Binary Image Segmentation through the Carl Friedrich Gauss Equation

Oppong-Twum Francis
University of Cape Coast
School of Medical Sciences
Department of Medical
Education / IT
Cape Coast - Ghana

Frimpong Twum
Kwame Nkrumah University of
Science and Technology
Department of Computer
Science
Kumasi - Ghana

J. B. Hayfron Acquah
Kwame Nkrumah University of
Science and Technology
Department of Computer
Science
Kumasi - Ghana

ABSTRACT

Binary image segmentation is a technique that enables an image to be divided into several related portions. In this research, the iterative, Otsu, multiple, adaptive and global algorithms were reviewed to evaluate the importance of the techniques and challenges that limit their usage.

The study utilized the Carl Friedrich Gauss equation to suppress the impact of noisy pixels. The matrix generated was converted into integers to generate a histogram. An arbitrary pixel is selected from the histogram as a threshold to partition the image into two classes. The threshold that generates the minimum variance from the classes is then multiplied by the optimization constant which ranges from 0.1 to 1, and the computed value is used for the segmentation process.

An improved Otsu Algorithm based on the Carl's Friedrich Gauss equation was evaluated with the Otsu, multiple, adaptive, and global algorithms. The signal to noise ratio that defines the sensitivity of a segmentation algorithm, and the running time that specifies the quantum of time required by an algorithm to execute were used as the metrics of performance. The experiments conducted using MATLAB and the Berkeley Image Segmentation Dataset was as follows:

The first experiment consisted of five noise free images. In the experiment, the adaptive obtained the highest sensitivity rating of 8.890dB. This was followed by this studies proposed Twum-Acquah algorithm at 5.623dB. The worst performance was recorded in the global at 2.367dB.

In the second experiment that consisted of noisy images, the proposed Twum-Acquah algorithm obtained the highest performance rating of 4.444dB, while the Adaptive which was at the bottom of the evaluation scored 0.851dB.

In terms of the running time, the fastest algorithms were observed in the global, Otsu and the multiple with a rating of 1.103, 1.264 and 1.392 seconds respectively, while the slowest was recorded in the Adaptive at 129.479 seconds.

General Terms

Binary image Segmentation, thresholding, Algorithms, Histogram, Non-uniform illumination.

Keywords

Twum-Acquah algorithm, Carl Friedrich Gauss equation, thresholding techniques.

1. INTRODUCTION

The main aim of segmenting binary images is to divide a digital image into various related portions pertaining to the image or to make sure the target (objects) in the image are completely separated from their respective background into something that is more relevant and easy to analyse[1].

In this chapter, the study examined the importance of image segmentation which includes the following:

- Medical Imaging: image segmentation enables certain defects in human organs to be identified, such as, the tumour.
- Facial Detection: segmentation enables a person to be identified from either a video frame or a digital image. It achieves this task by comparing the identified facial feature to a face database. The techniques that are normally deployed are the traditional, 3- dimensional recognition and skin-texture analysis [2].
- Vehicle identification in the presence of an occlusion: segmentation algorithm utilizes infrared, lasers and ultrasonic sensors to extract vehicular information even in the presence of an obstacle for a large number of transportation projects which include road and traffic light constructions [3].
- Identification of objects in satellite imaging: The histogram and clustering based segmentation techniques enable the distribution of vegetation, soil types and human settlements to be identified from images captured by a satellite [4].

Despite the importance of image segmentation, there are certain challenges that limit its usage which are elaborated below:

- Non-uniform illumination: intensity changes across a scene can cause some parts of objects to be brighter than others, especially, those in the light regions, which may not establish any relationship with the object in the scene [5].
- Noise: these are unwanted pixels which make segmented objects invisible or difficult to be seen with the naked eye. Image noise which includes the Gaussian, uniform, shot and poisonous noise are normally introduced by capturing devices, such as, the scanners and digital cameras [6].
- Poor histogram distribution: segmentation algorithm produces an undesirable result when several classes (instead of two) are generated from the histograms which are used to compute the threshold for the segmentation process [7].
- High time complexity: this defines the quantum of time required by an algorithm to execute.

Segmentation algorithms that take unreasonable amount of time to execute consume many valuable resources from the computing devices and increased waiting time of potential users. A typical example is the iterative algorithm which never halts execution, until the difference between two computed thresholds is insignificant [8].

1.1 Research Objectives

1. To ensure that the impact of poor illumination is mitigated in the output of image segmentation.
2. To improve the efficiency of Otsu image segmentation algorithm using the Carl's Friedrich Gauss equation by ensuring that minimal amount of time is used during execution.
3. To eliminate the impact of noise which makes detected objects invisible in image segmentation.

2. LITERATURE REVIEW

Image thresholding is a technique used in separating a foreground object from its background. The technique is affected by noise, relative size of object and its background, uniformity of illumination and its reflectance. In reviewing the algorithms for bianary image segmentation (iterative, Otsu, multiple, adaptive, global and sobel), the study analysed both the strength and weaknesses. Some of the thresholding techniques that were identified are as follows:

- Global: This method partitions an image into two dominant modes using a manually selected threshold value. It is regarded as the easiest method in binary image segmentation. The challenge with this method is that segmentation becomes a trivial task when non-uniform illumination is found in the image [9].

- Iterative: In this method, the algorithm automatically selects a threshold based on the differences that exist on successive computed thresholds (generated from the average pixel intensities). This algorithm produces a better result than the global, but has a high time complexity and consumes many valuable resources (CPU and memory) of the computing device [10].

- Adaptive: This method partitions an image into several categories and selects an appropriate threshold for each category for the segmentation process. It performs effectively for non-uniform illuminated images, but has a high time complexity due to the decomposition process [11].

- Multiple: As the name implies, this thresholding technique uses dual thresholds to extract foreground object from the background. It works best for objects with complicated backgrounds for which the global method fails to produce a good output [9].

- Otsu: This algorithm executes by dividing the image into two classes (foreground and background) such that their intra-class variance is negligible. It is the most widely used image segmentation technique, due to its simplicity and effectiveness. The Otsu algorithm fails drastically when several classes are generated from the computed histogram [12].

- The Sobel: This is an edge detection method which explores for pixels that are connected and lies at the mid-point of two regions, where there is a difference in the gray level intensities. The algorithm computes the magnitude of the pixels moving in the horizontal and vertical directions as the extracted edges from the image. The drawback of this algorithm is that it produces many missing pixels which make it difficult to identify the extracted object [13].

Since the Otsu algorithm is the most widely used image segmentation method, this study used the Carl Friedrich Gauss Equation to improve upon the algorithm.

3. METHODOLOGY

In this section, the study utilized the Carl Friedrich Gauss equation to mitigate the challenges that exist in the Otsu algorithm. The materials that were used are set of images from Berkeley Image Segmentation Dataset [14] and MATLAB application for implementation. The steps below outlined the procedures followed in developing the improved Otsu algorithm named Twum-Acquah algorithm:

- The Carl Friedrich Gauss equation expressed in equation 3.1 was used to develop a convolution template to reduce the impact of high frequency pixels that could disturb the visibility of the segmented object.

  $$Pd(m,u) = \left(\frac{1}{2\pi\delta^2}\right) e^{-\frac{(m^2+u^2)^1}{2\delta^2}} \qquad (3.1)$$

  Where *(m, u)* represents the index of the pixel, δ corresponds to the standard deviation with a value of 1.4, л denotes pi (3.142) and *e* signifies Euler's constant with a value of 2.718.
- The template was executed over a set of 5 x 5 matrices extracted from the image, after it has been padded with zeros.
- The minimum variance from each of the intensity values is computed using equation 3.2

  $$\sigma = \sum_{m=q}^{N} \frac{(m-u)^2}{f} \qquad (3.2)$$

  Where σ represents the variance, m corresponds to the pixel intensities starting from the minimum (q) up to the maximum (N), u denotes the mean of the pixels and f is the total frequency.
- The optimum threshold is determined using equation 3.3

  $$T_o = T * O_p \qquad (3.3)$$

  Where T_o corresponds to the optimal threshold, T is the threshold derived from the minimum variance and O_p signifies the optimization constant which ranges from 0.1 to 1. For an optimum performance, *0.9* is selected as the optimization constant.
- The image is segmented using the attained threshold.

The flowchart of the Methodology is shown in Figure 1

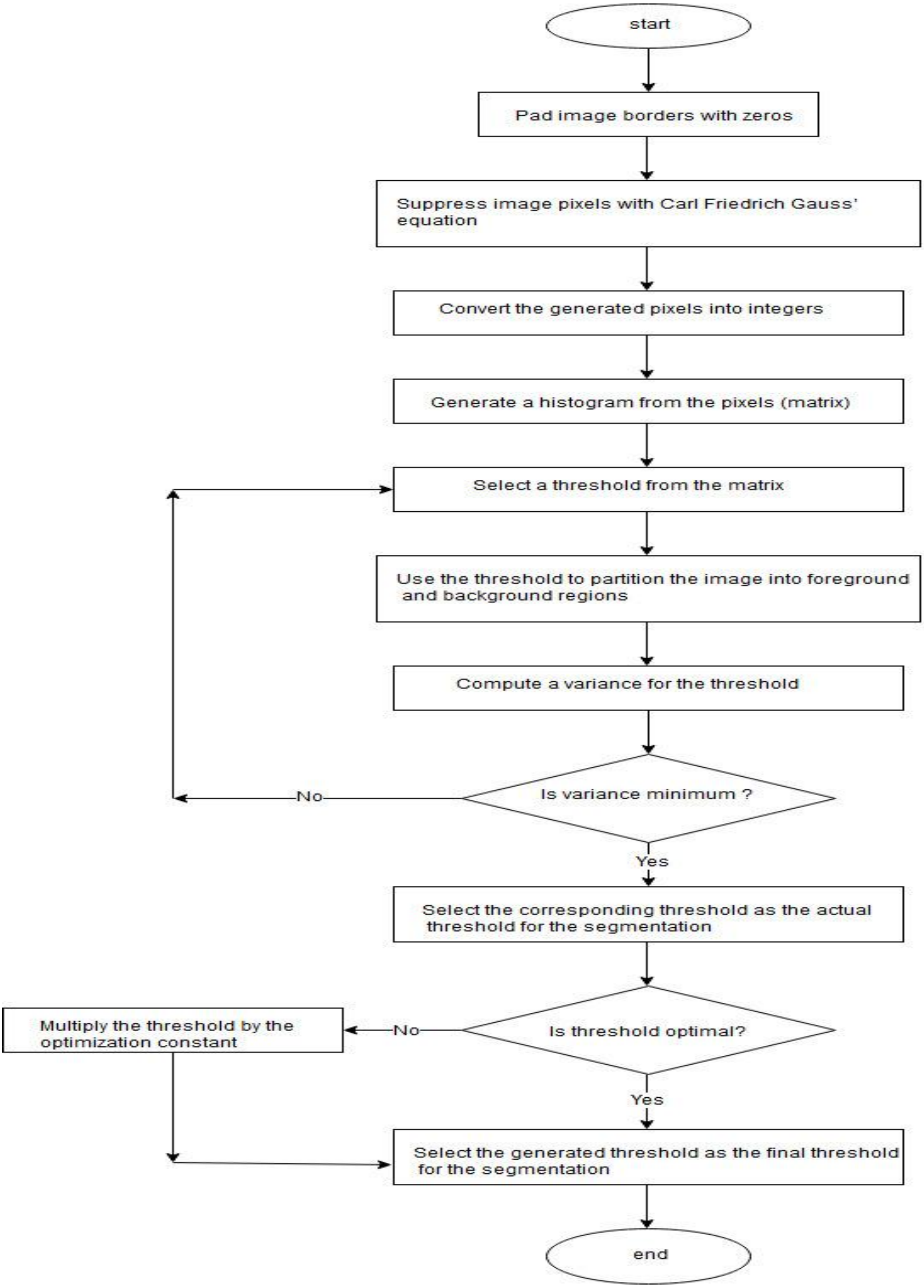
start
Pad image borders with zeros
Suppress image pixels with Carl Friedrich Gauss' equation
Convert the generated pixels into integers
Generate a histogram from the pixels (matrix)
Select a threshold from the matrix
Use the threshold to partition the image into foreground and background regions
Compute a variance for the threshold
Is variance minimum ?
No
Yes
Select the corresponding threshold as the actual threshold for the segmentation
Is threshold optimal?
No
Multiply the threshold by the optimization constant
Yes
Select the generated threshold as the final threshold for the segmentation
end

Figure 1: Flowchart of the design process

4. RESULT ANALYSIS AND DISCUSSIONS

In this chapter, the various segmentation algorithms were compared to each other for the best algorithm. The performance metrics used for the evaluation process are signal to noise ratio (SNR) and running time. The signal to noise ratio that defines the sensitivity of a segmentation algorithm, It is expressed in decibel. Equation 4.1 outlines the calculation for SNR.

$$\boldsymbol{SNR = 10log\left(\frac{signal\,power}{noise\,power}\right)} \qquad (\mathbf{4.1})$$

Where *signal power* corresponds to the mean of the intensity values of the image, while *noise power* denotes the standard deviation of the intensity values.

$$\boldsymbol{MSE = \frac{1}{Q*R}\sum_{n=1}^{Q}\sum_{m=1}^{R}(I(m,n) - G(m,n))^2} \quad (\mathbf{4.2})$$

Where MSE denotes the mean square error which is the error rate between the segmentation algorithm output $I(m,n)$ and the image$G(m,n)$, Q *and* R corresponds to the rows and columns in the image respectively and, m *and* n are the index of the pixel.

The execution time refers to the quantum of time required by a segmentation algorithm to complete execution. It is measured in seconds. The *tic* and *toc* functions in the Matlab enable this computation to be possible. The *tic* command initiates the timer, while the *toc* reads the elapsed time. These commands are put at the beginning and end of each algorithm's implementation (codes). The materials that would be needed for the evaluation are selected images from the Berkeley Image Segmentation Dataset, segmentation algorithms implementation and an evaluator (a Matlab program for evaluating the algorithms). The Matlab program (evaluator) accepts a segmented image from a segmentation algorithm as its argument, and computes a SNR and the Mean Square Error (MSE) for that algorithm.

A total of sixteen images were used for the evaluation. Out of these images, ten were real (noise free), while the remaining six were polluted with the speckle, uniform and the Gaussian noise. This pollution was necessary, so that the performance of the algorithms in the presence of noise could be verified.

Since one evaluation cannot be used to determine the performance of an algorithm, an average performance over all the various categories of the images is computed.

Figure 2 was run through the Otsu, Global, adaptive, Multiple and the proposed Twum-Acquah Algorithm and the output of the Matlab applications of the various algorithms are shown as follows.

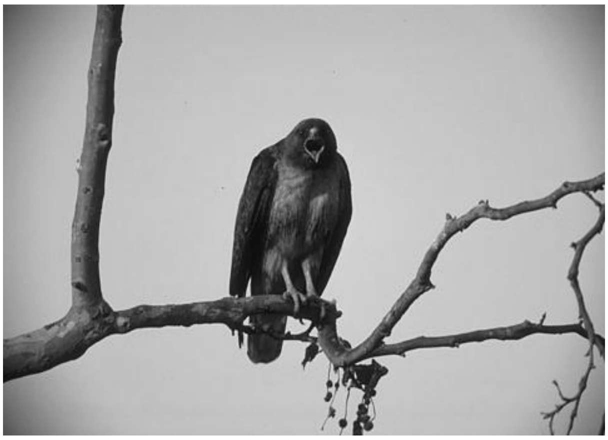

Figure 2: A noise free image of a bird

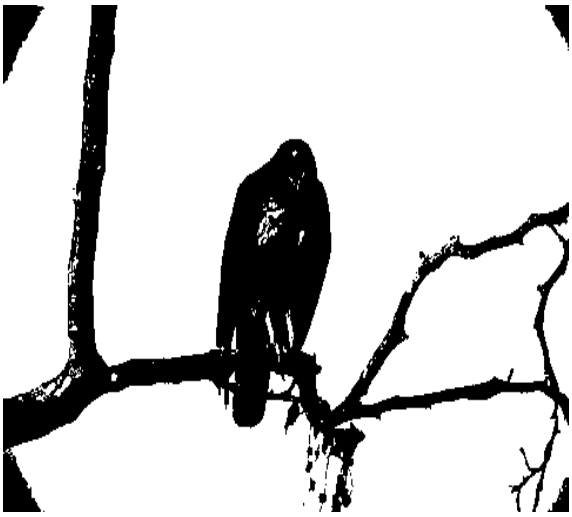

Figure 3: The output of Otsu Algorithm

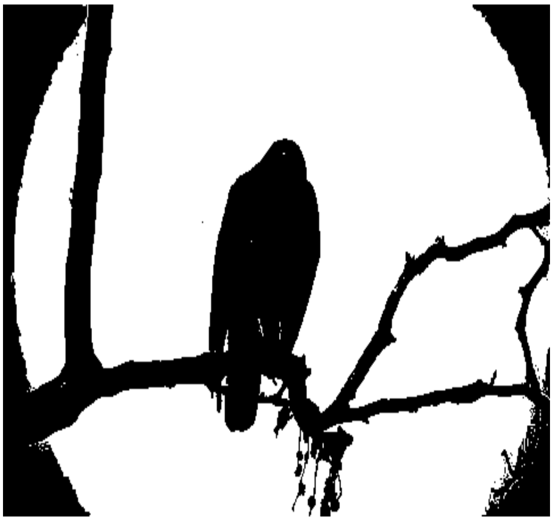

Figure 4: The output of Global Algorithm

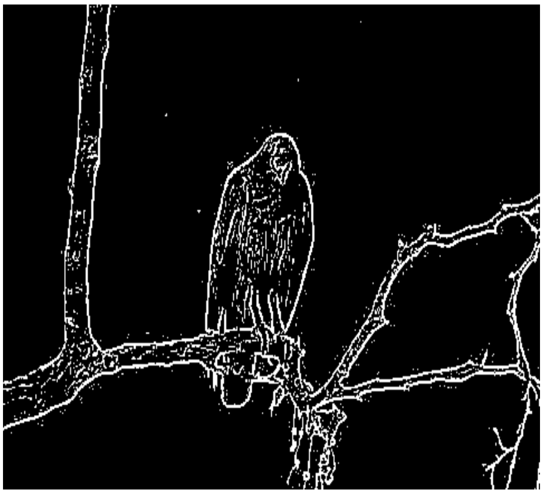

Figure 5: The output of the Adaptive Algorithm

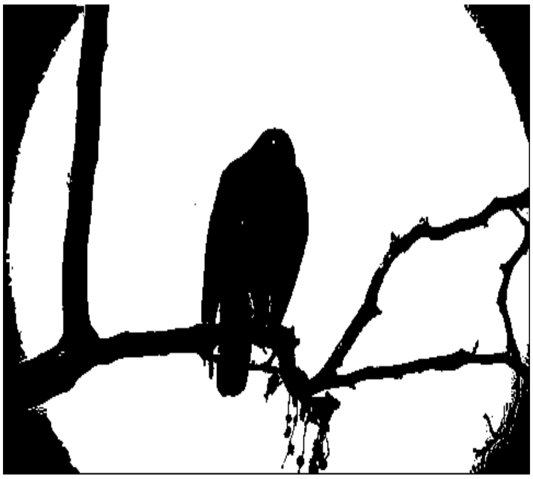

Figure 6: The output of the Multiple Algorithm

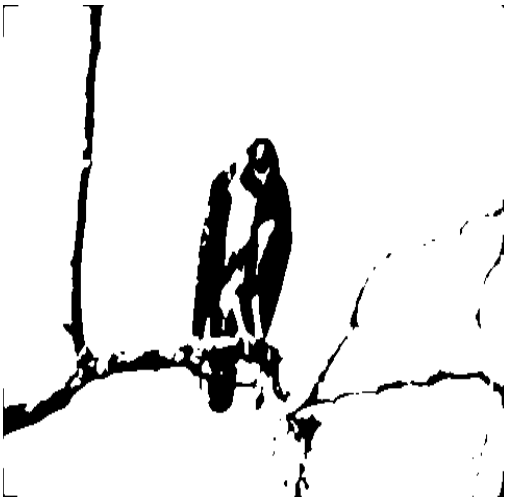

Figure 7: The output of the Twum-Acquah Algorithm

Table 4.1: Evaluating the performance of the algorithms over noise free image

Algorithms	Signal to noise ratio (SNR)	Mean Square Error (MSE)
Otsu	6.2157	0.39458
Global	4.0284	0.45066
Adaptive	11.776	0.24169
Multiple	4.273	0.445
Twum-Acquah	9.688	0.296

Table 4.1 recorded the performance of the various segmentation algorithms over noise free images. From the table, it can be seen that the Adaptive thresholding recorded the highest performance of 11.776 dB. This was followed by the proposed Twum-Acquah algorithm which attained a mean square error of 0.296 dB and a signal to noise ratio of 9.688dB. The least performed algorithm was exhibited by the Global thresholding with the highest MSE of 0.451dB and SNR value of 4.0284. The higher the MSE, the lower the algorithm performs.

Table 4.2 presents the algorithms performance over different noise free images. The average SNR is computed as shown

Table 4.2: Evaluating the average performance of the algorithms over different noise free images

Algorithms	SNR 1	SNR 2	SNR 3	SNR 4	SNR 5	AVG. SNR
Otsu	6.216	2.065	2.321	4.714	2.349	3.533
Global	4.028	1.298	2.313	2.587	1.607	2.367
Adaptive	11.776	7.149	12.982	5.539	7.004	8.890
Multiple	4.273	2.775	2.309	3.717	1.497	2.914
Twum-Acquah	9.688	2.895	2.405	6.930	6.197	5.623

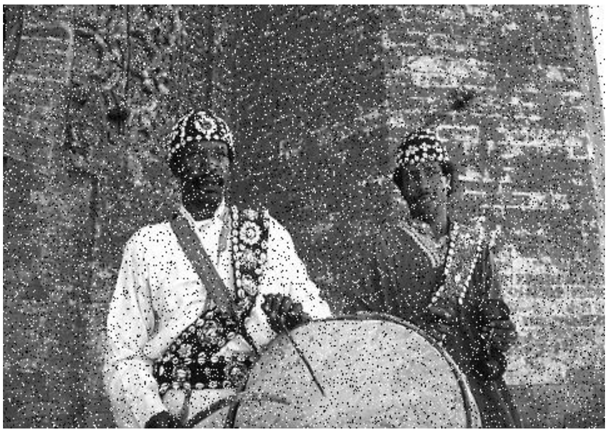

Figure 8: An original image polluted with the Uniform noise.

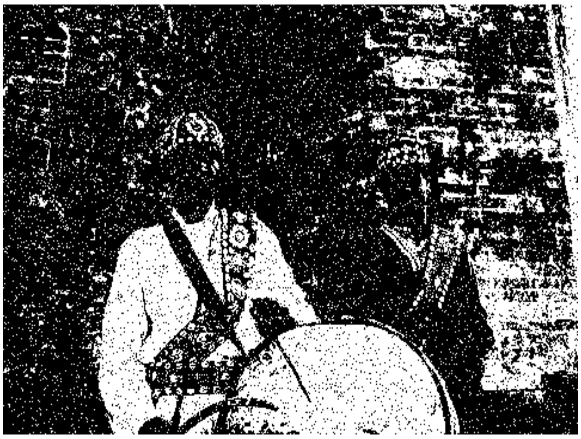

Figure 9: The output of Otsu Algorithm

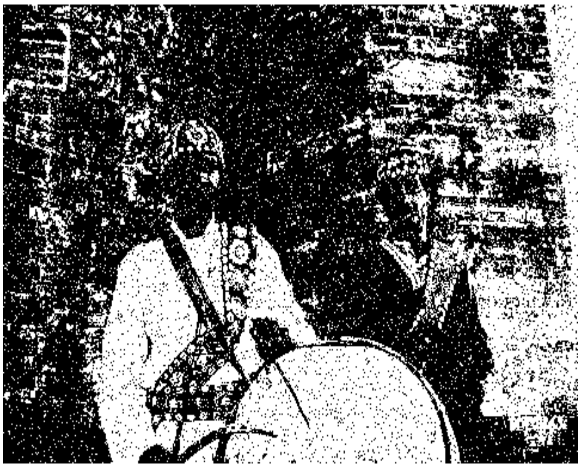

Figure 10: The output of Global Algorithm

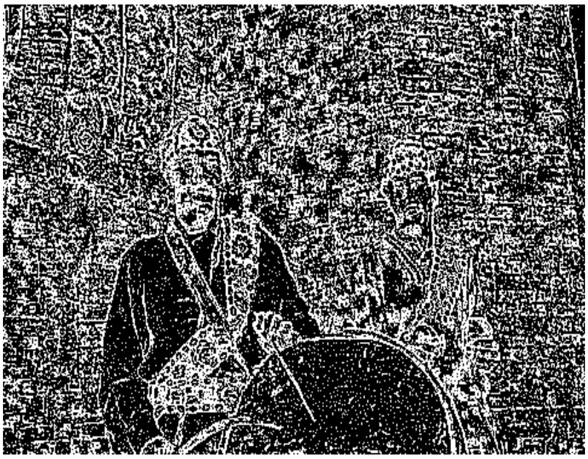

Figure 11: The output of Adaptive Algorithm

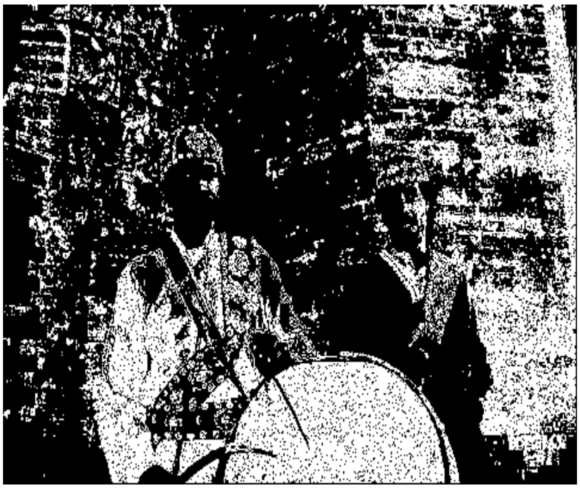

Figure 12: The output of Multiple Algorithm

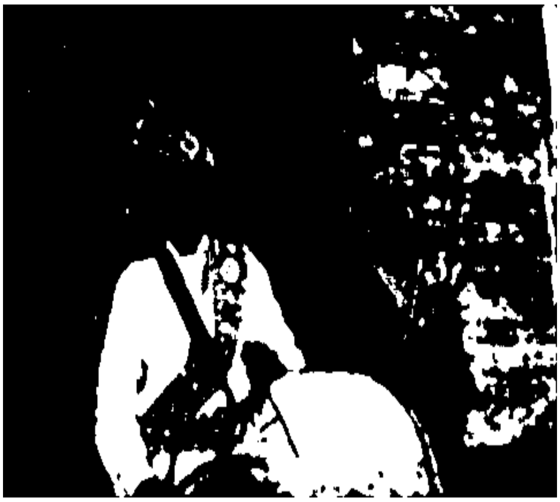

Figure 13: The output of Twum-Acquah Algorithms

Table 4.3: Evaluating the performance of the algorithms over the noisy image

Algorithms	Signal to noise ratio (SNR)	Mean Square Error (MSE)
Otsu	3.402	0.464
Global	1.908	0.488
Adaptive	2.596	0.478
Multiple	3.221	0.468
Twum-Acquah	5.947	0.402

Table 4.3 recorded the performance of the various segmentation algorithms over noisy images.

From Table 4.3, the proposed Twum-Acquah algorithm attained the highest SNR score of 5.947dB, followed by the Otsu with 3.402 dB. The Adaptive algorithm trailed the Multiple with a score of 3.221 dB. The worst performance was recorded by the Global with a score of 1.908 dB.

Table 4.4 outlines the average performance of the algorithms over the noisy images in the evaluation.

Table 4.4: Evaluating the average performance of the algorithms over the noisy images

Algorithms	SNR1	SNR 2	SNR 3	SNR 4	SNR 5	SNR 6	AVG.SNR
Otsu	3.402	1.514	0.200	1.752	0.418	0.400	1.236
Global	1.908	0.404	0.190	1.433	0.114	0.265	0.719
Adaptive	2.596	2.066	0.024	0.248	0.024	0.147	0.851
Multiple	3.221	2.504	0.224	1.854	0.029	0.171	1.334
Twum-Acquah	5.947	3.694	0.892	3.414	5.266	7.446	4.444

In order to address the study objective of improving the efficiency of Otsu image segmentation algorithm using the Carl's Friedrich Gauss equation by ensuring that minimal amount of time is used during execution, the noise free image depicted by Figure 14 was run through the algorithms and the running times were captured as shown in Table 4.5.

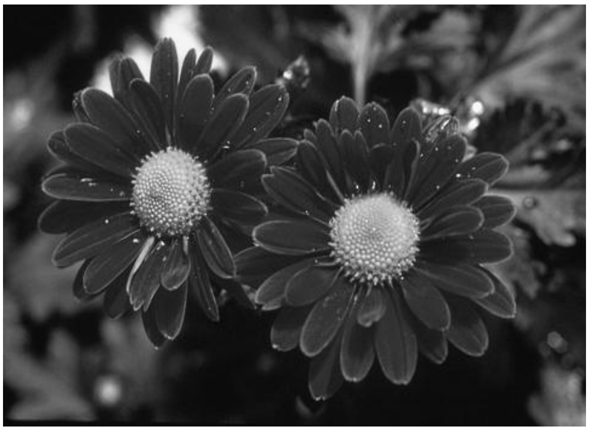

Figure 14: An original image of a flower

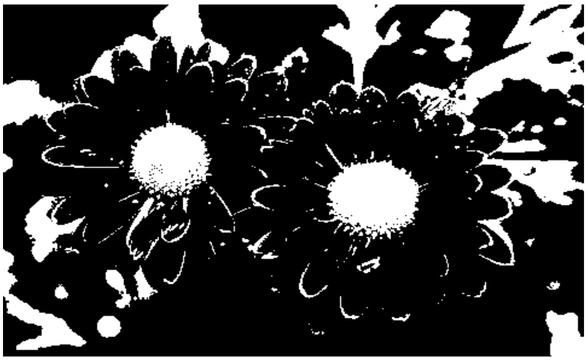

Figure 15: The Output of Otsu Algorithm

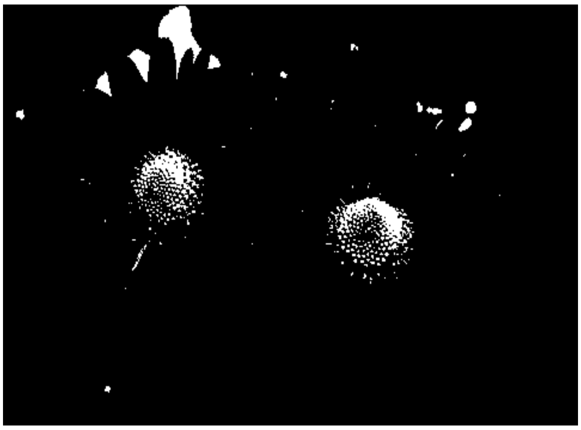

Figure 16: The Output of Global Algorithm

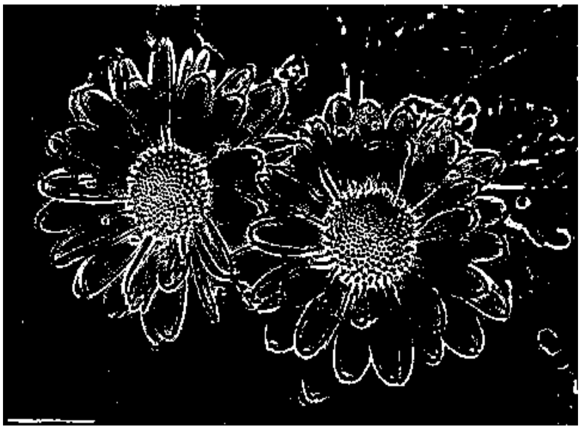

Figure 17: The Output of Adaptive Algorithm

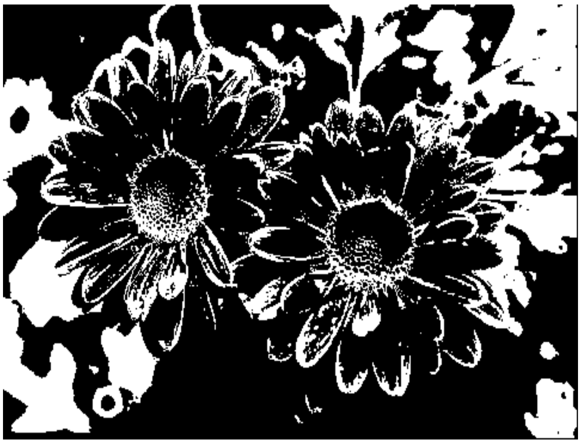

Figure 18: The Output of Multiple Algorithm

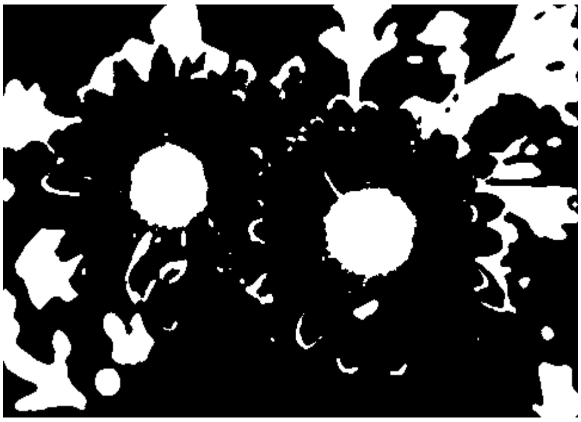

Figure 19: The Output of Twum-Acquah Algorithm

Table 4.5: Evaluating the running time of the algorithms over the selected image

Algorithms	RT 1	RT 2	RT 3	AVG. RT
Otsu	1.562	1.520	1.680	1.587
Global	1.285	1.336	1.247	1.289
Adaptive	142.912	146.088	151.063	146.688
Multiple	1.523	1.537	1.551	1.537
Twum-Acquah	22.710	21.748	22.599	22.353

From the result in Table 4.5, it could be seen that the multiple and the Otsu had almost the same execution time of 1.537 and 1.587 seconds respectively. While the global attained the best execution time of 1.289 seconds, the adaptive was seen with 146.88 seconds, being the worst performance algorithm. Although the performance of the Twum-Acquah algorithm in terms of running time was not the most efficient compared to that of the Global, Multiple, and Otsu algorithms, it was far better than Adaptive algorithm.

The average running time of the algorithms recorded in all the synthetic images are outlined in Table 4.6.

Table 4.6: Evaluating the average running time of the algorithms

Algorithms	Avg.1	Avg. 2	Avg.3	Avg.4	Avg. 5	ovr. Avg
Otsu	1.212	1.303	0.793	1.587	1.425	1.264
Global	1.124	1.200	0.698	1.289	1.202	1.103
Adaptive	145.100	176.074	90.933	146.688	88.602	129.479
Multiple	0.950	1.829	1.547	1.537	1.096	1.392
Twum-Acquah	22.987	24.326	15.507	22.353	16.230	20.281

5. CONCLUSION

The study utilizes Carl Friedrich Gauss equation to mitigate some of the challenges in image segmentation. The first experiment consisted of five noise free images. In the experiment, the Adaptive obtained the highest sensitivity rating of 8.890dB. This was followed by the proposed Twum-Acquah algorithm at 5.623dB. The worst performance was recorded in the Global at 2.367dB.

In the second experiment, a polluted image with the uniform noise was used to verify the performance of the algorithms in the presence of noise. From the result, the proposed Twum-Acquah algorithm obtained 4.444dB, while the second best algorithm (Multiple) had 1.334dB. The worst performance was recorded in the Adaptive at 0.851dB. When the second experiment (1.717dB) was compared to the first (4.665dB), it was realized that the second experiment had trailed the first by 2.948dB. This clearly signifies the poor response of the algorithms to noise.

In the third experiment, five images were selected to compute the running time of the algorithms. From the result, the slowest algorithm was the adaptive that recorded an average running time of 129.479 seconds. The reason for this score is that the algorithm partitions an image into regions, and computes a threshold for each segment before joining them together. This operation consumes so much resources of the computing device. The adaptive algorithm was followed by the Twum-Acquah algorithm at 20.281 seconds. In this algorithm, the Carl Friedrich Gauss equation is used to eliminate any unwanted pixel that could distort the visibility of the segmented object, before a threshold is finally computed for the segmentation process. This operation increases the execution time of the algorithm.

The fastest execution time was recorded in the Global, Otsu and the Multiple. These algorithms compute a single value as a threshold for the segmentation process. Since this task is trivial, a little overhead is put on the computing resources. An average execution time of 1.103, 1.264 and 1.392 seconds were respectively recorded by the Global, Otsu and the Multiple algorithms.

6. FUTURE SCOPE

Since the execution time of the Twum-Acquah algorithm is bad, the researcher would like to improve upon this time and the signal to noise ratio of segmented images in future.

7. REFERENCES

[1] Acharya, T., & Ray, A. K. (2005). Image Processing: Principles and Applications (Vol.15).https://doi.org/10.1 117/1.2348895

[2] Bonsar, K. (2008). How Facial Recognition Systems Work. Retrieved April 28, 2017, from http://computer.howstuffworks.com/facial-recognition.htm

[3] Kanhere, N. (2006). Vehicle Segmentation and Tracking in The Presence of Occlusions. Transportation Research Record, *1944*(1944), 89–97. https://doi.org/10.3141/1944-12

[4] Barghout, L., & Sheynin, J. (2013). Real-world scene perception and perceptual organization: Lessons from Computer Vision. *Journal of Vision*, *13(9)*, 700–710.

[5] Jain, A. (1986). Fundamentals of Digital Image Processing. New Jersey: Prentice-Hall Publisher.

[6] Rohankar, J. (2013). Survey of the Various Noises and Techniques for Denoising the Colour Image. International Journal of Application or Innovation in Engineering & Managemen, *2(*11*)*, 257–289.

[7] Shapiro, L. G., & Stockman, C. G. (2001). *Computer Vision*. New Jersey: Prentice-Hall Publisher.

[8] Sipser, M. (2006). Introduction to the Theory of Computation. Boston MA, USA: Course Technology Publisher.

[9] Rafael C. Gonzalez, R. E. W. (2001). *Digital Image Processing* (3rd ed.). Upper Saddle River, New Jersey 07458: Prentice Hall. Retrieved from http://www-cs.ccny.cuny.edu/~zhu/Capstone2007/DIP-Chapter10-Segmentation-notes.pdf

[10] Bansal, S., & Maini, R. (2013). A Comparative Analysis of Iterative and Ostu ’ s Thresholding Techniques. International Journal of Computer Applications, 66(12), 45–47.

[11] Chow, C. K., & Kaneko, T. (1972). Automatic boundary detection of the left ventricle from cineangiograms. Computers and Biomedical Research, *5*(4), 388–410. https://doi.org/10.1016/0010-4809(72)90070-5

[12] Vala, M. H. J., & Baxi, A. (2013). A review on Otsu image segmentation algorithm. International Journal of Advanced Research in Computer Engineering and Technology, 2(2), 387–389.

[13] Chandrasekar, G. . S. and D. C. (2012). A Compariso n of Various Edge Detection Techniques used in Image Processing. International Journal of Computer Science& I.T(IJCSIT), Vol9,No 1.

[14] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A Database of Human Segmented Images and its Application to Evaluate Segmentation Algorithms and Measuring Ecological Statistics. Proceedings of 8th International Conference on Computer Vision (pp. 416--423).

[15]