filter			
		Mean	Gaus	Median	2D Cleaner
5 %	3×3	52,225	53,293	54,354	50,487
	5×5	48,116	50,728	50,835	50,926
	7×7	46,478	50,25	47,924	50,995
	9×9	45,322	50,165	46,829	51,075
	11×11	44,405	50,155	45,102	51,088
10 %	3×3	50,998	51,88	54,309	44,288
	5×5	47,779	50,168	50,221	44,314
	7×7	46,286	49,8	47,709	44,434
	9×9	45,187	49,732	46,806	44,446
	11×11	44,302	49,724	45,081	44,354
15 %	3×3	49,855	50,596	54,135	40,767
	5×5	47,409	49,567	50,106	40,789
	7×7	46,068	49,311	47,911	40,885
	9×9	45,031	49,245	45,808	40,915
	11×11	44,178	49,237	45,085	40,943
20%	3×3	48,822	49,452	54,035	38,321
	5×5	47,035	48,981	48,89	38,340
	7×7	45,868	48,808	46,69	38,424
	9×9	44,884	48,767	45,799	38,462
	11×11	44,046	48,764	45,079	38,398

IV. CONCLUSIONS

Digital filters were used which in the last decades have taken great impetus for the treatment of images in different fields of science and in the case of the medical urology image processing better results are obtained in order to make better interpretations.

Various filters are used for medical image preprocessing such as mean filter, Gaussian filter, median filter and 2D Cleaner. The primary purpose of these filters is a noise reduction, but filter can also be used to emphasize certain features of an image or remove other features. Most of image processing filters can be divided into linear filters and nonlinear filters. Nonlinear filters include order statistic filters and adaptive filters. The choice of filter is often determined by the nature of the task and the type and behavior of the data.

Experimental results show that the optimized version of filter algorithms can well do with the relationship between the effect of the noise reduction and the time complexity of the algorithms. Thus for additive noise with low magnitude good results show 2D cleaner filter and Gaussian filter. The best noise reduction rate for complex noise was obtained by median filter with small Kernel 3×3, 5×5.

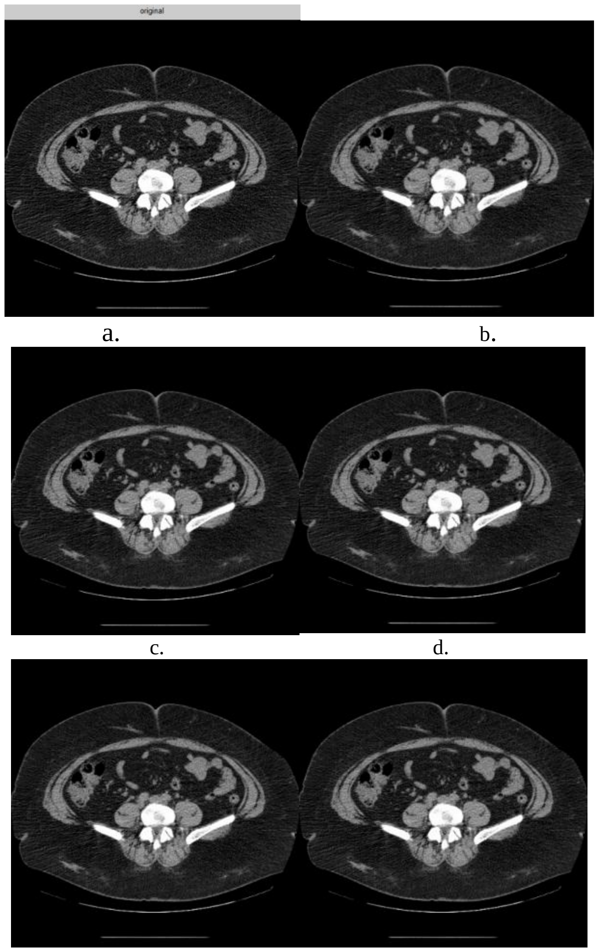


Fig. 7. Results for filter 2D Cleaner – Threshold 10. Original (a): - Kernels: 3×3 (b), 5×5 (c), 7×7 (d), 9×9 (e), 11×11 (f)

ACKNOWLEDGMENT

Very thanks to Feodor Petrovich Kapsargin professor from Medical Academy of Krasnoyarsk city Russia for the urology images.

REFERENCES

[1] Gonzalez RC, Woods RE. Digital Image Processing 3rd edition, Prentice-Hall, 2008. ISBN-13: 978-0131687288, (2008)

[2] Chandel et al. Image Filtering Algorithms and Techniques: A Review // International Journal of Advanced Research in Computer Science and Software Engineering 3(10), pp. 198-202, (2013)

[3] Gupta B, Singh Negi S Image Denoising with Linear and Non-Linear Filters: A REVIEW // International Journal of Computer Science Issues, Vol. 10, Issue 6, No 2, pp. 149-154, (2013)

[4] Lukin A Tips & Tricks: Fast Image Filtering Algorithms. 17-th International Conference on Computer Graphics GraphiCon'2007: 186–189, (2007).

[5] Pascal G. A Survey of Gaussian Convolution Algorithms. Image Processing On Line 3: 286–310, (2013)

[6] Young IT, Van Vliet LJ. Recursive implementation of the Gaussian filter. Elsevier Signal Processing 44: 139–151,(1995)

[7] Zing A. Extended Binomial Filter for Fast Gaussian Blur. Vienna, Austria, (2010)

[8] Suomela J. Median Filtering is Equivalent to Sorting, (2014). available http://users.ics.aalto.fi/suomela/median-filter/ Accessed 10 October 2016

[9] Weiss B. Fast Median and Bilateral Filtering. ACM Transactions on Graphics (TOG) 25 (3): 519–526, (2006)

[10] Cline D, White KB, Egbert PK. Fast 8-bit median filtering based on separability. In Image Processing ICIP 2007 IEEE International Conference 5: V-281–V-284, (2007)

[11] Perreault S, Hebert P. Median filtering in constant time. IEEE Transactions on Image Processing 16(9): 2389–2394, (2007)

[12] Casaburi's J free software download page available http://home.earthlink.net/~casaburi/download/ Accessed 14 October 2016

[13] Davies E. Machine Vision: Theory, Algorithms and Practicalities, Academic Press, (2012)

[14] Szeliski R. Computer vision. Algorithms and applications. Springer-Verlag London Limited, (2011)

[15] Ramesh J, Rangachar K, Brian G Schunck. Machine Vision. McGraw-Hill, Inc., ISBN 0-07-032018-7, (1995)

[16] Luis Cadena; Nikolai Espinosa; Franklin Cadena; Ramiro Rios; Konstantin Simonov, et al. "Contour detect in the medical image by shearlet transform ", Proc. SPIE 9524, International Conference on Optical and Photonic Engineering (icOPEN 2015), 95242W (July 17, 2015); doi:10.1117/12.2189725; available http://dx.doi.org/10.1117/12.2189725

[17] Luis Cadena; Franklin Cadena; Konstantin Simonov; Alexander Zotin and Grigory Okhotnikov. "Image compression algorithm using wavelet transform", Proc. SPIE 9971, Applications of Digital Image Processing XXXIX, 99711L (September 28, 2016); doi:10.1117/12.2235583; available http://dx.doi.org/10.1117/12.2235583

ISBN: 978-988-14047-4-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)