The application of a new algorithm for noise removal and edges detection in captured image by WMSN

Astrit Hulaj[1], Adrian Shehu[2], Xhevahir Bajrami[3]
Department of Electronics and Telecommunications, Faculty of Information Technology[1,2],
Polytechnic University of Tirana[1,2], Vienna university of technology, IHRT[3]
Square "Mother Teresa", 4, Tirana[1]. Sunny Hill, Victory2, 13, Prishtina[2]. Favoritenstrasse 9/E325A6[3]
ALBANIA[1], KOSOVA[2], AUSTRIA[3]
astrithulaj@hotmail.com, adshehu@tcn.al, xhevahirbajrami@gmail.com

Abstract: - Problems that have recently various countries in the world about the security of state borders, especially the green border, has done today many countries to think about the application of Wireless Multimedia Sensor Networks (WMSN) for the green borderline surveillance. However the application of this technology along the green border is followed by many challenges, one of which is power supply.

For optimization of energy consumption by the multimedia sensor nodes the conversion of captured images by the WMSN in black and white images is a very important factor. By converting the images to black and white, we can have a loss of details of image characteristics. To preserve the characteristics of the image, the edges detection of the image is a key factor. However, interfering noises from various sources in the image may affect the image edges damage. Therefore, it must first be realized removal of noise within the image and then the edge detection of the image.

Today there are some algorithms and filters that are applied for noise removal from the image and the edges detection of the image.

In this paper, we will present a new algorithm which enables that efficiently, initially noise removal when various types of noise are involved simultaneously in the image, and then the detection of pixels that correspond to the edges of the image captured by the WMSN.

Key-Words: - WMSN; borderline; edge detection; sensor; image; filter; algorithm.

1 Introduction

The images captured by Wireless Multimedia Sensor Network (WMSN) located along the green borderline [1] must be processed by the network devices and sent to the appropriate destination. Image processing is any form of information processing where input is an image, while the output is not necessarily an image. Output for example can be a set of features giving information about the input image. In the case of an image, the most important information is on the edges of the image. The edges of the image correspond to the object boundaries, changes in surface orientation and construction. Edges typically correspond to points in the image where the gray value changes significantly from one pixel to another. Therefore, edge detection has an important role in extracting useful information characteristics of the image, in order to identify rapid changes in the framework of the image [2].

Edge Detection is the process of locating the pixels that correspond to the edge of the image. Edge detection affects the image size reduction and filters out information that can be considered as less important, but maintaining important structural properties of the image.

Reducing the size of the image is directly related to the reduction of the transmission power of a sensor node. This reduction affects in optimization of energy consumption by the WMSN. This is directly related to the lifetime of Multimedia Sensor Node.

Since edges in most cases represent the object boundaries, edge detection is widely used in image segmentation, where in this case the images are divided into zones, which correspond to different objects.

2 Problem Formulation

The placement and management of WMSN along the green borderline, as specified in [1] is characterized by many challenges. One of these challenges is the power supply of sensor nodes. Power supply of sensor nodes is vital to the operation of WSNs. In other words, energy is WSN lifetime itself. Therefore, methods and techniques are continually explored that affect the reduction of energy consumption, so that the lifetime of sensor nodes to be longer.

Today, there are a number of papers that have addressed such a problem. Especially, it is very important in the case of application of WMSN along the green borderline, in those areas where human presence is difficult.

The task of a sensor node is the capture, processing and routing of images thought the network to the nearest Border Police Station. However, it is required to be realized with minimum energy consumption and load transmission link. One of the methods to realize an optimization of energy consumption is conversion images captured by WMSN in black and white images. Conversion to black and white image affects the image size reduces. Reducing the image size impact the reduction of the transmission power of sensor node and load reduction of the transmission link. By reducing transmission power, energy consumption is reduced. This directly affects the lifetime of the sensor node.

A black and white image with 128x128 pixels has a size of 2 kB. The same image with greyscale has a size of 16 kB [2].

The formula for the file size in kBytes is:

$$\textbf{Size of an image} = \frac{(\text{rows}) * (\text{columns}) * \text{bit depth}}{8(\text{convert to bytes}) * 1024} \qquad (1)$$

However, it should be noted that through the black and white images it can’t be seen all specific details of the objects. Today, with purpose of the detection these details of images some algorithms are presented. These algorithms (filters) tend that through the detection of edges of the capture as much details in the framework of an image. With details, we will understand detection pixels that represent edges of the image. However, within an image can have the presence of noise which can affect that the detection of the edges of the image be the inefficient. Therefore, in the case where noise is present, it must first be removed from the image and then image edges detection is realized. With the detection of image edges, less important information can be removed. The removal of this information directly impacts the image size reduction. In this paper a new algorithm which enables the removal of various noises and edges detection within one image at very satisfactory way is presented.

3 Existing filters for edge detection

In order to realize the edge detection of an image, the main purpose is detection of important events that occur in the image.

With edge detection, we understand each mathematical method which is used for identification of points in a digital image in which the brightness of the image rapidly changes or more formally, where are discontinuities.

To realize the detection of edges in an image, the main purpose is to capture important events that occur within the image. Usually, the edges are presented at the boundary between the two different regions in an image. Points at which the intensity varies rapidly are organized by lines that represent the image edges. Today there are some types of filters that enable the edges detection in a digital image.

These filters are:

- Sobel
- Prewitt
- Roberts
- Laplacian of Gussian
- Canny, etj

Sobel, Prewitt, and Robert filter are filters that using First Order Derivative. While, Canny and Laplacian of Gaussian are filters that using Second Order Derivative. Working principle of these filters below is presented.

3.1 Sobel Filter

Sobel filter is a filter that is used for edges detection. Edges Detection is realized through calculating the intensity gradient of the image at each pixel inside the image. This filter finds the direction of increasing the intensity from brighter to darker part, and the ratio of change in this direction [2].

The Sobel Filter consists of a pair of 3×3 convolution kernels. One of the possible changes in the horizontal direction (is sensitive to changes in the direction of the x-axis) and the other for possible changes in the vertical direction (is sensitive to changes in the direction of the y-axis). One kernel matrix is simply the other rotated by 90° [3]. These kernel matrices are shown in Figure 1, 1a and 1b respectively.

-1	0	+1
-2	0	+2
-1	0	+1

Gx
a)

+1	+2	+1
0	0	0
-1	-2	-1

Gy
b)

Fig. 1. Kernel Matrices for Sobel Filter a) vertical direction, b) horizontal direction..

These Kernel Matrices can be applied independently in the image. As a result we will have separate measurements for the gradient Gx (vertical direction) and Gy (horizontal direction). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient [11].

From Figure 1a we can see that the kernel matrices, containing coefficients of positive and negative values. As result, the gradient will contain positive and negative values. It makes that gradient with negative value appear dark, while gradient with positive value appear bright. In order to eliminate this, we need to apply absolute value and as a result will have that both negative and positive gradient appear as bright.

Measurement of an edge of the image is its amplitude and angle. It is realized through find gradient magnitude and its direction. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2)$$

Although, an approximate magnitude is computed using the equation:

$$|G| = |G_x| + |G_y| \quad (3)$$

The gradient direction is computed using the following equation:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (4)$$

Although Sobel Filter operation is fast, this filter is very sensitive to noise. The presence of noise in the image will affect that the magnitude of the gradients is degraded. As a result, Sobel Filter can’t offer the best edges detection in the image.

3.2 Prewitt Filter

This filter is used also for edges detection. The working principle of the Prewitt filter is similar to Sobel filter. Mathematically, Prewitt filter also consists of a pair of 3×3 convolution kernels. However, Prewitt Filter, unlike the Sobel Filter, uses other kernel matrices. These kernel matrices are shown in Figure 2, 2a and 2b respectively. Although, it should be noted that the differential gradient edge detection need a rather time-consuming calculation to estimate the orientation from the magnitudes in the x- and y-directions.

-1	0	+1
-1	0	+1
-1	0	+1

Gx
a)

+1	+1	+1
0	0	0
-1	-1	-1

Gy
b)

Fig. 2. Kernel matrices for Prewitt Filter a) vertical direction, b) horizontal direction.

This filter uses first order derivative for calculate the difference of intensity of pixels in an edge region. From figure 2 we can see that centre column of the Gx mask is zero, and centre row of the Gy mask is zero.

As a result, during the application of these masks aren't included in calculation the original values of an image but rather it calculates the difference of right and left pixel values around that edge. This impact on increasing the intensity of the edge and it happens relatively increases the value of the original image.

3.3 Robert Filter

Robert Filter, as Sobel and Prewitt filter uses first order derivative for the edges detection. Also, the application logic is similar to that of the Sobel and Prewitt filter.

Robert filter uses a pair 2x2 convolution masks. These masks are designed to respond maximally to edges running at 45 ° to the pixel grid, one mask for each of the two perpendicular orientations [4]. The masks can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (*Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient **[3] [5] [7]**. These matrices are rotated with each other for 90^0 and are shown in Figure 3, 3a and 3b respectively.

+1	0
0	-1

Gx
a)

0	+1
-1	0

Gy
b)

Fig. 3. Convolution masks of Roberts Filter

The gradient magnitude is given by the equation:

$$|G| = \sqrt{G_x^2 + G_y^2} \qquad (5)$$

Although, an approximate magnitude is computed using the equation:

$$|G| = |G_x| + |G_y| \qquad (6)$$

The angle of orientation of the edge giving rise to the spatial gradient is given by:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) - \frac{3\pi}{4} \qquad (7)$$

The Roberts cross operator is fast to compute (due to the minimal size of the kernels), but it is very sensitive to noise [10].

3.4 Laplacian of Gaussian Filter

The Laplacian Filter of an image highlights regions of rapid intensity change and is therefore often used for edge detection [5]. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The Laplacian filter is an operator of the second order derivative.

For Finding edges Laplace filter based method search for zero crossings in a second-order derivative. The Laplacian of Gaussian Filter normally takes a single gray level image as input and produces another gray level image as output.

In particular, we are interested in bringing these derivatives in the areas with constant gray level (flat segments), at the beginning and end discontinuity (with step and rapid discontinuities), and during rapid changes in gray level. These types of discontinuities can be used to model the noise points, lines and edges in an image. Bringing derivatives during the transition in and out characteristics of this image is also of interest [3]. The derivatives of a digital function are defined in terms of differences. There are various ways to define these differences [9].

However, for the first and second derivatives used these definitions:

1. Must be zero in flat areas (areas of constant gray-level values);
2. Must be nonzero at the onset and end of a gray-level step or ramp; and
3. Must be nonzero along ramps of constant slope.

Laplace filter for an image by *f(x, y)* intensity values is given:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} \qquad (8)$$

Because in every way derivatives are linear operations, Laplace filter is a linear filter. The first part in equation (10) is applied for edge detection in direction of the x-axis. While, the second part in equation (10) is applied for edge detection in direction of the y-axis. When we consider the first part of the equation (10) and derive in the direction of the x-axis, we have:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y) \qquad (9)$$

When derive in the direction of the y-axis, we have:

$$\frac{\partial^2 f}{\partial^2 x} = f(x, y+1) + f(x, y-1) - 2f(x, y) \qquad (10)$$

So, after replacing the equations 11 and 12 in equation 10 Discrete Laplacian Filter is given by the equation:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y) \qquad (11)$$

Equation (13) can be implemented using a 3x3 mask which is shown in Figure 4:

0	**+1**	**0**
+1	**-4**	**+1**
0	**+1**	**0**

Fig. 4. A 3x3 mask the Laplacian Filter.

With the application of this filter on an image, we will have great value at those points in the image, where the local gradient changes more quickly [10]. One of the shortcomings of the application of this mask is relative insensibility in regarding to the characteristics of false in approximately diagonal directions in regarding to the axis of the image.

The Laplace filter is very sensitive to noise. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise before applied Lapacian Filter [3]. This pre-processing step reduces the high frequency noise components prior to the differentiation step [3], [10]. In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result [3], [8], [4]. Laplacian of Gaussian presented with the following equation:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (12)$$

3.4 Canny Filter

Canny Edge Detector technique is very important for detecting edges in an image. This operator isolates noise from an image before finding, edges of an image without affecting the features of the image, and then applying the tendency to find the edges and the critical value for threshold [6].

Canny proposed three criteria of the evaluation the pros and cons of performance of edge detection [11]:

- Standard of ratio of signal to noise, that is real edge detection probability is higher and non-edge points sentenced to be lower the probability of edge points, so that the output of ratio of signal to noise is maximum;
- Standard of positioning accuracy, that is there is great possibility that the detected edge points is actually in centre of the edge;
- The unilateral corresponding standard that is the probability of multiple responses in single edge is low, and false edge The response should be the maximum inhibition.

For edge detection Canny is based on three criteria [8].

- The first criterion and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges.
- The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum.
- The third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the canny edge detector, the first smooth the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (no maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed [7]. Hysteresis uses two different thresholds $T1$ and $T2$ and if the magnitude is below the T1, it is set to zero (made a non-edge). If the magnitude is above the T2, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2 [7].

4. The Proposed Algorithms

So far, many methods have been applied for image edges detection. However, we will present two algorithms. The first algorithm is applied for detection of the edges of the image when the image hasn't been corrupted with noise. The second algorithm is as the combination of algorithm presented in [12], and the first algorithm. The second algorithm allows us that efficiently initially of the remove all interfered noises from different sources and then the pixels detection that correspond to the edges of the image.

These algorithms can be summarized as follows.

Let's take an image *A(i,j)* as input image with dimensions mxn. Where ***i*** take values *i∈(1....m)* and ***j*** take values *j∈(1...n).*

Algorithm 1: The algorithm goes through these steps:

Step 1. *Read input colors image;*
Step 2. *Give value for **Thresh;***
Step 3. *Convert read image to grayscale image, A(i,j);*
Step 4. *Return the size of matrix A(i,j) in separate variables m and n;*
Step 5. *Create the output using zeros with the data type for image A(i,j);*
Step 6. *Implementation using loops for rows and columns;*
Step 7. *Applications 3x3 filter within loops;*
Step 8. *Gradient magnitude calculation;*
Step 9. *Apply condition for finding edges in the image;*
Step 10. *Give value for **Factor;***
Step 11. *Find edges including Thresh and Factor values;*
Step 12. *Display image after edges detection;*

Algorithm 2: The algorithm goes through these steps:

Step 1. *Read input colors image;*
Step 2. *Give value for **Thresh;***
Step 3. *Convert read image to grayscale image A(i,j);*
Step 4. *Add salt & pepper noise with noise density of 0.2 in image A (i, j);*
Step 5. *In image with salt & pepper noise, add Poisson noise;*
Step 6. *In image with salt & pepper and Poisson noise, add Speckle noise;*
Step 7. *In image with salt & pepper, Poisson and Speckle noise, add Gaussian noise with standard deviation 0.02;*
Step 8. *Define the dimensions of the window (3x3 kernel mask);*
Step 9. *Create the output using zeros with the data type for image A(i,j);*
Step 10. *Return the size of matrix A(i,j) in separate variables m and n;*
Step 11: *Implementation using loop for rows and columns;*
Step 12. *Implementation using the second loop for rows and columns for design filter;*
Step 13. *Application 3x3 filter, within the second loop;*
Step 14. *Sorting the pixels of the filter applied from a pixel of lesser value or equal to the pixel of greater value;*
Step 15. *After step 13, we find Median pixel (fifth pixel) of the filter;*
Step 17. *Display image filtered;*
Step 18. *Implementation using second loops for rows and columns;*
Step 19. *Applications 3x3 filter within loops;*
Step 20. *Gradient magnitude calculation;*
Step 21. *Apply condition for finding edges in the image;*
Step 22. *Give value for **Factor;***
Step 23. *Find edges including Thresh and Factor values;*
Step 24. *Display image after edges detection;*

---

5. Results and discussion

In this section we will present compare and discuss the results obtained with the application of existing filters for edges detection within an image, and these results will be compared with results obtained with the application of the new algorithms.

Results were derived by using MATLAB 2014 and an image with 153x329 pixels.

In Figure 5 and 11 are presented the results obtained with the application of Roberts Filter. In Figure 11, compared with Figure 5 are presented the results in the case where from the image initially noises that have image corrupted are removed and then Roberts Filter is applied for edges detection of the image. In Figure 6 and 12 are presented the results obtained with the application of Prewitt Filter. In Figure 12, compared with Figure 6 are presented the results in the case where from the image initially noises that have image corrupted are removed and then Prewitt Filter is applied for edges detection of the image. In Figure 7 and 13 are presented the results obtained with the application of Sobel Filter. In Figure 13, compared with Figure 7 are presented the results in the case where from the image initially noises that have image corrupted are removed and then Sobel Filter is applied for edges detection of the image. In Figure 8 and 14 are presented the results obtained with the application of **LoG** Filter. In Figure 14, compared with Figure 8 are presented the results in the case where from the image initially noises that have image corrupted are removed and then **LoG** Filter is applied for edges detection of the image. In Figure 9 and 15 are presented the results obtained with the application of **Canny** Filter. In Figure 15, compared with Figure 9 are presented the results in the case where from the image initially noises that have image corrupted are removed and then **Canny** Filter is applied for edges detection of the image. While in Figure 10 are presented the results obtained with the application the algorithm 1, while in Figure 16 are presented the results obtained with the application the algorithm 2.

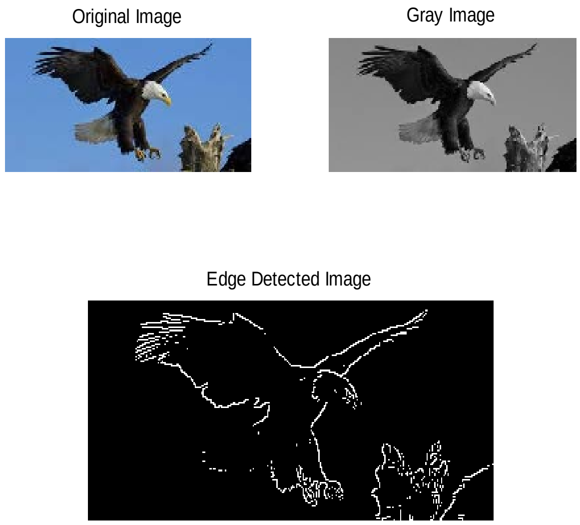

Fig. 5. Roberts Filter.

From the results shown in Figure 5, we can see that Roberts Filter does not provide the desired results. Comparing the results shown in Figure 5 with the results presented in Figure 10, we can see that in the case of Roberts Filter, we have lost many pixels along the edges.

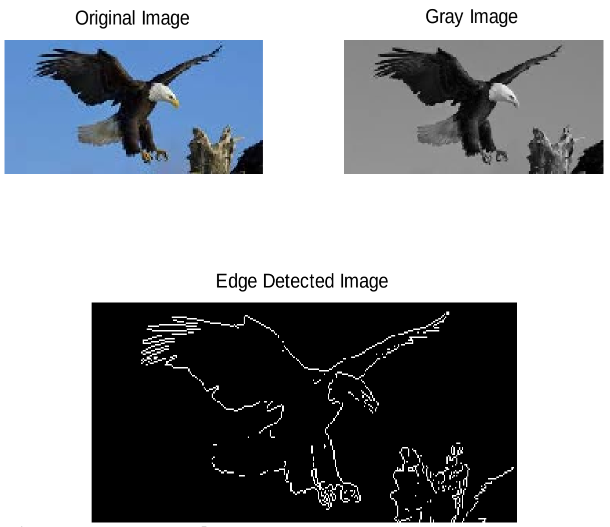

Fig. 6. Prewitt Filter.

From the results shown in Figure 6, we can see that Prwitt Filter, same as the Roberts Filter doesn't provide the desired results. Comparing the results shown in Figure 5 with the presented results in Figure 10, we can see that also in the case of Roberts Filter, we have lost many pixels along the edges.

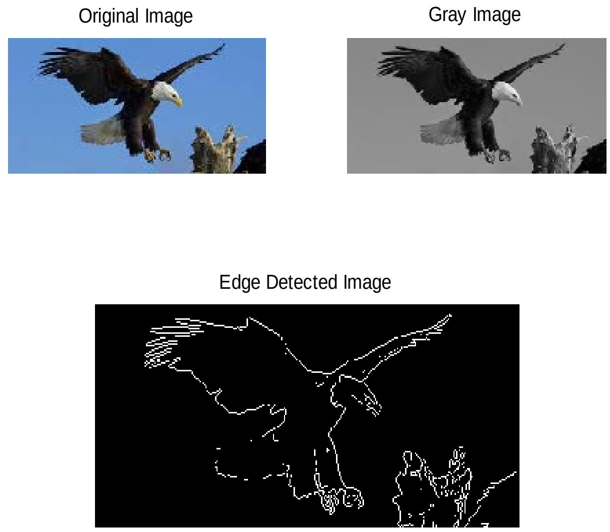

Fig. 7. Sobel Filter.

From the results shown in Figure 7, we can see that Sobel Filter, same as the Roberts and Prwitt Filter doesn't provide good results for edges detection. If the results obtained with the application of Sobel Filter compared with the presented results in Figure 10 we can see that in the case of Sobel Filter we have a lot of pixels no detected along the edges.

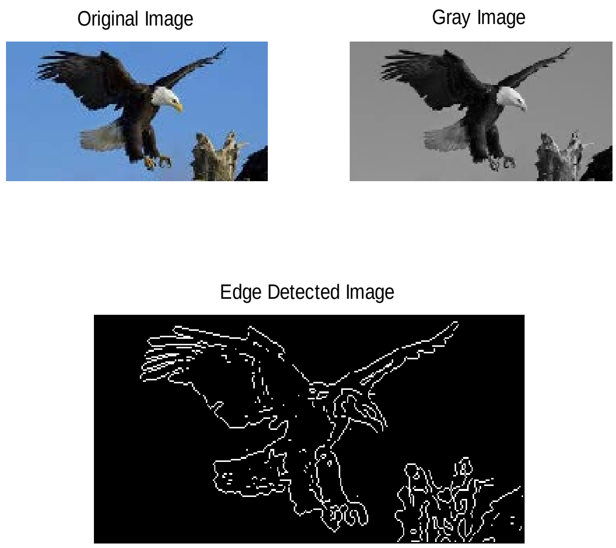

Fig. 8. LoG Filter.

From the results shown in Figure 8, we can see that LoG Filter, provide better results than Robert's, Prwitt and Sobel Filter. However, if the results obtained with the application of LoG Filter compared with the presented results in Figure 10 we can see that in the case of LoG Filter we have least pixels the detected along the edges. Especially, we can see changes in the tail, fingers and wings of the eagle.

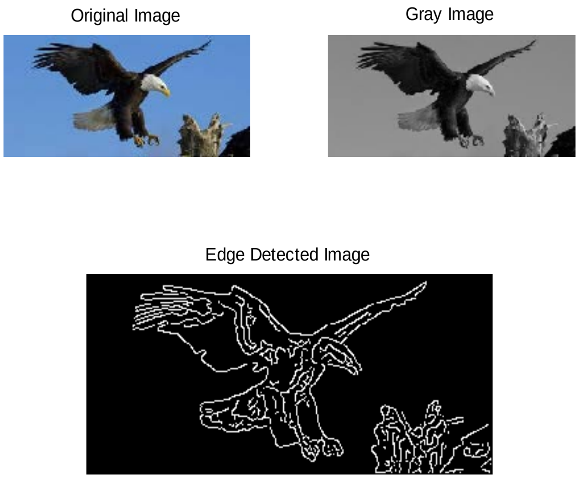

Fig. 9. Canny Filter.

From the results shown in Figure 9, we can see that Canny Filter, provide best results than LoG, Robert, Prwitt and Sobel Filter. However, if the results obtained with the application of Canny Filter compared with the presented results in Figure 10 we can see that in the case of application of Canny Filter isn't reached to edges detection in the tail of the eagle.

Another difference we can see at the bottom of the left wing of the eagle (Right in the figure 9). From Figure 9 and 10 we can see that in the case of application of Canny Filter detected edges are less highlighted than in the case of the results obtained by our proposed algorithm. The same changes we can see also to the toes of the Eagle. However, despite this, the application of Canny Filter, as shown in fig. 9 offers a better correlation of pixels, especially those pixels that are detected within the Eagle [2].

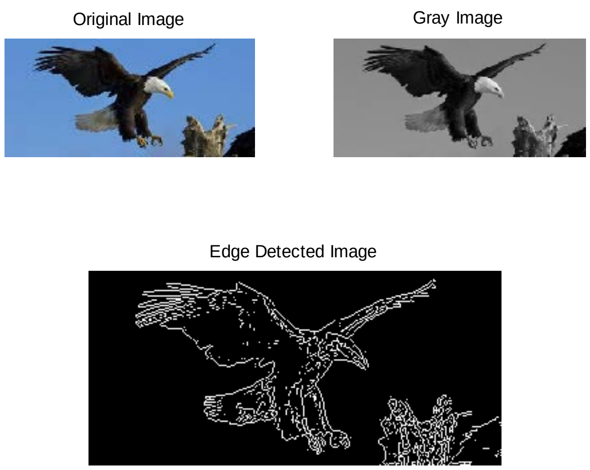

Fig. 10. New Algorithm.

From the results shown in Fig. 2, it can be seen that with the application of our proposed algorithm, we can obtain very good results in the detection of edges within the image captured by the sensor nodes or any other electronic device. If the results presented in Fig. 10 are compared with the results presented in Fig. 5, 6, 7, 8, 9 it can be seen that the proposed algorithm enables a very good detection of pixels that represent the edges of the image.

In following we will present the results obtained with the application of filters for edges detection in the case where noises that have image corrupted such as salt and pepper, Poisson, Speckle and Gaussian noise by applying the algorithm presented in [12], are removed.

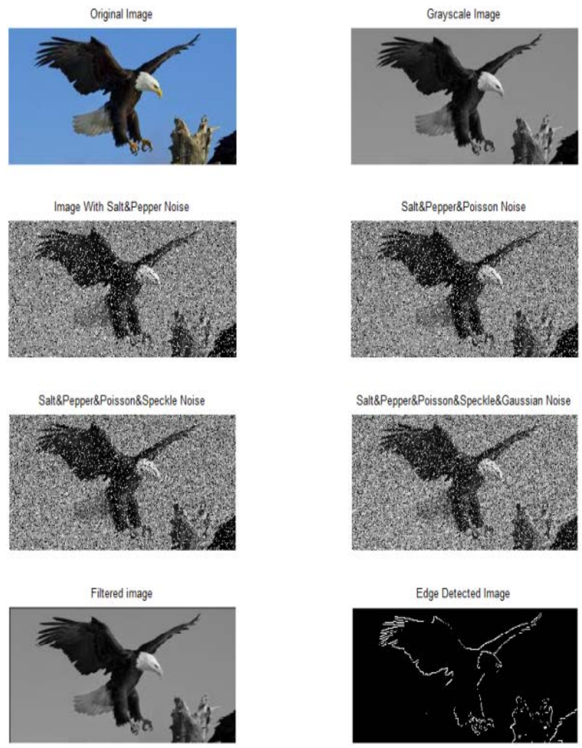

Fig. 11. Roberts Filter

From the results shown in the Fig. 11, we can see that with the application of Roberts Filter, in the image in which the various noises are removed, as in the algorithm 2, results are obtained also more unsatisfactory than those presented in Figure 5.

If the results presented in Fig. 11 are compared with the results presented in Fig. 10 and 16, we can see that in the case the application of Roberts Filter, we have lost many pixels along the edges, in the tail of the eagle, especially. In this part, almost none pixels that represents edges isn't detected [2].

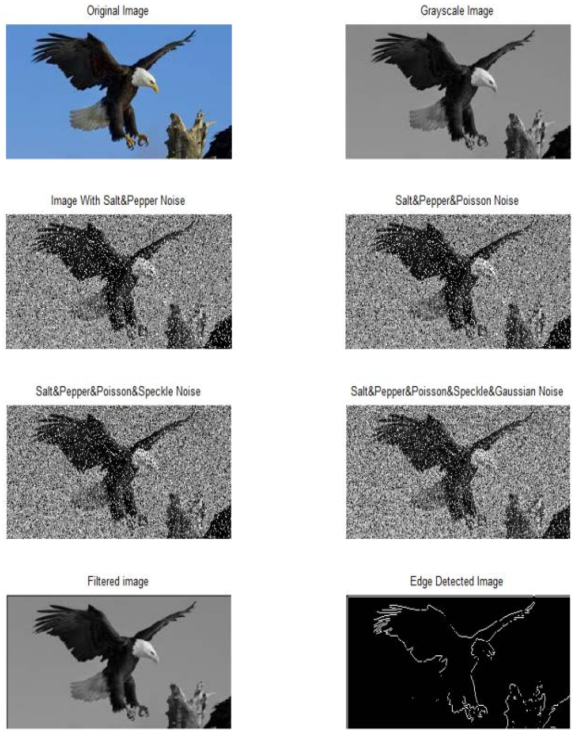

Fig. 12. Prewtti filter

From the results shown in the Fig. 12, we can see that with the application of Prweitti Filter, in the image in which the various noises are removed, as in the algorithm 2, results are obtained similar to those presented in Figure 11.

Fig. 13. Sobel filter.

From the results shown in the Fig. 13, we can see that with the application of Sobel Filter, in the image in which the various noises are removed, as in the algorithm 2, results are obtained similar to those presented in Figure 11 and 12. So the results are much poorer than those presented in Figure 7.

Fig. 14. Log filtër

Fig. 15. Canny Filter

From the results shown in the Fig. 13, we can see that with the application of Log Filter, in the image in which the various noises are removed, as in the algorithm 2, results are obtained slightly poorer than those presented in Figure 8. However, these results are much better than those presented in Figures 11, 12, and 13. If the results presented in Fig. 14 are compared with the results presented in Fig. 10 and 16, we can see that with the application of Log Filter, we have much less pixels detected along the edges of the image.

From the results shown in the Fig. 15, we can see that with the application of Canny Filter, in the corrupted image by the various noises, such as in the algorithm 2 are obtained results almost same as those presented in Figure 9. However, if the results obtained with the application of Canny Filter compared with the presented results in Figure 10 and 16 we can see that in the case of application of Canny Filter isn't reached to edges detection in the tail of the eagle.

Fig. 16. New Algorithm (algorithm 2).

From the results shown in the Fig. 16, we can see that with the application of Algorithm 1 in the image corrupted by the various noises, such as in the algorithm 2 are obtained enough satisfactory results in the comparison with existing filters for detection of the edges. Therefore results are almost the same quality as those presented in Figure 10.

6. Conclusion

As discussed, in the case of captured image by sensor node, for saving power transmission and load link it is very important that the image be converted to black and white and then be applied algorithms that enable as efficiently be performed detection the edges of the image. Therefore in this paper we introduce two algorithms which are very effective in the detection of the pixels that correspond to the edges of the image.

From comparing the results obtained with the application of traditional filters with those obtained with the application of the proposed algorithms, we can see that the proposed algorithms offer the very good results and concrete. Therefore, based on the presented and analyzed results we can conclude that these algorithms are quite suitable to be applied for edges detection of the captured image by wireless multimedia sensor node.

In the future in order to test the algorithms performance in the wireless multimedia sensor and other electronic devices, the real tests will are applied.

References:

[1] Astrit Hulaj, Adrian Shehu, Xhevahir Bajrami, Application of wireless multimedia sensor networks, 27th DAAAM International Symposium on Intelligent Manufacturing and Automation, Vol. 27, 2016, In Press.

[2] Adrian Shehu, Astrit Hulaj, Xhevahir Bajrami, An algorithm for edge detection of the imagefor application in WSN, *Applied Physics, System Science and Computers (APSAC 2016)*, Vo. 1, No. 1, 2016. In Press.

[3] Sharma, Pooja, Gurpreet Singh, and Amandeep Kaur.: Different Techniques of Edge Detection In Digital Image Processing. *International Journal of Engineering research and Applications* Vol. 3, No.3, 2013, pp. 458-461.

[4] Igbinosa, Ireyuwa E, Comparison of Edge Detection Technique in Image Processing Techniques, *International Journal of Information Technology and Electrical Engineering, Vol.* 2, No.1, 2013, pp. 25-29.

[5] Maini, Raman, and Himanshu Aggarwal, Study and comparison of various image edge detection techniques, *International journal of image processing (IJIP)*, Vol. 3, No.1, 2009, pp. 1-11.

[6] Huertas, Andres, and Gerard Medioni, Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks, *Pattern Analysis and Machine*

Intelligence, IEEE Transactions on 5, 1986, pp. 651-664.

[7] Juneja, Mamta, and Parvinder Singh Sandhu. Performance evaluation of edge detection techniques for images in spatial domain, *International journal of computer theory and Engineering*, Vol. 1, No. 5, 2009, pp. 1793-8201

[8] Narendra, V. G., and K. S. Hareesha. "Study and Comparison of various Image edge Detection tehniques used in Quality inspection and Evaluation of Agricultural and Food products by Computer vision." *International Journal of Agricultural & Biological Engineering,* Vol. 4, No. 2, 2011, pp. 83-90.

[9] Gonzalez, Rafael C., and Richard E. Woods, Digital image processing, 2002, pp. 116-141.

[10] Solomon, Chris, and Toby Breckon, *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*, John Wiley & Sons, 2011.

[11] Bin, Li, and Mehdi Samiei Yeganeh, Comparison for image edge detection algorithms, *IOSR Journal of Computer Engineering,* Vol. 2, No. 6, 2012, pp. 01-04.

[12] Astrit Hulaj, Adrian Shehu, Xhevahir Bajrami, An efficient algorithm for different noises removal by WMSN, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 24, 2016, In Press.