Int. J. Advance Soft Compu. Appl, Vol. 13, No. 1, March 2021
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Parallel Algorithm for Smoke Image Detection

Sally Almanasra

Faculty of Computer Studies, Arab Open University, Saudi Arabia
e-mail: s.almanasra@arabou.edu.sa

Abstract

In this paper, a fast smoke image detection algorithm is proposed to extract smoke features of fire. The irregular shapes and color variation of smoke, makes it challenging to detect smoke accurately. The situation is even more complicated in complex sites like forests, where traditional smoke sensors are hard to install and have difficulty detecting smoke. Hence, the power of image processing is utilized. We propose a parallelized smoke image detection algorithm for the original algorithm in [1]. The new algorithm relies on dividing the received images into segments and analyzing these segments using multi-threading techniques. Parallelism is utilized at both the macro and micro levels. One thread is created to handle each received image from the area under surveillance. Concurrently, multiple threads are generated to analyze each image independently from the other received images. Various detection experiments are conducted, and the results show that the proposed algorithm is able to enhance the detection speed without compromising the detection accuracy. Compared with the original algorithm, the proposed algorithm operates with higher speed, making it suitable for monitoring more complex and secure environments.

Keywords: *Image processing, multi-core processors, parallel computing, smoke detection.*

1 Introduction

In the past decade, many early fire detection solutions have been developed to avoid or even minimize the destruction that fire may cause in indoor or outdoor environments. The destruction might not only affect the particular site; it may also have a severe economic impact or, most importantly, put human lives at risk. Many of the common early fire detection solutions rely on detecting physical smoke particles using sensors [2-5]. However, sensor-based solutions are only suitable for indoor applications and require a prediction on the possible source of fires in order to place the sensors close to these sources.

Alternatively, multiple image-based detection algorithms are proposed to support both indoor and outdoor environments. These algorithms offer an effective alternative to detecting smoke, and they are developed to detect smoke in captured images using various image processing techniques. However, recognizing the visual characteristics of smoke in high-quality images received in real-time fashion is considered challenging. This process requires utilizing computation resources to execute a massive amount of processing on the received images from the site under surveillance.

In this paper, we present a parallelized version of the image-based smoke detection algorithm found in [1]. The original algorithm is designed to support sequential processing that includes target extraction, color analysis, and subtracting image blocks. The algorithm shows high accuracy in detecting smoke in both indoor and outdoor environments using colored digital images. Therefore, our proposed parallelized algorithm focuses on the performance aspect of the algorithm by presenting a new multi-threading model that works to support various level of parallelism. Our technique allows for a greater level of control over the parallelized tasks. The tasks are designed to support multiple operations independently, assuring a higher level of parallelism. Our parallelized algorithm is also designed to consider the utilization of multi-core architecture available in current common processors in the market. Such capabilities enable our algorithm to work efficiently on commonly used computing devices and does not require any additional hardware as shown in many other domains [6-9].

The rest of the paper is organized as such: section 2 depicts the survey on related works; section 3 reviews the sequential original algorithm; section 4 presents the proposed parallelized smoke image detection algorithm; and section 5 illustrates an experimental result, followed by the conclusion with future Scope.

2 Related Work

Several methods are proposed to detect smoke based on image processing techniques. Some of these techniques are based on basic image processing methods (e.g. filtering, masking, color analysis, etc.) as in [10-17], clustering techniques [18], fuzzy logic [19], neural networks [20], and machine learning [21]. However, in this section, we will review the methods in the field of parallel computing that are developed to support smoke detection algorithms.

The research in [22] presents a parallel algorithm for smoke detection based on Convolutional Neural Networks (CNN). This algorithm utilizes negative and large-scale smoke/fire videos of both indoor and outdoor environments. For the real-time detection of smoke, the algorithm named YOLOv2 is developed. This algorithm enhances the resolution of the image output by removing one pooling layer for smoke detection. Data and other computational problems in the research are resolved with the help of a parallel computing toolbox along with a GPU and a multi-core processor. The proposed detection method and algorithm in this research stands out for its real-time, accurate performance and low-latency as compared to other traditional methods of smoke detection.

The author in [23] proposes a smoke detection algorithm suitable for urban areas. The algorithm is supported by the CUDA toolkit to accelerate compute-intensive application. This algorithm involves image acquisition, smoothing, and threshold running in the GPU for detecting the smoke, allowing the algorithm to detect smoke in real time. With the help of the NVidia GPU and CUDA, the proposed algorithm is parallelized to achieve the speedup of about 83%. However, for parallel computing, CUDA streams are not explored to acquire the real-time efficiency of the outcome.

Another research study in [24] utilizes parallel computing techniques in image-based smoke detection. This method analyzes the roughness of boundaries and the density of edge pixels in the image. Rigid moving objects are distinguished from fluid smoke regions by making a comparison between previous and current frames. The underlying structure of the method is based on parallel computing using GPUs and the CUDA toolkit, enabling the method to acquire fast processing for both high- and low-resolution video frames.

3 Sequential Smoke Image Detection Algorithm

In this section, we briefly review the basic components of the original sequential image smoke detection algorithm [1]. The algorithm performs two core processes: target extraction and color analysis. The process starts upon receiving real-time images from the area under surveillance through a process called the Target Extractor (TE). This process extracts the target blocks which might be a candidate for smoke block based on the smoke-free base image of the site. The internal structure of the TE is illustrated in Figure. 1.

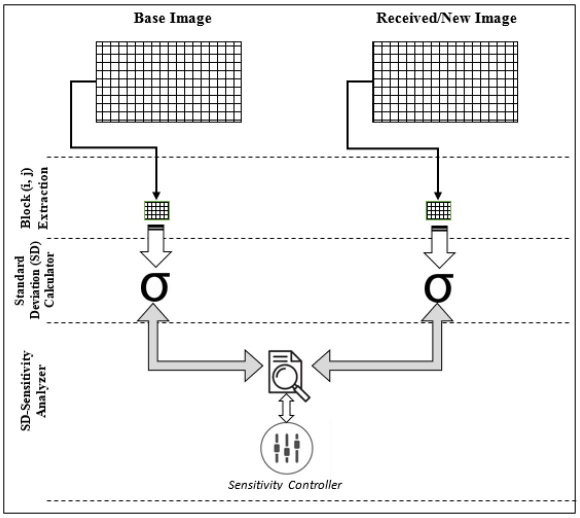


Figure 1. The internal structure of TE Component [1]

After that, the Color Analysis (CA) process is initiated by masking the newly received images to compare their overall intensity against the base image. The CA aims to separate the RGB blocks and examine the standard deviation of the three mean values of the extracted blocks. This process is followed by a curve fitting process that is applied over the block intensity of both the base image and the current image received from the site. Consequently, the final decision on whether smoke is detected or not is based on the difference between the fitted curves of both images.

4 Parallel Smoke Image Detection Algorithm

The proposed parallel smoke detection algorithm is designed based on intensive analysis of the original sequential analysis. The parallelized algorithm is carefully designed to avoid violating the core processes of the sequential algorithm and to assure the same level of detection accuracy. Figure 2 illustrates the sequential nature of the internal design of the original algorithm. The design shows that image *i* is processed using *n* operations over all *m* blocks at time *t*.

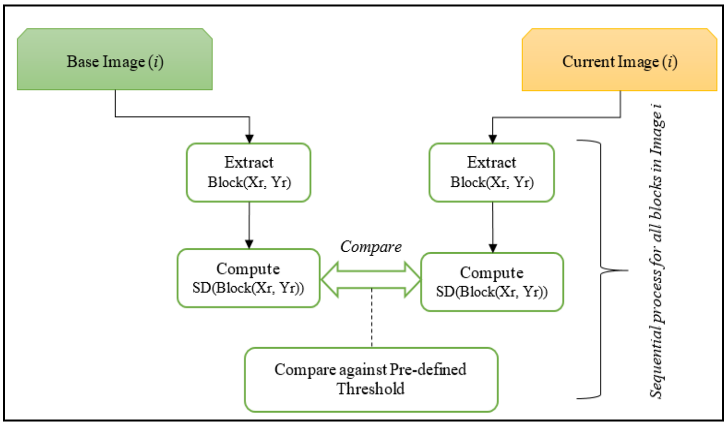


Figure 2. The sequential structure of the smoke detection algorithm

To parallelize these processes, we redesigned the internal components of the sequential algorithm to support parallelism. Our new algorithm is designed to offer the highest possible utilization of the available computation resources available at the users' machine as presented in Figure 3. The Machine Cores Detector starts by detecting the number of cores available in the running processor. This step initiates the macro level of parallelism, where *T* threads are generated and associated with *C* Cores (*i.e. P* = *C*).

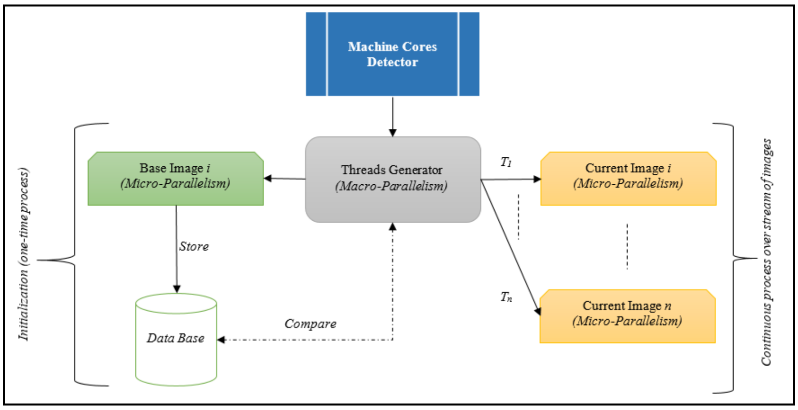


Figure 3. The sequential structure of the smoke detection algorithm

Each thread is responsible for processing one image *i* at time *t*. For instance, if we run our algorithm on a quad-core processor, the algorithm generates four threads to process four images at four cores concurrently. Note that one thread is also generated to process the base image and stores its attributes in the corresponding database. This is a one-time process that is needed at the very beginning stages to enable the algorithm to compare the attributes of the newly captured images against the base image's attributes.

The second dimension of parallelism is designed to support parallelizing the micro operations carried out for processing image *i*. These operations aim to generate attributes of the image by extracting the target blocks, analyzing the colors, calculating standard deviation, and comparing the fitting curve against the base image stored in the database. Initially, each image *i* is divided to four blocks, and each block is handled by a separate thread t_i. Accordingly, each received image is managed by four separate threads as illustrated in Figure 4.

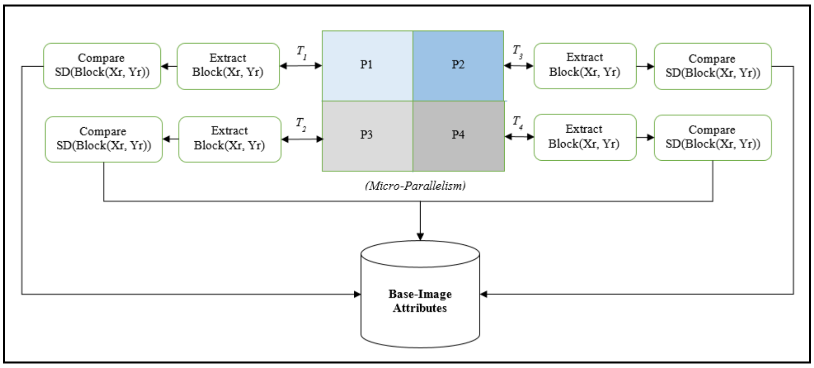


Figure 4. Multi-threading model to process each received image

The micro-parallelism design helps the algorithm to complete the analysis of each part of the image independently and in less processing time. Each thread executes the same set of operations over each part of the image and compares the fitting curve of that particular part against the corresponding part of the base image in the database. This process does not require any intercommunication between the threads, as each thread works on its own resources and generates its own decision. However, the final decision of whether the smoke is present or not is based on the existence of smoke in any of the image's parts.

The design of the algorithm is adaptive to work on any machine and to generate threads based on the available cores in that machine. Such an adaptive design allows the algorithm to fully utilize the available computing resources. In addition, the threads are controlled efficiently to avoid any communication between the threads at both the macro and micro levels.

5 Results and Discussions

Our proposed algorithm is initially tested against the same set of images used for testing the original sequential algorithm [1] using the images found in [25]. This is to examine the impact of the new parallel structure on the accuracy of smoke detection. The base image is divided into four parts and internal operations are applied. The generated attributes of the base image are then stored in the database for further comparison against the attributes of the new images received from the area under surveillance. Figure 5-A shows one of the smoke images used in our experiment, while Figure 5-B presents the intensity of the image before applying the curve fitting process.

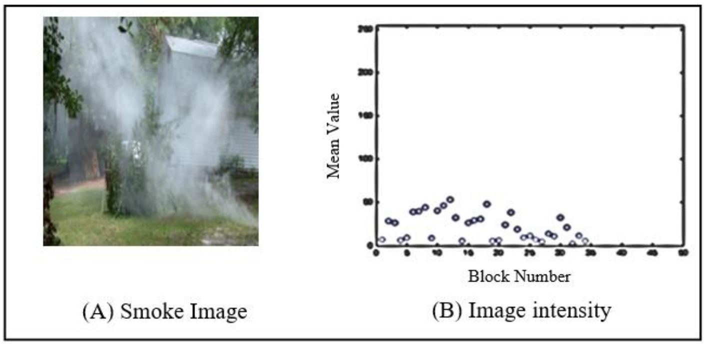


Figure 5. The newly received-image (A) and its intensity (B) before applying curve fitting

Consequently, a mask is applied over the captured image to study the intensity of the image's blocks. Figure 6 shows the resulted masked image (6-A) and its corresponding intensity (6-B). The resulting image clearly identifies the smoke blocks in the image.

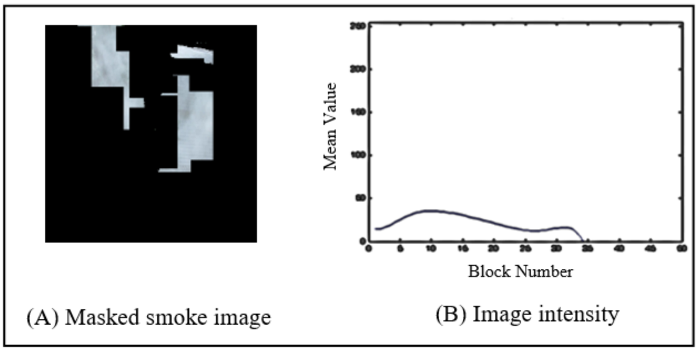
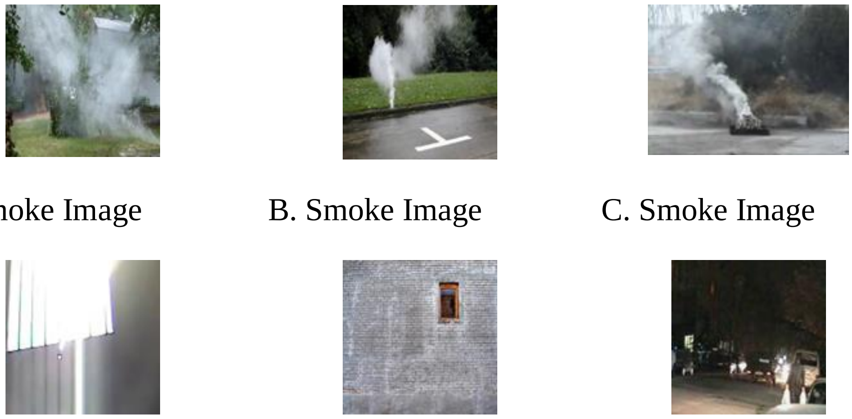


Figure 6. The masked received-image (A) and its intensity (B) after applying curve fitting

The decision of whether smoke is present or not is formulated based on the comparisons between the two fitted curves of both the base and current images. The set of images used in our experiment is shown in Figure 7. The testing results over the set of images show that our method is able to detect the smoke blocks in most of the images similarly to the achieved results in [1].



A. Smoke Image B. Smoke Image C. Smoke Image

D. None Smoke Image E. None Smoke Image F. None Smoke Image

Figure 7. The smoke (A-C) and non-smoke (D-F) images used in our experiment

Figure 8(A-C) show that the masked blocked are recognized as smock blocks, while Figure 8(D-F) show that the masked grey-level blocks are not recognized as smoke. These findings prove that our parallelized structure does not compromise the detection quality nor the accuracy of the original algorithm.

It is noticeable that our parallelized algorithm is able to identify candidate smoke blocks effectively, as illustrated in Figure 8. Similar to the original sequential detection algorithm, our algorithm produces false alarm on some non-smoke images. We run our experiment on over 300 images. The results show that the detection rate is 94.7%, which makes it 0.4% less accurate compared to the sequential version.

In terms of performance, our method is tested on two different computing machines. The first machine has Core i7 processor (dual-core processor) of 2.8GHz, a memory RAM of 3 GB, and a hard disk of 80 GB. This machine is denoted as DCM. The second machine has Core i7 processor (quad-core processor) of 1.8GHz, a memory RAM of 16 GB, and a hard disk of 1 TB. This machine is denoted as QCM. Our parallelized algorithm (denoted as P-Algorithm) is tested on the aforementioned machines, and the results are compared against the sequential algorithm (denoted as S-Algorithm).

A.1 Smoke Image

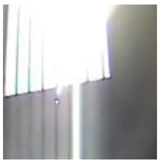
B.1 Smoke Image

C.1 Smoke Image

A.2 detected smoke blocks of A.1

B.2 detected smoke blocks of B.1

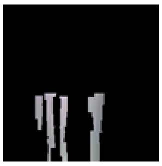
C.2 detected smoke blocks of C.1



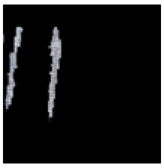
D.1 None Smoke Image

E.1 None Smoke Image

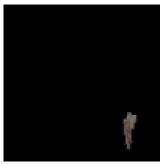
F.1 None Smoke Image



D.2 detected blocks of D.1



E.2 detected blocks of E.1



F.2 detected blocks of F.1

Figure 8. Smoke (A1-C1) and non-smoke (D1-F1) images and their corresponding masking (A2-F2)

Results in Figures 9 and 10 show that the S-Algorithm is not fully utilizing the extra processing units available in QCM compared to the performance gained on DCM. The average enhancement ratio is found to be 14.69% when tested on QCM. This is due to the sequential structure of S-Algorithm that is not capable of splitting tasks among the available cores in the machine.

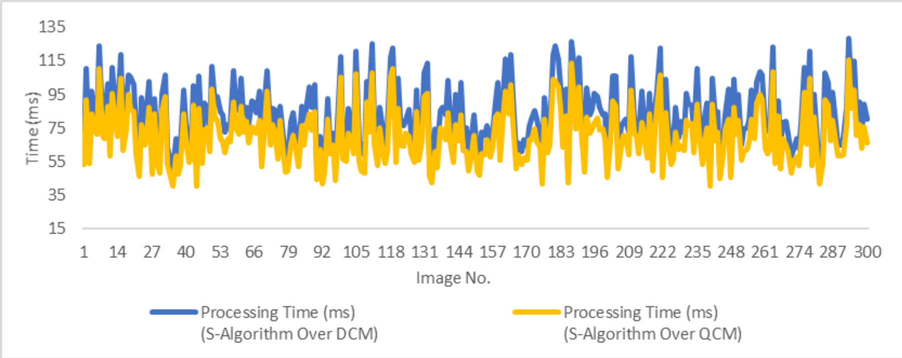


Figure 9. Performance of S-Algorithm on DCM and QCM

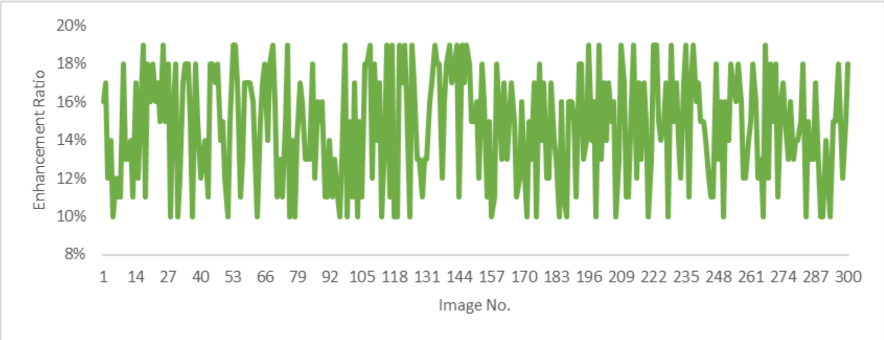


Figure 10. Enhancement ratio achieved by S-Algorithm on QCM compared to DCM

In contrast, P-Algorithm shows a greater utilization of the extra cores available on QCM and DCM. Results in Figures 11 and 12 show that processing time of images decreased significantly on DCM and QCM.

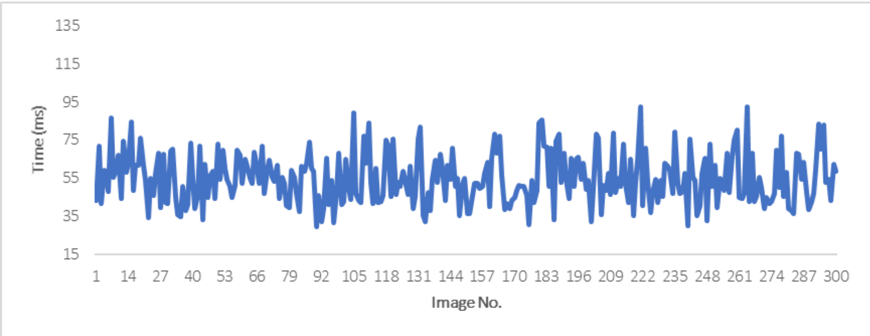


Figure 11. Performance of P-Algorithm on DCM

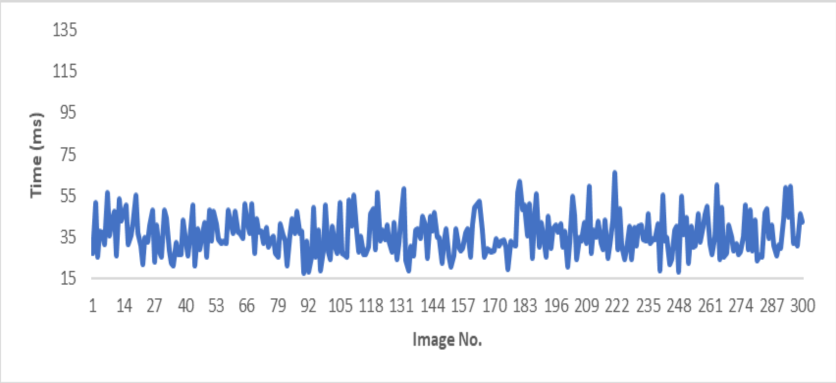


Figure 12. Performance of P-Algorithm on QCM

As our parallelized algorithm supports running multiple threads concurrently, the utilization of the extra four cores in QCM is maximized. Experimental results show that the average enhancement ratio in the processing time is found to be 33.37% on DCM compared to the S-Algorithm, and 55.91% on QCM. This makes the performance gap between the S-Algorithm and P-Algorithm too big, as the S-Algorithm could achieve only 14.69% enhancement on QCM compared to DCM. Figures 13 and 14 illustrate the performance of both the S-Algorithm and P-Algorithm and their enhancement ratios achieved upon running both algorithms on DCM and QCM respectively.

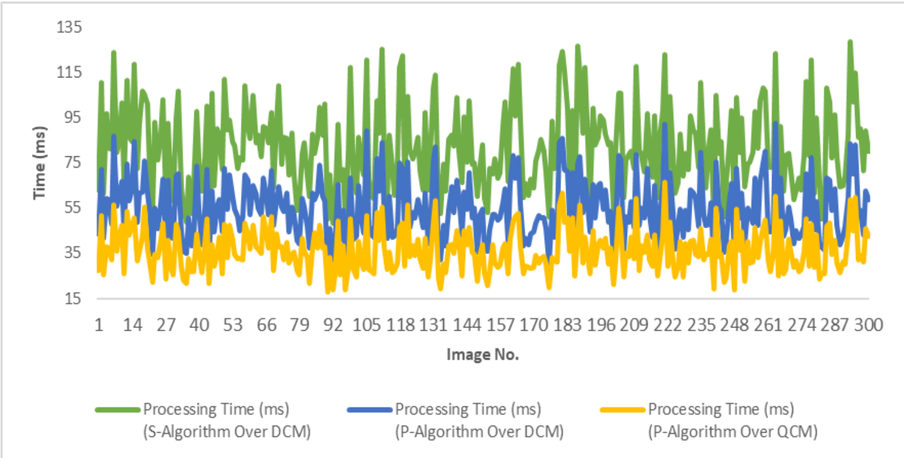


Figure 13. Performance of S-Algorithm and P-Algorithm on DCM and QCM

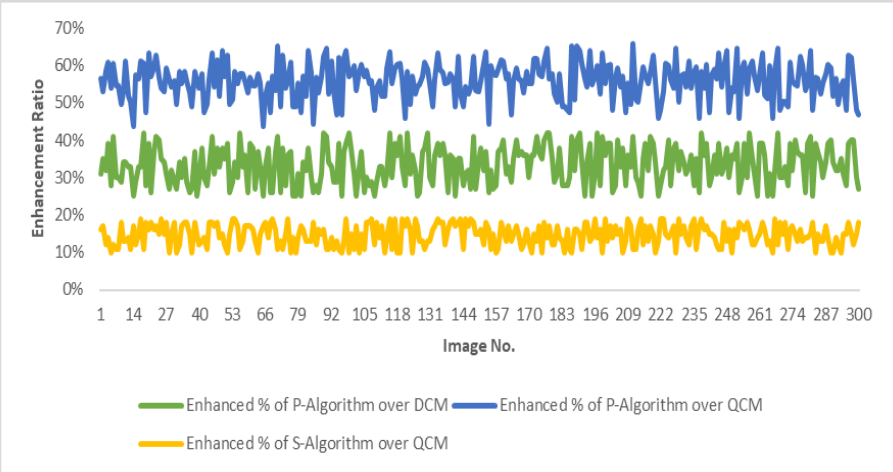


Figure 14. Enhancement ratios achieved by S-Algorithm and P-Algorithm on DCM and QCM

In conclusion, experimental tests show that the performance of the parallel smoke detection algorithm outperforms the original sequential algorithm. The multi-threading scheme implemented in our algorithm played a pivotal role in this achievement. The components of our algorithm are designed to operate independently, which allowed the algorithm to process 2 and 4 concurrent images on DCM and QCM respectively.

6 Conclusion

Sensor-based smoke detection methods are efficient for specific indoor applications. Researchers have proposed several alternative methods based on image processing techniques. Most of these alternatives show high accuracy in detecting smoke in outdoor environments. However, processing images in real time is challenging, as it requires extensive analysis in a short period of time [26-27]. From the structural point of view, the existing techniques are designed to operate sequentially. This leads to less utilization of existing computing power as in multi-core processors.

To overcome this performance issue, we propose a parallelized smoke detection algorithm based on the original sequential algorithm presented in [1]. The algorithm generates multiple threads to perform independent operations on the multi-core processor available in the computing machine. The received image from the site under monitoring is divided into four parts. Each part is then associated with a separate thread to execute a set of image processing operations. The analyzed results of each part are then compared against the base image’s attributes stored in the database.

Results show that parallelism had a positive impact on the overall performance of smoke detection. The parallel detection algorithm is able to process more images at one time, which makes it suitable for monitoring highly sensitive sites.

ACKNOWLEDGEMENTS
The author would like to thank Arab Open University for supporting this study.

References

[1] Almanasra S., Alshahrani A. (2020). Alternative real-time image-based smoke detection algorithm. *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 3, pp. 123-128.

[2] L. Kaiser. (2019). What Is an Air Sampling Smoke Detection System?. Mission Critical Fire Protection, May 11, 2019. [Online]. Available: https://www.orrprotection.com/mcfp/blog/air-sampling-smoke-detection-system.

[3] Shixing L., Xinxin L., Wei Y., Changzheng C., Kun Y., Maoxiang Y., Yongming Z. (2014). Aspirating fire detection system with high sensitivity and multi-parameter. In Proc. 2014 International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, Japan: IEEE.

[4] Trumble, T. (1976). United States of America Patent No. US3982130A.

[5] Bosch I., Serrano A., Vergara L. (2013). Multisensor network system for wildfire detection using infrared image processing. *The Scientific World Journal*, https://doi.org/10.1155/2013/402196.

[6] Adnan A. Hnaif, Khalid Mohammad Jaber, Mohammad A. Alia, and Mohammed E. Daghbosheh. (2020). Parallel Scalable Approximate Matching Algorithm for Network Intrusion Detection System. *The International Arab Journal of Information Technology*, Vol. 18, No.1, pp. 77-84.

[7] Adnan A. Hnaif, Ali Aldahoud, Mohammad A. Alia, Issa S. Al'otoum and Duaa Nazzal. (2019). Multiprocessing scalable string matching algorithm for network intrusion detection system. *International Journal of High Performance Systems Architecture*, Vol. 8, No. 3.

[8] Khaled Suwais. (2014). New Parallel Model for Rabbit Stream Cipher over Multi-core Processor, *WSEAES Transactions on Information Science and Applications*, Vol. 11.

[9] Khaled Suwais and Azman Samsudin. (2008). High Performance Multithreaded Model for Stream Ciphers, *International Journal of Computer Science & Network Security*, Vol. 8(3): 228-233.

[10] Bchir I. (2019). Image-based smoke detection using source separation. *Spectroscopy and Spectral Analysis*, vol. 39, no. 3, pp. 982-989.

[11] Xu Z. (2007). Automatic fire smoke detection based on image visual features. In Proc. 2007 International Conference on Computational Intelligence and Security Workshops, Heilongjiang, pp. 316-319.

[12] Pu R., Lee S. (2011). Study of smoke detection by analysis of saturation in video. In Proc. 2011 International Conference on Fluid Power and Mechatronics, Beijing: IEEE, pp. 169-172.

[13] Hauser M., Li Y., Li J. (2016). Ray A. Real-time combustion state identification via image processing: a dynamic data-driven approach. In Proc. American Control Conference, Boston: IEEE, pp. 3316-3321.

[14] Mutar A., Dway H. (2018). Smoke detection based on image processing by using grey and transparency features. *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 21, pp. 6995-7006.

[15] Mercado J., Medina K., Perez G., Suarez A., Meana H., Orozco A., Villalba L. (2019). Early fire detection on video using LBP and spread ascending of smoke. *Sustainability*, vol. 11, no. 3261, pp. 1-16.

[16] Ye S., Bai Z., Chen H., Bohush R., Ablameyko S. (2017). An effective algorithm to detect both smoke and flame using color and wavelet analysis. *Pattern Recognition and Image Analysis*, vol. 27, no. 1, pp. 131-138.

[17] Lee C., Lin C., Hong C., Su M. (2012). Smoke detection using spatial and temporal analyses. *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 7, pp. 4749-4770.

[18] Fujiwara N., Terada K. (2004). Extraction of a smoke region using fractal cording. In Proc. IEEE International Symposium on Communications and Information Technology, Sapporo: IEEE.

[19] Silva M., Abu Baker M., Petra M. (2017). State of the art of smoke and fire detection using image processing. *Int. J. Signal and Imaging Systems Engineering*. vol. 10, no. 1, pp. 22-30.

[20] Xu G., Zhang Y., Zhang Q., Lin G., Wang Z., Jia Y., Wang J. (2019). Video smoke detection based on deep saliency network. *Fire Safety Journal*, vol. 105, pp. 277-285.

[21] Avgerinakis K., Briassouli A., Kompatsiaris I. (2012). Smoke detection using temporal HOGHOF descriptors and energy color statistics from video. In Proc. International Workshop on Multi-Sensor Systems and Networks for Fire Detection and Management, Antalya.

[22] Saponara S., Elhanashi A., Gagliardi A. (2020). Real-time video fire/smoke detection based on CNN in antifire surveillance systems. *Journal of Real-Time Image Processing*, https://doi.org/10.1007/s11554-020-01044-0.

[23] Antoun W., Jalloul M. (2019). Real-time fire detection using CUDA. In Proc. ISC High Performance Conference, Beirut: ISC-HPC.

[24] Filonenko A., Hernández D.C., Jo K.H. (2017). Fast smoke detection for video surveillance using CUDA. *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp.725-733.

[25] Q. Zhang, "Research webpage about Smoke Detection for Fire Alarm," State Key Laboratory of Fire Science, March 10, 2020. [Online]. Available: http://smoke.ustc.edu.cn/datasets.htm.

[26] Zainudin, Z., Shamsuddin, S.M. (2019). Hasan, S. T. Deep learning for image processing in WEKA environment. *International Journal of Advances in Soft Computing and its Applications*, 11 (1), pp. 1-21. 28.

[27] Nugroho, H.A., Khusna, D.A., Frannita, E.L. (2019). Detection and classification of breast nodule on ultrasound images using edge feature. *International Journal of Advances in Soft Computing and its Applications*, 11 (1), pp. 64-80.

Notes on contributors

Dr. Sally Almanasra is assistant professor at the faculty of computer studies, Arab Open University, Saudi Arabia. Her main teaching and research interests include Game Theory, Image Processing, and Parallel Computing.