Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 483 – 486

ISBN 978-83-60810-14-9
ISSN 1896-7094

Similar Neighborhood Criterion for Edge Detection in Noisy and Noise-Free Images

Ali Awad and Hong Man
Stevens Institute of Technology Hoboken, NJ 07033 USA
Email: {aawad, hman}@stevens.edu}

***Abstract* — A novel approach for edge detection in noise-free and noisy images is presented in this paper. The proposed method is based on the number of similar pixels that each pixel in the image may have amongst its neighboring in the filtering window and within a pre-defined intensity range. Simulation results show that the new detector performs well in noise-free images but superior in corrupted images by salt and pepper impulse noise. Moreover, it is time efficient .**

I. Introduction

THERE are many techniques in the literature used for edge detection some of them are based on error minimization [1], maximizing an object function [2], fuzzy logic [3], genetic algorithms [4], neural network[5], and Bayesian approach[6]. But the most popular approaches are the gradient- based filters such as Sobel filter [7], and Canny method [8]. However, they show unsatisfactory performance in noisy images. In this paper, we present a new method based on the similarity criteria by which any pixel in the image has a specific number of similar pixels in the filtering window and within a predefined intensity range is labeled as an edge point. In this approach, we say that pixel y is similar to pixel x if the absolute intensity difference between x and y is $\leq D$. Where D is a pre-defined intensity value represents the maximum intensity difference between any two similar pixels. It is clear that, the edge pixel has a large intensity differences with its neighboring pixels [9]. Therefore, for a pre-defined value of D we find that the similar pixels number of an edge pixel is small compared to that of a pixel located in a smoothing area. Thus, we can say that the location of each pixel in the image is specified by two factors. 1- The intensity difference of the pixel with its neighbors. 2- The number N of similar pixel that any pixel in the image may have within the intensity range of $[0, D]$. As a result, we divide the pixel location into two sectors the first one includes the edge pixels and the second one contains the smoothing areas pixels. The general characteristics of the pixels in the first division are 1- They are very small number compared to the total number of the pixels in the image. 2- The intensity difference between the edge pixel and its surrounding ones is high 3-The similar number of the edge pixel is small. The pixels in the second division have the following features 1-They are majority, since they represent most of the image pixels. 2-The intensity differences between them are very small due to the homogeneity among them. 3-The numbers of their similar pixels are high. Let us look at the following example for pixels in 7x7 window from Lena image in Fig.1-a, we find that there are N=14 similar pixels make with the middle edge pixel x=218 intensity differences ≤ 20. N is a small number because x is located between two regions of high intensity variations, but in fig.1-(b), there are N=48 similar pixels make with x=194 intensity differences ≤ 20. N in this case is a large number because x is located among smoothing area pixels. That also is very obvious in Fig.2. It shows that 67.3% of all the Lena image pixels have similar pixels number between (45-48] and the other 32.7% remaining pixels have similar pixels number less than or equal to 45 pixels in a 7x7 window.

240	227	207	182	169	159	159
247	237	217	192	175	158	156
249	239	229	205	185	164	158
251	243	232	218	198	175	165
255	247	240	232	213	189	173
255	252	244	235	221	200	182
255	252	249	239	226	211	193

(a)

193	187	187	195	195	194	200
192	187	186	194	195	194	199
191	189	186	194	194	194	199
191	190	186	194	195	194	199
190	190	187	196	196	195	201
189	191	188	197	197	196	202
189	191	189	198	198	197	203

(b)

Fig 1. 7x7 windows in different regions in Lena image.(a) shows pixels in abrupt areas (b) pixels in smoothing areas.

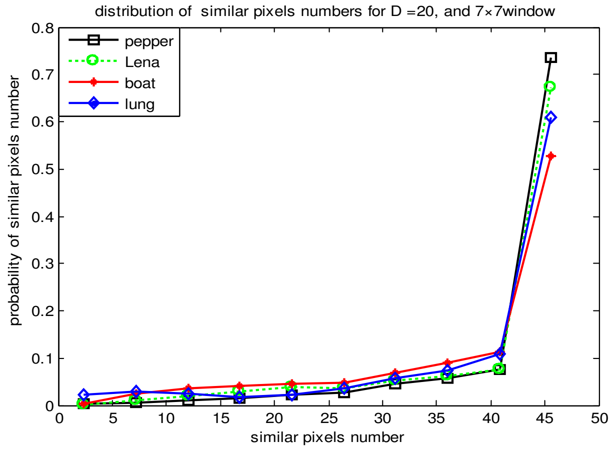

Fig.2 The distribution of similar pixels numbers within a 7×7 window size and within an intensity difference ≤ 20 for different images.

Fig.3- *a, b, c,* and *d* show the locations of the pixels that have similar pixels number $N \leq 40$ within D=20, $N \leq 25$ within D=20, $N \leq 10$ within D=10 and $N \leq 5$ within D=5, respectively for Lena image. It is obvious that all of those pixels are minority pixels and located on the edges of the image. Therefore, we can define a new parameter called similarity parameter $S = \dfrac{D}{N}$ to measure the similarity of a

pixel with its surrounding ones in the filtering window within an intensity range [0, D] and would give an acceptable edge detection results. It is clear from the last example that at $S \approx 1$ satisfactory performance is obtained. This value can be used as a threshold.

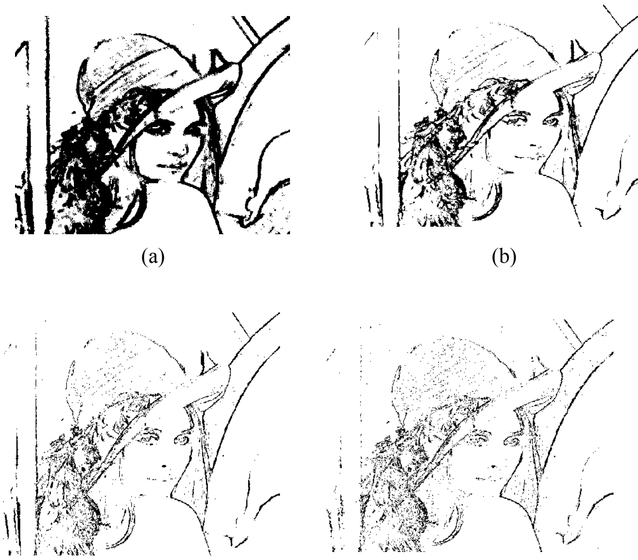

Fig.3 The locations of the minority pixels in Lena image based on their similar pixels number N *within* an intensity range [0, D] (a) $N \leq 40$, D =20 (b) $N \leq 25$, D =20, (c) $N \leq 10$, D =10, (d) $N \leq 5$, D =5.

II. Algorithm Description

Define the filtering window $W(p_{ij})$ of $k \times k$ size centered at the pixel $p_{i,j}$ and the location (i, j) in the image P. In this algorithm each pixel p_{ij} is transformed to a binary value L_{ij} based on a pre-defined value of S^{th} in each phase r as :

$$T(p_{ij,r}) = L^r{}_{ij} \quad \forall p_{ij,r} \in P_r \tag{1}$$

$$L^r_{ij} = \begin{cases} 255 & if \;\; S_r \, {}^{\geq}\!/\!{}_{\leq} \, S_r^{th} \\ 0 & else \end{cases} \tag{2}$$

$$S_r = \frac{D_r}{N_r} \leq \frac{D_r}{k^2 - 1} \qquad r = [1,2] \tag{3}$$

${}^{\geq}\!/\!{}_{\leq}$ means one of the two signs $\{\geq \; or \leq\}$. As S decreases the similarity level increases and hence more edge pixels detected. S^{th} is the threshold of the similarity level that helps judging if the current pixel x is an edge pixel or not. The proposed approach consists of two phases as illustrated below:

A. Edge Identification Phase

In this phase we try to differentiate between the edge pixels and the other pixels in the image as the following. Take the absolute intensity difference between the current central pixel $p_{i,j}$ and each of its surrounding pixels in the window as:

$$d(p_{i-s,j-t}, p_{i,j}) = |p_{i-s,j-t} - p_{i,j}| \tag{4}$$

where $\{s, t = 0 \pm 1, \ldots, \pm \frac{k-1}{2}, (s,t) \neq (0,0)\}$

Then, count the number $C(p_{ij})$ of the pixels in the filtering window that make intensity differences $\leq D$ with $p_{i,j}$

$$C(p_{ij}) = \; number \; \{d(p_{i-s,j-t}, p_{i,j}) \leq D\} \tag{5}$$

any pixel $p_{i\text{-}s,j\text{-}t}$ satisfies (5) is called a similar pixel to p_{ij}. The pixel p_{ij} is considered as an edge point and replaced by a zero in the corresponding location in another binary image U if $C(p_{ij}) \leq S_1^{th}$, otherwise it is replaced by 255 as:

$$U(u_{i,j}) = \begin{cases} 0, & C(p_{i,j}) \leq S_1^{th} \\ 255 & else \end{cases} \tag{6}$$

Note that, the threshold $S_1^{th} \cong 1$ delivers satisfactory results for many images.

B. Complementary phase:

If the detected image is noise –free image, then we have to stop by phase 1. But if the detected image is a noisy image then image U will contain all the detected edge pixels, the noisy pixels that satisfy the first threshold, and the background white pixels. Note that the noisy pixels in image U have a small number of similar pixels compared to that of the edge pixels, because the edge pixels are condensed along the edge lines; however the line shape, but the noisy pixels are scattered through the background pixels (white pixels) which make a large intensity differences with the noisy pixels. Also, we can increase the similar pixels number for the edge pixels by increasing the first threshold. Therefore, for extracting the edge points we need to repeat step 1 and 2 on the image U. Any pixel $u_{ij} = 0$ in U is considered as an edge point in a binary image V if $C(u_{ij}) \geq$ the second similarity parameter S_2^{th} , otherwise it is replaced by 255 in V as the following:

$$V(v_{i,j}) = \begin{cases} 0, & C(u_{ij}) \geq S_2^{th} \\ 255 & else \end{cases} \tag{7}$$

S_2^{th} should be decreased as the noise rate increases and vice versa. In all the simulation experiments we maintain the value of D constant and change only the value of N .

III. Simulation Results

To show the performance of the proposed approach we apply it on different 8-bit grey-level images and compare the results with other well known methods in the literature as Sobel and Canny detectors. The results are compared subjectively in terms of the edge quality, and computationally in terms of the relative processing time in seconds for the different methods (measured by MATLAB COMAND “etime”). CPU of 1.73GHZ and RAM of 1MB are used in all the simulation experiments.7×7 window size is used with the proposed detector. Similarity parameter $S_1^{th} \cong 0.8$ is used with the noise-free image and in the 1st phase of the noisy images. The threshold Th=120 is used with Sobel method and the ones that are used with Canny approach are Th_{max}=120, Th_{min}=70.

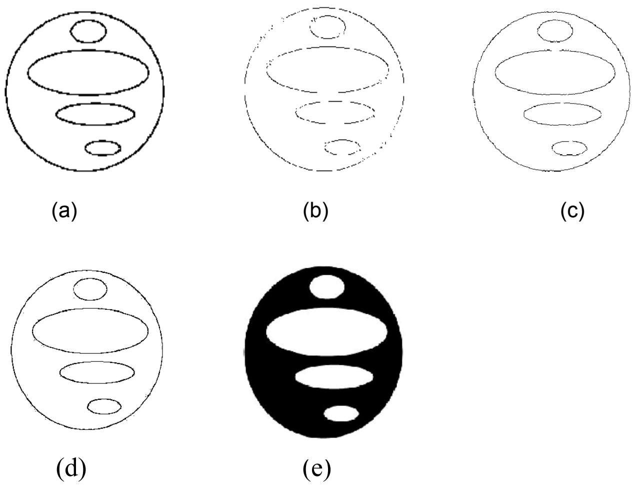

Fig.4 Edge detection results for different filters on a synthetic image: (a) Sobel (t= **2.5 sec**), Th =120 (b) Canny (t= **5.1 sec**), Th_{max}=120, Th_{min}=70(c) EMO (t= **25.1 sec**), (d) New (t= **3.9** sec), (D_l=20, N_l=25), (e) original image.

Fig. 4 is applied on a noise-free synthetic image of 323×393 size. In this experiment we compare our method with EMO approach [2] that is proposed for uncorrupted images, besides Sobel and Canny methods. It is clear that the proposed method shows a continuous thin edge line while the other methods suffer either from discontinuity in the edge line or show a thicker edge. Besides, the processing time of the proposed detector is comparable to Canny and Sobel filters while it is faster 6.4 times than the EMO filter, see the caption of fig. 4. The slow convergence that shown by the EMO method is due to the enormous using of the subtraction operations, i.e., it needs 4x18 subtraction operations in each window in the four directions.

Fig. 5 and Fig. 6 show the results of different methods for edge detection in corrupted pepper and Lena images with 20% and 25% salt and pepper impulse noise rates, respectively. Pepper and Lena images are both of 512× 512 sizes. For appropriate comparison the corrupted images are firstly restored by using 3 ×3 median filter since it high efficiency in impulse noise removal, and then we apply the Sobel and Canny approaches on the restored images. It is noticeable that the proposed method delivers better performance than Canny, and Sobel methods since it effectively removes the impulse noise and maintains the main image features, while the other methods are still contain residual noise and miss some of the image details. The edge lines that are obtained by the new method look somewhat thick; the reason is that some of the neighboring noisy pixels are able to satisfy the threshold criterion. However, they illustrate the main feature of the image to be used for any higher level image processing task. Moreover, the proposed detector is faster than the Sobel and Canny approaches, respectively. Note that the computational times

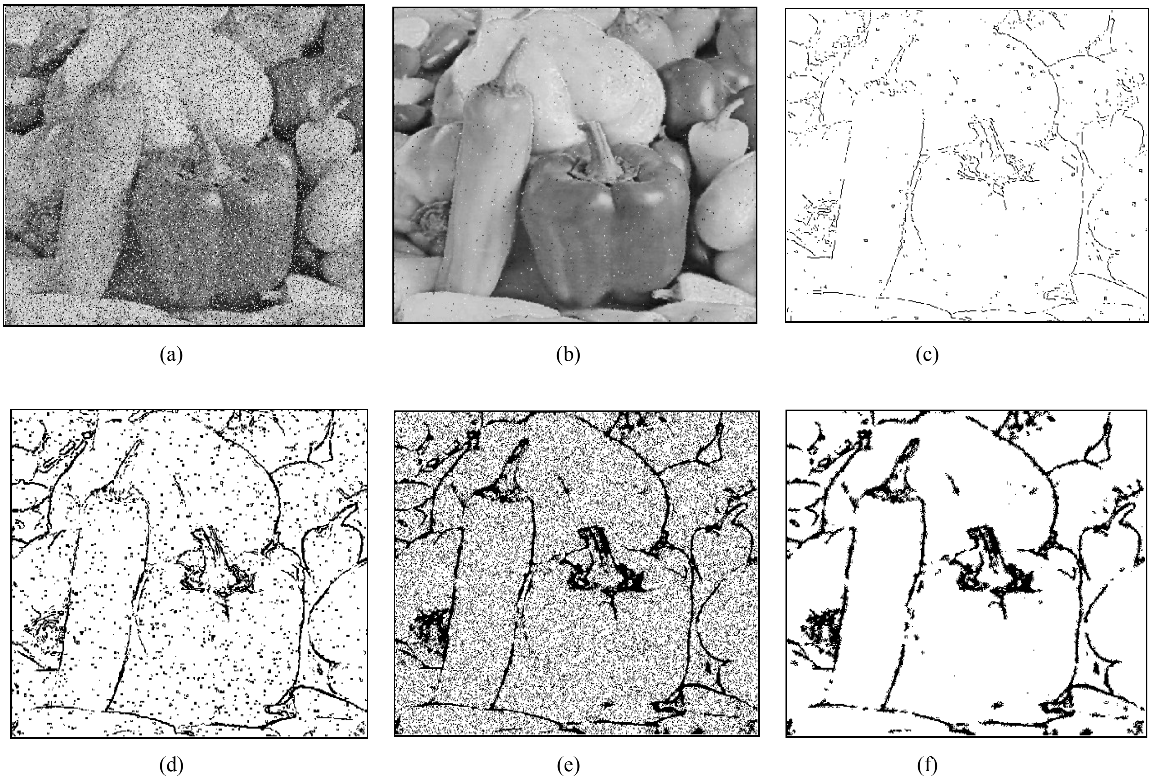

Fig. 5 Edge detection for 20% impulse noise corrupted pepper image: (a) corrupted image, (b) restored version by median filter,(c) Canny after restored image, time= **21.3** sec, Th_{max} =120, Th_{min} =70 (d) Sobel after restored image, time= **15.1** sec , $Th=120$ (e) proposed filter after corrupted image -1 st phase (N_1^{th} =25, D_1 =20) time=7 sec (f) proposed filter-2 nd phase (N_2^{th} = 20, D_2 =20,), time= **12.3** sec

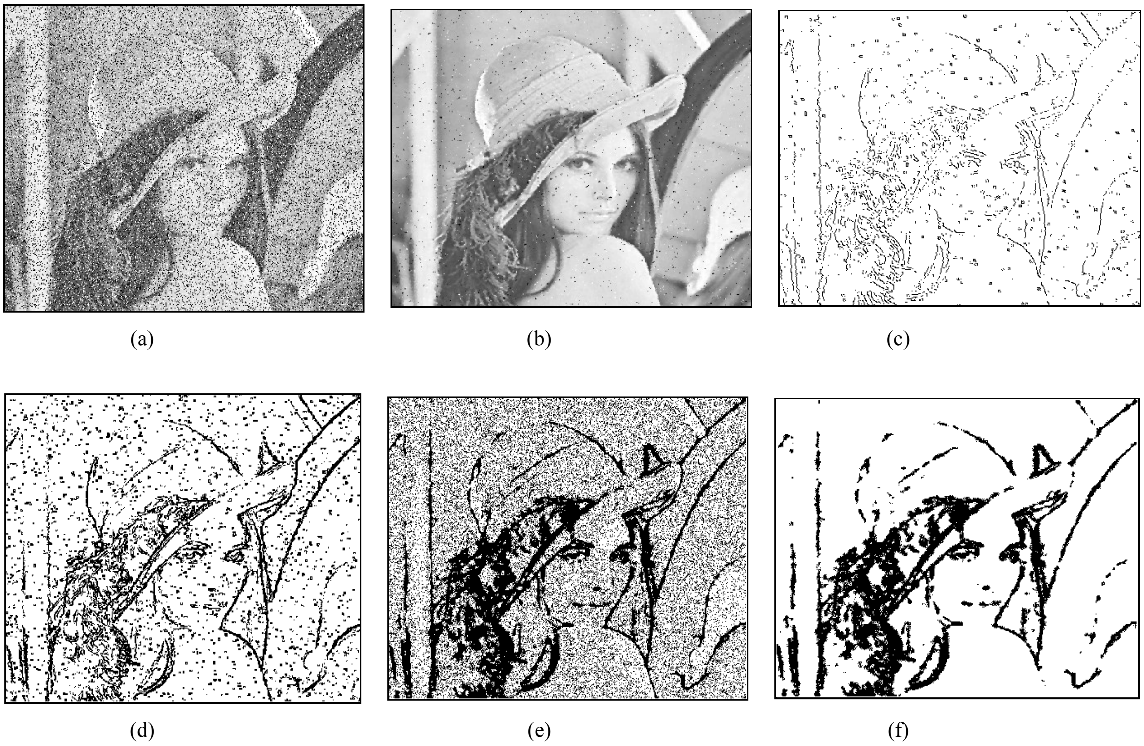

Fig.6 Edge detection for 25% impulse noise corrupted Lena image: (a) corrupted image, (b) restored version by median filter,(c) Canny after restored image, time= **21.3** sec, Th_{max} =120, Th_{min} =70 (d) Sobel after restored image , time= **15.1** sec, *Th*= 120 (e) proposed filter -1 st phase after corrupted image(N_1^{th} =25, D_1 =20) time=7 sec (f) proposed filter-2 nd phase (N_2^{th} = 25, D_2 =20,), time= **12.3** sec,

that are obtained after one or more iterations are the same for all the methods. In the 2nd phase the edge quality increases as the similarity parameter decreases and vice versa. The reason is that, as the noise rate increases the number of similar pixels for the residual noise in the 1st phase images increases. Therefore, we have to decreases S_2^{th} by increasing N_2 as shown in fig. 5 and 6. Since the edge pixels have larger number of similar pixels than any other noisy pixels, we expect that most of the original edge pixels will satisfy the second similarity parameter threshold.

IV. Conclusion

A high performance edge detection approach based on the similarity criteria is proposed in this paper. In which, the pixels that have minimum numbers of similar pixels is consider as edge pixels in free-noise images, and the pixels that have maximum numbers of similar pixels are considered as edge pixels in the noisy images. Simulations results indicate that the proposed approach achieves superior performance than other well known methods, particularly in images corrupted by impulse noise. Moreover, it is time efficient method and has a low computational complexity.

References

[1] C. Spinu, Garbay, and J. M. Chassery Edge detection by estimation and minimization of errors," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 695–704, August 1997.

[2] C. C. Kang, and W. J. Wang, "A novel edge detection method based on the maximizing objective function," *Pattern Recognition*, vol. 40 no.2,pp. 609–618, Feb. 2007.

[3] J. Wu, Z. Yin, and Y. Xiong, "The fast multilevel fuzzy edge detection of blurry images," *IEEE Signal Processing Letters,* vol. 4, no. 5, pp. 344–347, May 200.

[4] L. Caponetti, N. Abbattists, and G. Carapella, "A genetic approac to edge detection," in *Int. Conf. Image Processing*, vol. 94, 1994, pp. 318–322.

[5] V. Srinivasan, P. Bhatia, and S. H. Ong, "Edge detection using neural network," *Pattern Recognit.*, vol. 27, no. 12, pp. 1653–1662, 1995.

[6] T. J. Hebert and D. Malagre, "Edge detection using a priori model," in *Int. Conf. Image Processing*, vol. 94, 1994, pp. 303–307.

[7] R. Gonzalez, and R. Wood, "Digital Image Processing," Addison Wesley, 1992, pp 414–428.

[8] J. Canny, "A computational approach to edge detection," *IEEE Tranns. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–697, June 1986.

[9] E. Abdou and W. K. Pratt, "Quantitative design and evaluation enhancement/thresholding edge detectors, " Proc. IEEE , vol. 67, pp. 753–763, 1979.