UNIVERSITY OF
FORWARD
THINKING
WESTMINSTER

WestminsterResearch
http://www.westminster.ac.uk/westminsterresearch

Mobile Hardware Based Implementation of a Novel, Efficient, Fuzzy Logic Inspired Edge Detection Technique for Analysis of Malaria Infected Microscopic Thin Blood Images

Bias, S., Reni, S. and Kale, I.

This is the published version of the article Bias, S., Reni, S. and Kale, I. (2018) Mobile Hardware Based Implementation of a Novel, Efficient, Fuzzy Logic Inspired Edge Detection Technique for Analysis of Malaria Infected Microscopic Thin Blood Images. Procedia Computer Science., 141, pp. 390-395.

It is available at https://dx.doi.org/10.1016/j.procs.2018.10.187.

© 2018 The Author/s. It is made available under the CC-BY-NC-ND 4.0 license https://creativecommons.org/licenses/by-nc-nd/4.0/

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch: ((http://westminsterresearch.wmin.ac.uk/).

In case of abuse or copyright appearing without permission e-mail repository@westminster.ac.uk

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 141 (2018) 374–381

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2018)

Mobile Hardware Based Implementation of a Novel, Efficient, Fuzzy Logic Inspired Edge Detection Technique for Analysis of Malaria Infected Microscopic Thin Blood Images

Stephen Bias[a], Dr Saumya Reni[b], Prof Izzet Kale[b]

[a]*Network Rail Infrastructure Ltd, 1 Eversholt Street, London, NW1 2DN, UK*
[b]*University of Westminster ADVRG, 115 New Cavendish Street, London, W1W 6UW, UK*

Abstract

This paper proposes a novel, efficient, low complexity algorithm for edge detection, with a cheap, easily accessible, networkable hardware implementation, specifically focused on the analysis of malaria infected thin blood smears. The algorithm presents a new and dynamic thresholding technique that eliminates inter-cell interference based on histogram analysis. Following this, binary image morphological processing is performed which is shown to outperform the same operation on the much more complex greyscale images. Edge tracking is done via a simplified fuzzy logic inspired rule system. The entire system is implemented on multiple platforms to test widespread compatibility but primarily developed for a battery powered standalone raspberry pi with low power, low resolution touchscreen and hardware buttons. The entire algorithm was pitted against the much more complex but still very well performing Canny algorithm, which despite the age, is still one of the most comprehensive edge detection techniques available; modern variants were considered and reviewed, but ultimately given the level of outperformance, they were not viable options.

© 2018 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Selection and peer-review under responsibility of the scientific committee of ICTH 2018.

Keywords: Canny; Edge detection; Fuzzy logic; Giemsa staining, Hardware; Histogram; Malaria; Morphological processing; Morphology; Networkable; Platelets; Rapsberry Pi; Red blood cells; VHSIC; White blood cells

*Corresponding Author. Tel.:+44(0)20 7911 5000 ; Fax: +44(0)20 7911 5037. E-Mail: S.Reni@westminster.ac.uk

1877-0509 © 2018 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Selection and peer-review under responsibility of the scientific committee of ICTH 2018.
10.1016/j.procs.2018.10.187

1. Introduction

In the current landscape of computer vision, edge detection is fundamental and has a wide array of uses spanning medicine to traffic (as shown in [1] and [2]). Increased demand for faster, more efficient software and hardware solutions to increasingly complex problems has led to an increase into the research done into this field. Automatic disease detection is not a new field and has been the subject matter of papers for quite some time now; however, modern examples continue to increase in complexity in order to deal with the demand of more complex and sensitive applications. Primarily edge detection works to identify and mark the edge points of an image but environmental factors, capture equipment, and subject matter all have varying degrees of effects on the image quality, meaning full parametrization of the algorithm is often needed in order to gain the best results across a wide range of inputs. Malaria has been a global epidemic for years and the latest available WHO (World Health Organisation) malaria fact sheet (2018) estimates that in 2016 it killed 445000 people [3]. The disease is transmitted in the bites of infected *Anopheles* mosquitos and is caused by the plasmodium parasite. The standard diagnosis is the analysis of thin blood films that show the presence of the disease indicated by the giemsa stained parasitic components as well as platelets, debris, and other artifacts, making these images particularly difficult to work with.

An application of particle swarm optimization (PSO) and fuzzy c-mean (FCM) as a solution to edge detection issues in CT scan is suggested in [1], however already implementing methods such as these would prove to be incredibly counter intuitive to the goal of this algorithm, which is to reduce complexity as much as possible. The results in provided in [1] are tangible but the subject matter varies massively to ours and isn't directly comparable. A simple fuzzy logic rule-based algorithm is described in [4] which uses very simple input images and offers a viable and computationally cheap method by way of comparator operations. However, these rules aren't comprehensive enough to define how edges would form in real images. Another simple fuzzy logic-based edge detection is proposed in [5] where the membership function to the fuzzy set is distinctly resembling of a threshold, where the membership to the set is predefined and in no way dynamic, where the results show that it works to mark edges, it struggles with textured surfaces.

This paper focuses on the reduction in complexity, feasible on a mobile or low power portable platform and presents a new, novel edge detection technique for use on microscopic thin malaria blood films. The inspiration for the algorithm presented comes from a fuzzy logic rule system that can dynamically decide where to place the threshold similar to that presented in literature [4] and [5]. In this paper, the baseline will be the well performing canny algorithm, where complexity will be the (average) number of operations per pixel as well as the type of operation. Furthering the cause to keep complexity down but still needing to provide and exemplar noise removal technique, wavelets [6] and median filters [7] and adaptive methods such as in [8] were reviewed, but with complexity reduction being our remit, these were avoided due to the long and complex methods implemented. A study of the different types of noise was undertaken in [9] and shows in detail the types of noise that are expected, considering this an integrated solution to noise removal was implemented dealing with many types of destructive noise and removing unwanted artefacts and entities, and completing the cell structure to aid segmentation

The algorithm presented is described in [10] and shows the simple construction and in-depth discussion of the method used to segment the images, in 3 straightforward steps

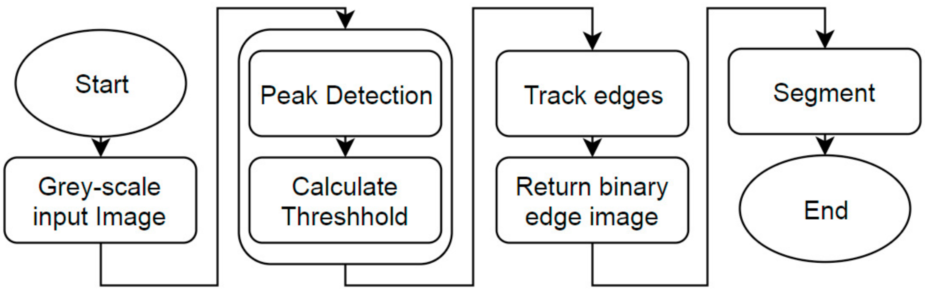


Fig 1. System level diagram

2. Materials and Method

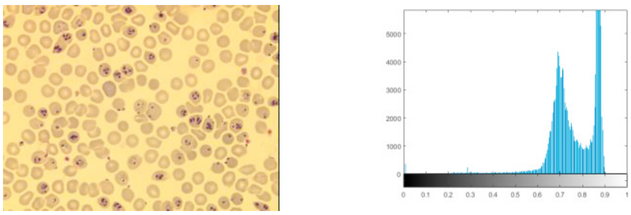
Shown in fig.1. is the system level diagram, it shows the high-level steps taken to segment the image. The algorithm can be split into 3 key areas and are run in this order:

- Threshold acquisition and application
- Unwanted artefact removal
- Edge tracking and segmentation

Every aspect of the algorithm was crafted with consideration for efficiency and complexity. While there are many suitable existing algorithms, the recent rise in cheap and affordable but low power mobile computing has driven the creation of a more appropriate process.

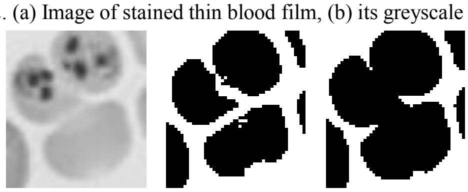
2.1 Thresholding

The first process is a dynamic threshold acquisition that is calculated via histogram analysis (can be seen in fig.2.), whereas modern thresholding techniques lean towards localised thresholding. This is a group operations and requires significantly more operations per pixel as each pixel is interrogated at least once individually, this method works with 255 values versus the 195072 in the 512(W) by 381(H) sample images. The algorithm use parametrised manually trained peak detection. The largest peak is assumed to be the background peak and each subsequent peak is an object peak. The objects that are to be segmented are significant and are assumed to be at least 1% of the total number of pixels. It was found in experimentation that the value between the background peak and the red blood cell (RBC) peak worked well as a threshold but caused significant inter-cell interference (fig.3.) and thus was weighted towards the object peak as to combat this issue. Thresholding acts as a blanket noise suppression, as it forces noise to be on either one side of the boundary or the other. The result is a simple binary image that is computationally much easier to work with and includes all information needed as well as platelet and other unwanted artefacts that should be marked for removal.



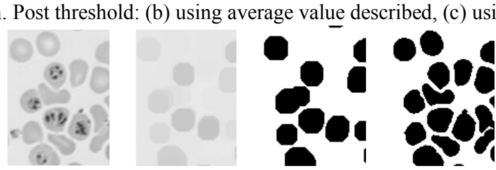
(a) (b)

Fig. 2. (a) Image of stained thin blood film, (b) its greyscale histogram



(a) (b) (c)

Fig. 3. (a) Original section. Post threshold: (b) using average value described, (c) using mid value between the peaks



(a) (b) (c) (d)

Fig.4. (a) section of blood sample, (b) after morphological closing (c) post- threshold of (c), (d) Post threshold using proposed algorithm

2.2 Platelet and artefact removal

Platelets, cell debris, and other artefacts are not the focus of segmentation and hence should not be tracked for edges. The centers of the cells that have an intensity value that matches the background in the original greyscale image, and these should not be segmented either. The simple procedure to remove platelets and fill in the centers is as follows:

- All of the white (background and cell centers) comprised of 500 or fewer connected pixels are removed
- Morphological closing with small disc shaped structuring element to remove any entities tied to cells and further reduce inter-cell interference
- All of the black (foreground) components comprised of 150 or fewer connected pixels are removed

The values of 500 and 150 come from experimentation and are based on the approximate magnification and field of view while taking the image along with the resolution of the image and relative expected sizes of the center pallor of an RBC. All RBCs are expected to be within a certain size range a ratio can be derived from these numbers and applied to images of different resolution and zoom. This combined with thresholding make for effective noise removal so much so that extra filtering is not needed. These operations (or equivalents thereof) where tried on greyscale images, to ensure that there wasn't a significant loss of data in the binary version – while complexity reduction is important, it is critical to retain as much information about the RBCs as possible given their sensitive nature. In fig.4. we can see the binary version is significantly better at retaining information and at separating out cells too. The reduction in complexity also means less image manipulation thus leading to a greater resemblance to the input image.

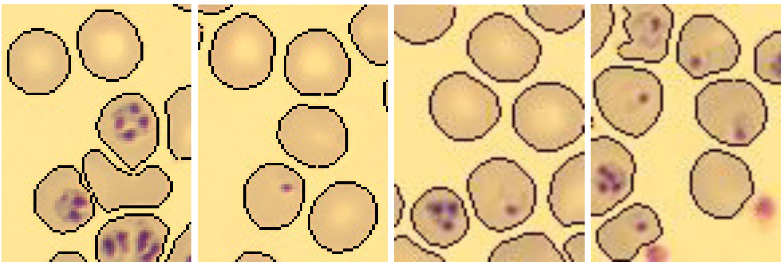
1.3 Edge tracking and segmentation

Defined in fig.5. are the 8 semi-ambiguous kernels that define how edges form in binary images. To track the edges, the output of a moving 3 by 3 window is compared to these rules, and if a match is found, the corresponding origin pixel in the original image is marked as an edge (in this case marked black). After multiple experiments these rules were refined and parts of the window need not even be interrogated, as regardless of their state, the origin will be marked as an edge. This method requires absolutely no multiplication or division operations, and is based entirely on additions and comparators that are significantly quicker.

Slices of a sample with the edges marked is shown in fig.6. and is very representative of the samples used. It contains almost exactly the average number of cells, it has large areas of no cells, small clusters and overlapping cells; large amounts of staining and heavy parasitic presence.

$$\begin{bmatrix} 0 & 0 & X \\ X & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X & 0 & 0 \\ 1 & 1 & X \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & X & 0 \\ 1 & 1 & 0 \\ 1 & 1 & X \end{bmatrix} \begin{bmatrix} 1 & 1 & X \\ 1 & 1 & 0 \\ 1 & X & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ X & 1 & 1 \\ 0 & 0 & X \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & X \\ X & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & X & 1 \\ 0 & 1 & 1 \\ X & 1 & 1 \end{bmatrix} \begin{bmatrix} X & 1 & 1 \\ 0 & 1 & 1 \\ 0 & X & 1 \end{bmatrix}$$

Fig. 5. Kernels defined



(a) (b) (c) (d)

Fig.6. Examples of segmented cells

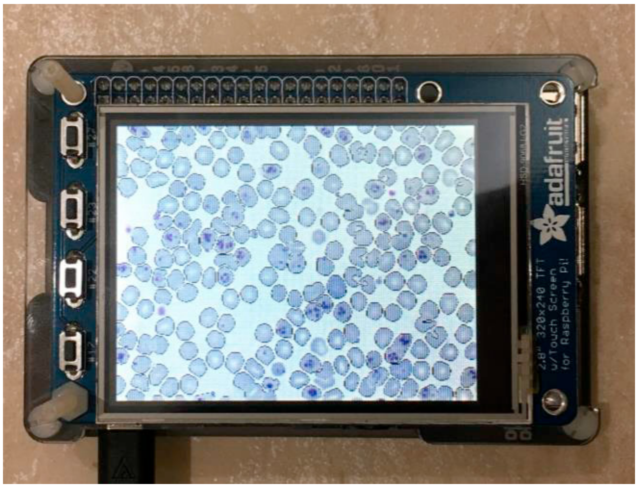


Fig.7. Raspberry Pi with attached screen, displaying segmented sample

2.4 Hardware implementations

There were two options explored for hardware implementation, the first was on an FPGA (field programmable gate array) using very high speed integrated circuit hardware description language (VHSIC). Due to the algorithms dependency on comparators, and relatively simple process, it provides speed and versatility. The VHSIC code was synthesized and successfully simulated, however the FPGA is comparatively expensive, bulky, and often needs specialist knowledge. With the largest portion of the mobile processor market and widest availability, an ARM based system was considered more palatable, so a Raspberry Pi 3 (RPi3) was decided on. Readily available, cheap, versatile and huge amounts of documentation available openly with very little specialist knowledge needed to operate.

The setup shown in fig.7. was a battery powered RPi3 with attached low-resolution touchscreen and hardware buttons for operation. A huge benefit of the RPi3 is its ability to integrate and interact with almost any modern (and dated) IT system. Anything from mobile phones to supercomputers are able to communicate with the RPi3 and exchange information, hence making them the perfect device to implement a machine learning algorithm with, each capturing and analysing its own data and sharing amongst a network of connected devices, however this would be a best case scenario, and design of this system was focused on field use in sub-optimal conditions.

The algorithm was designed in the user friendly MATLAB language and from there it was piece by piece translated into Python, an ARM native, widely used, and open source programming language [11]. However, the structure was altered slightly to be more modular; the python variant took on an object oriented approach and allowed for the swapping in and out of vital parts of the algorithm. This means that the algorithm can take on many unique challenges and the approach changed depending on the input image type.

2.5 Results

Firstly, several values should be defined (equations 1-5), so that an accurate statistical analysis can be carried out. These are sensitivity, specificity, accuracy, positive predictive value (PPV), and negative predictive value (NPV) [12-14]. These values are extremely important in understanding not only the efficacy of the algorithm but how it fairs against others. The standard that it is tested against is the Canny algorithm [15]. In order to understand how these will be calculated, some parameters must first be defined as well:

Table 1. Parameters for statistical analysis

	Item in sample is a blood cell	Item in sample is not a blood cell
Item in sample is segmented	True Positive (TP)	False Positive (FP)
Item in sample is not segmented	False Negative (FN)	True Negative (TN)

$$Sensitivity = {}^{TP}/_{TP + FN} \tag{1}$$

$$Specificity = {}^{TN}/_{TN + FP} \tag{2}$$

$$PPV = {}^{TP}/_{TP + FP} \tag{3}$$

$$NPV = {}^{TN}/_{TN + FN} \tag{4}$$

$$Accuracy = {}^{TP + TN}/_{TP + FP + FN + FN} \tag{5}$$

Using equations 1-5 on both algorithms the following values were recorded:

Table 2. Performance measures of both algorithms

	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)	Accuracy (%)
Proposed algorithm	90.53	84.93	98.49	45.29	90.06
Canny	89.41	9.29	91.52	7.42	82.71

Table 3. Performance measure parameters for both algorithms

	TP	TN	FP	FN
Proposed algorithm	2667	231	41	279
Canny	2634	25	244	312

Each algorithm was tested on the same sample set containing 2946 cells and 270 platelets and other artefacts, totaling 3216 objects, shown in table 3. The specificity on the Canny algorithm is very low, much to its detriment, the algorithm fell over when it when it came to picking up on some of the more troublesome cells but would segment some of the other background noise, including over staining and burst cells. The NPV value is also very low because canny had a very low number of true negatives and a very high number of false negatives. Most notably the proposed algorithm outperforms Canny in accuracy and sensitivity, two areas canny reaches high percentages.

3. Discussion

The results show that in all aspects the algorithm out performs a high performing, and very relevant algorithm. Due to the objective of setting out to be less complex, the original goal was to match the performance of the canny algorithm on true positives but instead succeeded in overtaking it, showing real promise.

Some of the marginal differences in performance measures that might make negligible difference in other industrial applications that might improve the ability for computer aided diagnosis by slim percentages can have an astounding effect in absolute numbers from the perspective of the individual.

As technology improves and processes get faster, complexity is often now overlooked as modern processors don't have the same overheads in regards to time as they used to, but neglecting this vital factor to speed can lead to clunky, needlessly long processes that (in the context of image processing) can sometimes be to the detriment to the outcome, as seen earlier in section 1.2.

The algorithm presented in this paper and in [10] has real world applications, and has shown resilience amongst difficult conditions whilst maintaining standards and furthering the original goal of efficiency. With a standalone device capable of running the algorithm under 30 seconds that can offer portability, accessibility, and with the ability to be run of almost any supply of at least 5V, the implementation can be considered to have met the goals set out.

Machine learning and neural networks have seen a huge rise in popularity, and convolutional neural networks (CNN) have shown incredible promise in the field of image recognition. Several options where reviewed, and ultimately deemed unsuitable or out of scope – this is because these types of models take extreme computing power and are not suitable for low powered devices where the requirement to run locally, often without an internet connection is required. The RPi3 is cheap, accessible, and modular. Given the right environment, it is the perfect candidate to run a machine learning approach, but conditions aren't always right, leading to the need to process locally. As a concept, the device is considered standalone, with external connection not required to run.

4. Conclusion

This paper proposes a novel edge detection technique comprising of a new type of histogram-based analysis. Alongside the fast and comprehensive edge tacking routine, this was deployed for a blood image specific procedure and has proven its ability to be modular, on an RPi3, a truly networkable and cheap device with an ARM processor. This is just one type of deployment for a standalone device, whereas the algorithm can be run from any desktop or mobile processor, including native compatibility with ARM based mobile phones.

The proposed research described in this paper provides a variably sensitive algorithm with greater ability to pick objects from even low quality blood images. The future work on this area focuses on the optimisation of the algorithm which currently plagued by the same issues that are in common with other edge detectors, including inability to distinguish between overlapping objects of similar or identical intensityty, and the possible integration with machine learning

References

[1] S Shun, S Yan, Y wang, Y Li, (2014) "Medical CT edge detection algorithm based on improved fuzzy clustering analysis" *Intelligent Systems and Engineering Applications*

[2] N V Hung, N T T Hien, P T Vinh, N T Thao, N T Dzung (2017) "A utilization of edge detection in a modified bicubic interpolation used for frame enhancement in camera-based traffic monitoring" *Information and Communications*

[3] WHO (2017) "Malaria", *http://www.who.int/mediacentre/factsheets/fs094/en/*

[4] A A Alshennawy, AA Aly (2009) "Edge Detection in Digital Images Using Fuzzy Logic Technique" *World Academy of Science, Engineering, and Technology, 52*

[5] A Essa (2016) "Edge detection techniques using fuzzy logic", *Signal Processing and integrated Networks*

[6] A Pizurica, W Philips, M Acheroy (2003) "A versatile wavelet domain noise filtration technique for medical imaging" ", *IEEE Transactions on Medical Imaging, Vol.22, No.3*

[7] G Qiu, (1996) "An improved recursive median filtering scheme for image processing", IEEE Transactions on image processing, vol.5, no.4

[8] A Karami, R Heylen, P Scheunders (2015) "Band-specific shearlet-based hyperspectral image noise reduction" *IEEE Transactions on Geoscience and Remote Sensing, vol.53, issue 9*

[9] R Verma, J Ali (2013) “A comparative study of various types of image noise and efficient removal techniques” *International Journal of Advanced Research in Computer Science and Sofrware Engineering, vol.3, No.10*

[10] S Bias, S Reni, I Kale, (2017) “A novel fuzzy logic inspired edge detection technique for analysis of malaria infected microscopic thin blood images” *IEEE Life sciences conference vol.1 no.1*

[11] Python Software Foundation (2018) “About” *https://www.python.org/about*

[12] L Newberg “Some useful statistics definitions” (Last accessed 15/07/2018) *Renssalaer Polytechnic Institute Department of Computer Science, http://www.cs.rpi.edu/~leen/misc-publications/SomeStatDefs.html*

[13] R Parikh, A Mathai, S Parikh, G C Sekhar, T Thomas (2008) “Understanding and using sensitivity, specificity, and predictive values” *India Journal of Ophthalmology, vol.56, no.1*

[14] A K Akonbeng, (2007) “Understanding diagnostic tests 1: sensitivity, specificity, and predicative values” *Acta Pædiatrica, vol.96, issue 3*

[15] J Canny (1986) “A computational approach to edge detection” *IEEE Transactions on Pattern Analysis and Machine Intelligence vol. pami8, no.6*