CARD International Journal of Science and Advanced Innovative Research (IJSAIR)
ISSN: 2536-7315 (Print) 2536-7323 (Online)
Volume 1, Number 3, December 2016
http://www.casirmediapublishing.com

A Particle Swarm Optimization Based Edge Detection Algorithm for Noisy Coloured Images

Sadiq B.O[1], Zakariyya O.S[2], Abdulrahman O.A[3], Usman M.A[4] and Alao R.A[5]

[1,3]Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria[1]
[2,4,5]Department of Electrical and Electronics Engineering, University of Ilorin, Kwara State
Email: bosadiq@abu.edu.ng

ABSTRACT

This paper presents an improved edge detection algorithm using particle swarm optimization (PSO) based on vector order statistics. The proposed algorithm was implemented using MATLAB 2013 script. The algorithm addressed the performance of edge detection in noisy coloured images, with a view to minimizing broken, false and thick edges whilst reducing the presence of noise. A collection scheme based on step and ramp edges was applied to the edge detection algorithm, which explores a larger area in the images in order to reduce false and broken edges. The efficiency of this algorithm was tested on two Berkeley benchmark images in noisy environments with a view to comparing results both visually and quantitatively with those obtained using proven edge detection algorithms such as the Sobel, Prewitt, Roberts, Cannyand Laplacian edge detection algorithms. The Pratt Figure of Merit (PFOM) was used as a quantitative comparison between the proposed algorithm and the proven edge detection algorithms. The PFOM on the test images in noisy environment for the Sobel, Prewitt, Roberts, Laplacian, Canny and the proposed edge detection algorithms are 0.4191, 0.4191, 0.2807, 0.2811, 0.5606 and 0.8458 respectively. This showed that the developed algorithm will perform better than the existing edge detection algorithm in multimedia systems.

Keywords: Image Edge Detection, Particle Swarm Optimization (PSO), Pratt Figure of Merit (PFOM), Noisy coloured Images and Vector Order Statistics

INTRODUCTION

Detection of edges in images plays a vital role in the areas of surveillance systems, biometrics, network security, vehicle detection and tracking, remote sensing amongst others. However the images produced by digital cameras are often corrupted by noise due to dust particles, environmental factors etc. Noise are unwanted, high frequency component, random variation of image intensity which are inherent in digital images. Noise in images occur during either the capturing stage or the transmission stage due to physics-like photon nature of light and thermal energy inside the sensors (Verma & Ali, 2013). The presence of noise in digital images simply means that the pixels in the image shows a different intensity value instead of the original pixels values. In digital images, the number of corrupted pixels will show the quantification of noise present in the image (Neupane *et al.*, 2012). There exists different types of noise in digital images, depending on the type of disturbance. These types of noise are as follows (Verma & Ali, 2013): The Impulsive noise (Salt & Pepper noise), Amplifier noise (Gaussian noise), Multiplicative noise (Speckle noise). The impulsive noise appear in form of black and white dots in an image. This type of noise occur in the image due to sharp and sudden change in image signal. Dust particles during the image capturing stage or corrupted transmission channel are the major causes of this type of noise (Abdul & Funjan, 2013). A Gaussian noise is a statistical noise having a probability density equal to that of the normal distribution. This type of noise occur during acquisition stage due to the environmental condition (Solomon & Breckon, 2011). The multiplicative noise otherwise known as speckle noise are unwanted signal that worsen the resolution of the active radar and synthetic aperture radar (SAR) images. this type of noise originates due to coherent processing of back scatter signals from different distributed points (Verma & Ali, 2013).

An edge can be defined as a single pixel with local discontinuity in intensity(Sadiq, 2015). Edges in images are also high frequency components which make it difficult to identify in noisy environment. Edge detection is a process of identifying these local discontinuities in images using an algorithm (Sadiq *et al.*, 2015a). Particle swarm optimization algorithm was first introduced in 1995 by Eberhart and Kennedy (Venkata & Babu, 2012). The technique was a population based heuristic optimization problem solving algorithm, which was based on the idea of the social behaviour of bird flocking, fish schooling and swarm theory (Roomi & Rajee, 2011). Particle swarm optimization has five basic parameters which are (Kaur & Singh, 2012) Particle, velocity, fitness, P_{BEST}, and $G_{BEST.}$ Coloured images are three dimensional (3-D) which consists of the Red, Green and Blue channel of 8-bit each. During processing of coloured images, the RGB components in the image is viewed as a vector field that maps a point in the image plane to a three-dimensional (3-D) vector in the colour space. The processing of image is known as the vector order statistics(Sadiq *et al.,* 2015b).

RELATEDWORKS

A number of researchers such as, Chen & Chen, (2010) and Xin *et al.,* (2012) presented edge detection algorithms in noisy environments. The work of Dutta & Chaudhuri, (2009)presented a colour edge detection algorithm in RGB colour space. The algorithm in the RGB colour space used a median filter to suppress the noise in the image, then a maximum directional difference of the sum of grey values when each component of the image taken separately were calculated for each pixel. However, in this procedure there are lots of edges in the colour image that will not be detected due to the transformation technique used. Hence missing edges exists in the generated output edge map. The work of Chen & Chen, (2010)presented an algorithm for edge detection in RGB colour space. This algorithm used a Kuwahara filter to smoothen the original image before applying edge detection. An adaptive threshold selection method was applied to predict the optimal threshold value. An edge thinning algorithm was used to extract the edges considering each channel independently in RGB colour space. But with the application of the Kuwahara filter in smoothening process, edges are often displaced or removed due to the presence of noise in the image. Xin *et al.,* (2012) introduced an algorithm with a view to improving the canny edge detection algorithm to operate on colour images. The algorithm introduced the concept of quaternion weighted average filter (QWAF) and vector analysis to deal with the weakness of the traditional canny edge detection. The algorithm used QWAF with a sliding window of 9x9 to remove the Gaussian noise present in the image, and non-maximum suppression (NMS) based on interpolation for edge thinning. However, the performance of the algorithm highly depended on the size of the sliding window. This implies more blurring as well as detecting thicker edges. The outline of broken and false edges appears less using this algorithm but the computation time is increased due to the sliding window.

Rashmi *et al.*, (2013) studied and presented a comparative analysis between various existing edge detection algorithms, which were Prewitt, Sobel, Canny and Robert edge detection algorithms. Prewitt, Sobel and Robert edge detection algorithms were classified as traditional edge detection algorithms and are very sensitive to noise but faster and easier to implement. The Canny edge detection algorithm was reported to be better applied to images as compared to the traditional edge detection algorithms due to the fact that it is less sensitive to noise than the traditional edge detection algorithms because of the Gaussian filter used. However, this algorithm still produced false and broken edges due to the presence of noise in the image and highly depended on the value of the adjustable parameter σ which is the standard deviation of the Gaussian filter.

Gupta &Mazumdar, (2013) presented an extension of the Sobel edge detection algorithm for images. The standard Sobel edge detection algorithm used a 3x3 convolution kernel on an image. This was extended to a 5x5 convolution kernel in order to deal with noise present in the image. Using a Sobel edge detection algorithm is relatively inexpensive in terms of computation time. However, the gradient approximation it produces is inaccurate, thus false, broken and thick edges exists in the output image.

In view of the shortcomings associated with the related works, there is need to introduce an effective noise filtering algorithm in order to minimize the effect of false and broken edges that exists in the generated output edge map.

NOISE REMOVAL USING PARTICLE SWARM OPTIMIZATION

The role of particle swarm optimization is to solve image enhancement problem by tuning the parameters with a view to obtaining the best combination according to an objective criterion that describes the contrast of the image. The swarm is initialized randomly, with a group of particles and it then searches for optima by updating through iterations. Two best values are used to update each particle in every iteration. The first one is the best solution of each particle achieved so far known as P_{BEST} and the other is the best solution tracked by any particle among all generations of the swarm known as G_{BEST}(Venkata & Babu, 2012). With respect to the two best values obtained, a particle updates its velocity and position with the help of the following equations (Kaur & Singh, 2012).

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (1)$$

$$V_i^{i+1} = W^t V_i^t + C_1 R_1 (Pbest_i^t - X_i^t) + C_2 R_2 (Gbest_i^t - X_i^t) \quad (2)$$

Where: X_i^t and V_i^t denote the position and velocity of the i^{th} particle at time instance t

W^t is the inertial weight at t^{th} instant of time

C_1 and C_2 are positive acceleration constant

R_1 and R_2 are random values generated in the range [0, 1], sampled from a uniform distribution.

The particle swarm optimization technique is initialized with a view to choosing candidates solution randomly within a search space. The algorithm uses the objective function to determine candidate's solution, thereby operating on the resultant fitness values (Blondin, 2009). The general process of implementing particle swarm optimization algorithm is described in(Eberhart & Shi, 2001).The inertia weight w and the acceleration coefficient C_1 and C_2 are the particle swarm optimization parameters which are user supplied. The acceleration coefficient are positive constant within the range of [0 2] while the inertia weight is within the range of [0.8 1.2]. The inertia weight is responsible for keeping the particles moving in the same direction by either suppressing the particles inertia or accelerating the particle in its original ongoing direction (Blondin, 2009). This also controls the particle swarm optimization convergence rate. The values R_1 and R_2 are random values in the range of [0 1]. These values are generated for each velocity update (Shi, 2004).

Some of the advantages of using of Particle Swarm Optimization are as follows (Setayesh *et al.,* 2013):

i. Fewer numbers of parameters: particle swarm optimization is easier to implement because it uses only one parameter which is the velocity calculation.
ii. Using particle swarm optimization algorithm to handle the edge detection in noisy images does not require any post-processing technique (such as a linking technique).

The acceleration coefficient C_1 and C_2 controls how far the particle will move in a single iteration and are usually a random number in the range [0 2]. To improve the performance of the algorithm, the acceleration coefficient C_1 and C_2 are set to equal integer values. Following (Setayesh *et al.,* 2013) the values $w = 0.81$, $C_1 = 1.4962$ and $C_2 = 1.4962$ are used because they are standard parameters that do not depend on the data . The value w is the inertia weight that controls the convergence rate of the particle swarm optimization algorithm. The particle swarm optimization based on vector order statistics edge detection algorithm used a population size of 5 and maximum number of iterations of 10. These parameters are chosen with a view to reducing the computational time during edge detection. In order to measure the quality of a reconstructed image from a noisy environment, a mathematical model is required with a view to determining how much the image has been recovered. These mathematical models are the mean square error (MSE) and peak signal to noise ratio (PSNR) (Ndajah *et al.,* 2011). The mean square error is used as a signal fidelity measure to compare two signals by providing a quantitative score with a view to determining the level of error or distortion between them. The mean square error is calculated using equation (3)(Abdul & Funjan, 2013)

$$MSE = \frac{\sum_{p,q}[I_1(p,q) - I_2(p,q)]^2}{p * q} \qquad (3)$$

Where; I_1 is the reconstructed image

I_2 is the noisy image

p * q is the dimension of the row to column.

The MSE is however usually expressed as the peak signal to noise ratio measure as in equation (4) (Abdul & Funjan, 2013)

$$\mathrm{PSNR} = 10\log_{10}[\frac{T^2}{MSE}] \quad (4)$$

Where: T is the range of pixel intensities in an image.

The value of T is calculated using equation (5) (Ndajah *et al.*, 2011)

$$T = 2^N - 1 \quad (5)$$

For a unit8 image and unit16 type image of 8-bits and 16-bits respectively, the values of T are computed as:

$T = 2^8 - 1 = 255$, $T = 2^{16} - 1 = 65535$

Where N is the number of image bits

METHODOLOGY

The methodology adopted for this work is presented as follows:

i. Reconstruct the image from noisy environment to clean environment using the particle swarm optimization technique.
ii. Using Vector Order Statistics, generate a 3x3 window size pixel as presented in Sadiq *et al.*, (2015b)
iii. Determine the Euclidean distance between each pixel in the window.
iv. Apply a pixel collection scheme to the reconstructed coloured images.
v. Use non maximum suppression to reduce thick edges.
vi. Determine which pixel is an edge or not using a threshold value. Thus generating the final output edge map.

The benchmark images used as the input and as validation for the proposed algorithm were standard images obtained from www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping. The existing proven edge detection algorithms use a single pixel to determine if a pixel is an edge or not and failed to perform in noisy environment. However, this lead to false and broken edges. The significance improvement recorded in the image corrupted by noise was as a result of a collection of pixel based approach for coloured images presented in Sadiq *et al.*, (2015a) in combination with the PSO median filtering technique. The Minimum Vector Range (MVR) was employed with a view to further reducing the presence of noise in the images as presented in Sadiq *et al.,* (2015b). The flow chart of the proven edge detection algorithm and the proposed algorithm is presented in figure 1a and 1b

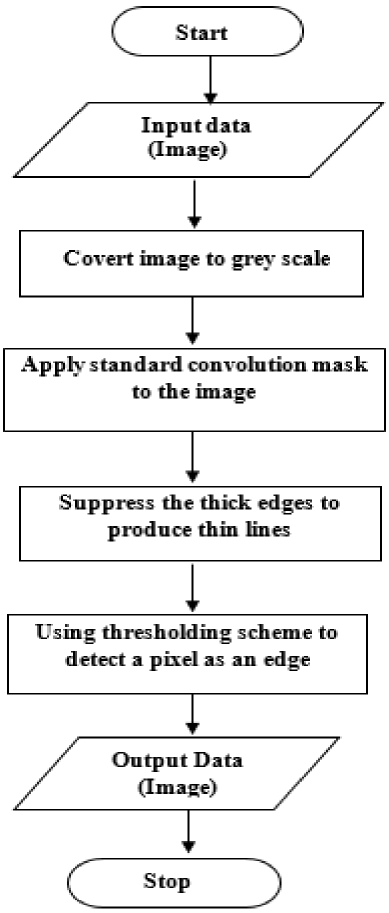


Figure 1a: Flow chart of the proven edge Algorithm

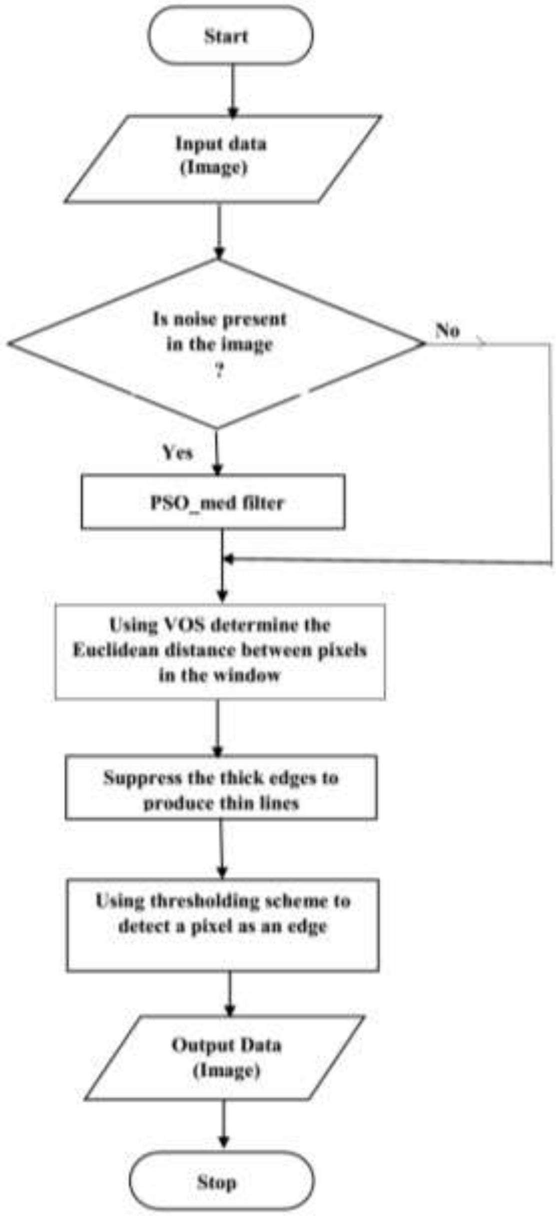


Figure 1b: Flow chart of the proposed Detection Algorithm

INITIALIZING THE ALGORITHM

The algorithm was developed and implemented in MATLAB 2013 script. The step by step approach is described as follows:

Noise Filtering

Input the image

The benchmark images are used as an input. The images are selected from the database of the computer system. Figure 2 shows the extracted image from the computer database.

Figure 2: Extracted Input Image form the Computer Database

1. The program checks if noise is present in the input image. If there exists noise in the image, the particle swarm optimization median filter is initialized.
2. Initialize the position $x(t)$ and velocity $v(t)$ for the particle.
3. For every pixel in the population size, evaluate the fitness function.
4. Generate values for the weight, acceleration coefficient and the random values.
5. Update the position and velocity.
6. When $f(present(t+1)) < f(Pbest(t))$, update the Individual Best for i (particle), the set of weights that yields the (Best Fitness value) minimum MSE.
7. When $f(Gbest(t)) < f(present(t+1)))$, update the Global Best, the set of weights that yields the minimum MSE in a global sense (i.e.,) Best of Individual Best's.
8. The algorithm is iterated until convergence is reached. This convergence yields Gbest (t), the optimal set of weights that minimizes the mean square error. With the set Gbest (t) as weights, the filter estimates the corrupted pixel.

Vector Order Statistics

9. Using the reconstructed image as an input image, generate a 3x3 pixel window from the image.
10. For each pixel in the window a vector of size 3 is used to describe the colour, this is written as $P_{p,q}$ RGB. The vector is the RGB values of that pixel.
11. A new set of scalars A_0 to A_8, is calculated for each pixel by determining the Euclidean distance between a given $P_{p,q}$ RGB and all other $P_{p,q}$ RGB in the window. The Euclidean distance between a pair of vector is the difference in each vector's R value squared added to the differences in G values squared added to the difference in B values squared and the sum in a square root. This result in a 9 distance which is A_0 to A_8.
12. This process is repeated for each pixel in the window, resulting in a single scalar for each pixel. So, the initial 3x3x3 matrix has been transformed into a 3x3x1.
13. The new 3x3x1 matrix is reshaped into a 9x1 array where the index corresponds to the pixel number.
14. The original 3x3x3 window is reshaped in a manner related to the reshaping of the 3x3x1 matrix to result in a 9x3 matrix where the first dimension corresponds the same pixel number as the 3x3x1 array index, and the second dimension is the RGB values for that pixel.
15. The 9x1 array is sorted into ascending order, and the same rearrangement of indices is applied to the first dimension of the 9x3 matrix.
16. The minimum vector range is then calculated by finding the Euclidian distance between the first pixel and the last pixel resulting from the ordering of the sort.
17. If the vector range is above a user set threshold then the window contains an edge, and the centre pixel of the window is set to 1 to represent an edge at that location. Figure 3 shows the output result of the algorithm before edge suppression.

Fig 3: Output of the Vector Order Statistics

18. Suppress the thick edges produced by the vector order statistics with a view to achieving thin and continuous edge lines.

    The output result after suppressing the thick edges is depicted in Figure 4

Figure 4: Output Result of the Suppressed Edges

19. Calculate the Peak-Signal-Noise Ratio that determines the quality of the restored image by using equations 3 and 4

RESULTS AND DISCUSSION

Four images in the output result are displayed in Figures 5. The first image represents the benchmark image before it was corrupted with noise, the second image represents the corrupted image, the third image represents the output of the vector order statistics and the fourth image represents the final output image obtained based on collection of pixels. The peak-signal-noise ratio at different noise levels is shown in Table 1. The peak-signal-noise ratio at different noise levels shown in Table 1 was obtained using equation 3 and 4 as a measure of quality between the original image and the reconstructed image. The higher the peak signal-to- noise ratio, the better the quality of the reconstructed image and the better the detection of edges.

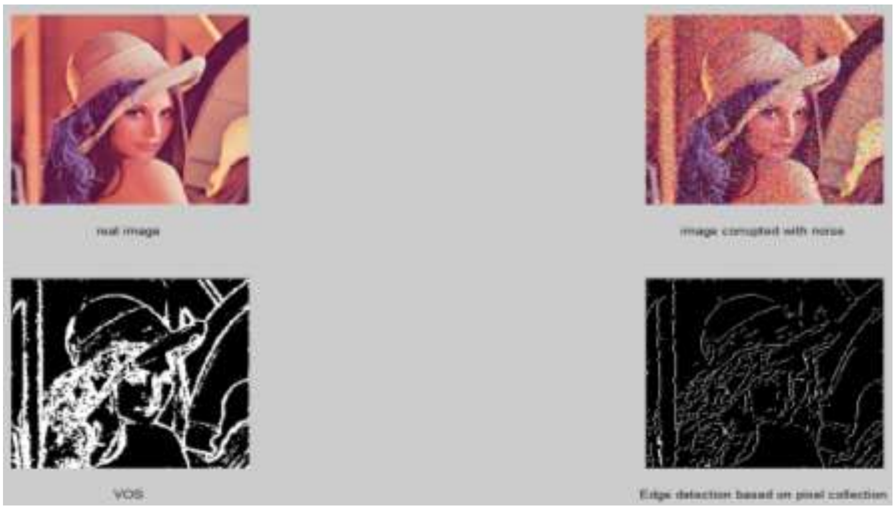


Figure 5: Output Result of Test Image

Table 1: Noise Levels with their Respective Peak Signal-to-Noise Ratio

Percentage pixel affected by noise (%)	PSNR (dB)
10	62.7104
20	59.8679
30	58.1529
40	56.9401
50	54.9414
60	54.1182
70	53.3830
80	53.7189
90	53.0920

The plot of peak-to-signal noise ratio is shown in Figure 6 for the various noise levels. Figure 6 shows that, as the noise level in the images increases, the peak signal-to-noise ratio decreases. This implies that the quality of the reconstructed image decreases with increase in noise level.

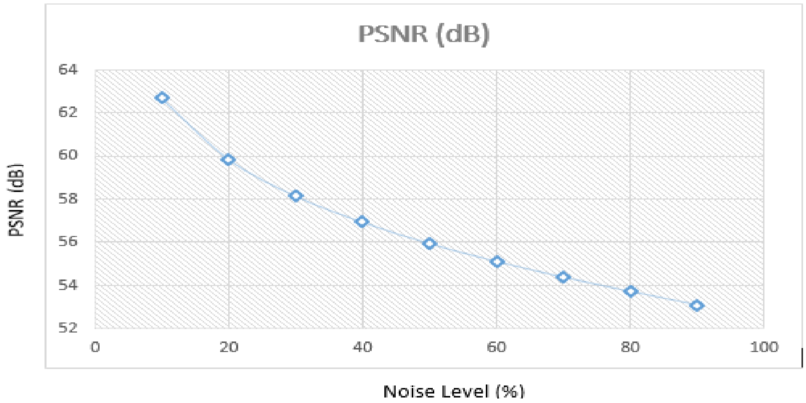
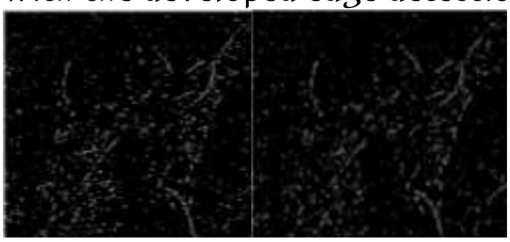


Figure 6: Peak Signal to Noise Ratio for Various Noise Levels

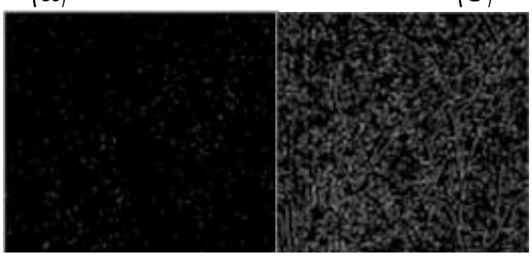
The computation time of the algorithm depends on the type of image to be processed (dimension of the image), the level of noise present in the images and the specification of the computer system running the program. It is also dependent on the number of iterations and population size in the particle swarm optimization algorithm. The specification of the system used to run the algorithm are as follows:

Processor: Intel ® Core ™ i3-2350M CPU @ 2.30GHz
RAM: 4GB
System type: x64-based processor

The Figures 7(a)–7(f) are the output results in noisy environment of the traditional exsisting edge detection algorithms in comparison with the developed edge detection algorithm.



(a) (b)



(c) (d)

(e) (f)

Figure 7: Output Result of Existing Algorithms in Comparison with the Proposed Algorithm

The output result (a) is the Sobel edge Detection, (b) Prewitt Edge Detection Algorithm, (c) Roberts Edge Detection Algorithm, (d) Laplacian Edge Detection Algorithm, (e) Canny Edge Detection Algorithm and (f) Proposed Edge Detection Algorithm

The Figure 7(f) showed that the edges are clearer, visible and jointed that the Figure (a-e). This implies that the proposed algorithm outperformed the existing algorithms.

The best way to determine the performance on an image processing algorithm is by visual appearance. However, the Pratt Figure of Merit (PFOM) is a method used to provide a quantitative comparison between edge detection algorithms in image processing (Dutta &Chaudhuri, 2009). The PFOM is determined by a mathematical expression as in equation 6. The PFOM measures the value of detected edges between 0 and 1. As the value get closer to 1, it shows better detected edge values.

$$R = \frac{1}{Max(N_I, N_A)} \sum_{k=1}^{N_A} \frac{1}{1 + md^2(k)} \qquad (6)$$

Where: N_I is the number of actual edges
N_A is the number of detected edges
m is a scaling constant set to 1/9.
d (k) denotes the distance from the actual edge to the corresponding detected edge

The Pratt Figure of Merit is sensitive to different expected errors, it maximizes when the edge map is perfect and decreases as the error in the edge map increases. The Pratt Figure of Merit measures values between 0 and 1, depending on the quality of the edge detection algorithm used. As the values determine by the Pratt Figure of Merit moves towards 1, it shows best edge detection algorithm (Panetta & Wharton, 2008).

Table 2: PFOM for various Edge Detection Algorithms

Edge detection algorithm	Image with noise
Sobel edge detection algorithm	0.4191
Prewitt edge detection algorithm	0.4191
Robert edge detection algorithm	0.2807
Laplacian edge detection algorithm	0.2811
Canny edge detection algorithm	0.5606
Proposed edge detection algorithm	0.8458

The table for the Pratt Figure of Merit (PFOM) showed that without noise present in the image, the value obtained are closer to the value 1 than the values obtained when noise is present in the image for the existing proven edge detection algorithms. The PFOM value obtained for the proposed edge detection algorithm in clean and noisy environment showed lesser margin than those of the existing proven edge detection algorithm. These results signify that the proposed algorithm performed better than the existing proven edge detection algorithms.

CONCLUSION AND FURTHER WORK

This research work presented an improved edge detection algorithm using particle swarm optimization based on vector order statistics. In order to address the shortcomings of the existing traditional edge detection algorithms that could not process coloured images directly unless been converted to grey scale, Vector Order Statistic technique was employed. The profile edge intensity was applied to generate a collection scheme. This collection scheme was obtained using set of pixels with respect to the step and roof edge profiles. The collection scheme was then applied as a mask in both vertical and horizontal directions to the images. An improved value of 0.8458 was obtained in noisy environment using the PFOM. Further work should implement the algorithm on real physical systems in the areas of biometrics, vehicle detection and tracking, remote sensing amongst others.

REFERENCES

Abdul A. & Funjan W. (2013), Denoising An Image Based On Particle Swarm Optimization (PSO) Algorithm,*journal of Babylon University/Pure and Applied Sciences, Vol.21 No.5,* pp.1511-1518.

Blondin J. (2009). Particle Swarm Optimization: A Tutorial. *retrieved from www.cs.armstrong.edu/saad/csci8100/pso_tutorial.pdf, on 23/11/2014,* pp.1-5.

Chen X. & Chen H, (2010), A Novel Color Edge Detection Algorithm in RGB Color Space,*Institute of Electrical and Electronics Engineers Transcations (IEEE),* pp.793-796.

Dutta S. & Chaudhuri B. (2009), A Color Edge Detection Algorithm in RGB Color Space, *IEEE-International Conference on Advances in Recent Technologies in Communication and Computing,* Vol.1 No.1,pp.337-340.

Eberhart R., & Shi Y. (2001), Particle Swarm Optimization: Developement, Applications and Resources *Institute of*

Electrical and Electronics Engineers Transcations (IEEE), Vol.3 No.1, pp.81-86.

Gupta S.& Mazumdar S. G. (2013). Sobel edge detection algorithm,*International journal of computer science and management Research, Vol.2 No.2*, pp.1578-1583.

Kaur A., & Singh M. D. (2012), An Overview of PSO- Based Approaches in Image Segmentation *international journal of engineering and technology, Vol.2 No.8*, pp.1349-1357.

Ndajah P., Kikuchi H., & Yukawa M. (2011), An Investigation on The Quality of Denoised Images. *INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING, 5*(4), pp.423-434.

Neupane B., Aung Z., & Woon W. L. (2012), A New Image Edge Detection Method Using Quality-Based Clustering. *Proceedings of the IASTED International Conference Visualization, Imaging and Image Processing*, pp.20-26.

Panetta K., & Wharton E. J. (2008), Logarithmic Edge Detection with Applications *Journal of Computers, Vol.3 No.9*, pp.11-19.

Rashmi Kumar M., & Saxena R. (2013), Algorithm and Technique on Various Edge Detection: A survey. *Signal & Image processing: An International Journal (SIPIJ), Vol.4 No.1*, pp.65-75.

Roomi M., & Rajee J. (2011), Speckle Removal in Ultrasound Images Using Particle Swarm Optimization Technique,*IEEE-international conference on recent trends in information technology*, pp.926-931.

Sadiq B.O, Sani. S. M., Garba S, (2015a), Edge Detection: A Collection of Pixel based Approach for Colored Images, *International Journal of Computer Applications (IJCA), Vol.113 No.5*, pp.29-32.

Sadiq B. O., Sani S.M., &Garba S. (2015b), an approach to improving edge detection for facial and remotely sensed images using vector order statistics,*The International Journal of Multimedia & Its Applications (IJMA), Vol.7 No.1*, pp.17-25.

Setayesh M., Mengjie, & Johnston M. (2013), A novel particle swarm optimization approach to detecting continuous, thin and smooth edges in noisy images,*Journal of Image Processing Elsevier Inc.*, pp.28-51.

Shi Y. (2004), Particle Swarm Optimization. *institute of Electrical and Electronics Engineers Transcations (IEEE) Neural Network Society*, pp.8-13.

Solomon C., & Breckon T. (2011). Fundermentals of Digital Image Processing A Practical Approach With Examples in MATLAB (First ed., pp. 1-109): John Wiley & Sons Ltd.

Venkata& Babu J. (2012), Color Image Enhancement Using Praticle Swarm Optimization. *Internation Journal of Engineering Science and Technology (IJEST), Vol.2* No.3, pp.474-480.

Verma R. & Ali J. (2013), A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques,*International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3* No.10, 617-622.

Xin G. & Xiaoguag H. (2012), An improved Canny edge detection algorithm for color image. *Institute of Electrical and Electronics Engineers Transcations (IEEE)*, pp.113-117.