Hardware implementation of Modified Annular Ring Ratio for Blood Cell Detection in Thin Blood Smear Images

Stecowiat, A., Reni, S. and Kale, I.

This is the published version of the article Stecowiat, A., Reni, S. and Kale, I. (2018) Hardware implementation of Modified Annular Ring Ratio for Blood Cell Detection in Thin Blood Smear Images. Procedia Computer Science, 141, pp. 390-395. It is available at:

https://dx.doi.org/10.1016/j.procs.2018.10.185.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 141 (2018) 390–395

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)

Hardware implementation of Modified Annular Ring Ratio for Blood Cell Detection in Thin Blood Smear Images

Alexandre Stecowiat[a,b,*], Saumya Kareem Reni[b], Izzet Kale[b]

[a]*IPB ENSEIRB-MATMECA, Bordeaux School of Engineering, 1 Avenue du Dr Albert Schweitzer, 33400 Talence, France*
[b]*Applied DSP and VLSI Research Group, University of Westminster, 115 New Cavendish St, Fitzrovia, W1W6UW London, United Kingdom*

Abstract

Malaria continues to spread in endemic areas. This deadly disease is the subject of multiple researches in terms of patient diagnoses. Post treatment diagnoses are necessary to make sure that patients treated for malaria continue to be free from Plasmodium protozoan parasites. Fast and automated analyses are possible with image processing of blood cell samples. This paper proposes a modified version of an image processing algorithm named Annular Ring Ratio, which identifies and locates the blood cell present in the thin blood smear images, to make the algorithm amenable to efficient hardware implementation through the elimination of the costly division process. The Annular Ring Ratio process identifies circular shapes through the calculation of a ratio between two circular areas. With proper configuration it can detect cell and parasite positions leading to the identification of infected cells and further estimate the level of infections.

Keywords: Post treatment malaria diagnosis; Plasmodium parasites; Annular Ring Ratio method; Red Blood Cells; Cell count; FPGA

Nomenclature

ARR	Annular Ring Ratio
RBC	Red Blood Cell

* Corresponding author. Tel.: +33-(0)67-712-4806; fax: +44-(0)20-7911-5089.
E-mail address: astecowiat@enseirb-matmeca.fr

1877-0509
10.1016/j.procs.2018.10.185

1. Introduction

Malaria is one of the most dangerous tropical diseases in the world. It is mainly present in Africa, Asia and South America. Without being diagnosed and treated early it can be fatal. In 2016, they were an estimation of 216 million cases worldwide and it kills an estimation of 445 000 peoples according to the World Health Organization (WHO) [1].

This disease is carried by the female anopheles mosquitoes that act as the matrix. With a bite, these mosquitoes release Plasmodium falciparum parasites into the human body. These parasites mature in the liver, enter the bloodstream and start infecting RBCs. These parasites are growing and multiplying in RBCs leading to infected cells burst and release more parasites into the blood. To diagnose patients, manual microscopy is considered as "the gold standard". Current malaria diagnosis has been described by [2]. These techniques are time consuming for experts and sometimes inaccurate. Fast, accurate and automated analyses needs to detect the number of infected cells and the stage of infection. Researchers continue to invent new algorithms to outperform current analyses.

This research paper focuses on a novel technique called Annular Ring Ratio (ARR). It has been proposed by [3]. This method is robust, has low level of complexity in preprocessing of the image and is insensitive to non-uniform illumination and colour variations. It is configurable for blood cell sample of different sizes and provides around 95% of result accuracy. This research paper describes a modified version of ARR method which enables a flexible and power efficient hardware implementation of the algorithm by avoiding the costly division process. It contributes to the research in image processing and the creation of a standalone, portable, automated, accurate and low-cost, malaria, and other optical imaging based, diagnosis system.

2. Methodology

2.1. Preprocessing

Before using the ARR algorithm, the image needs to be refined with 2 processes. First the image will be converted from RGB to grayscale. This conversion will reduce unnecessary computation by using 8-bits for pixels values instead of 24. RGB to grayscale conversion is carried out using a weighted averaging technique by putting a weight on each channel and summing these values to assess grayscale value.

$$\forall_{x,y \in image} Grayscale(x,y) = RW.Red(x,y) + GW.Green(x,y) + BW.Blue(x,y) \tag{1}$$

Where, RW, GW and BW are Red, Green and Blue weights for optimal contrast. By calculating every possibility, the result shows that the best combination was to put all the weight on the Green channel and not use the Red and Blue channel values (RW=0,GW=1,BW=0).

The next step is to carry out morphological filtering in order to remove the unwanted noise and artefacts in the image. The filtering technique is the novel method proposed in [2] which performs a morphological dilation using a ring shaped structuring element followed by erosion using a disk shaped structuring element. It is based on the idea

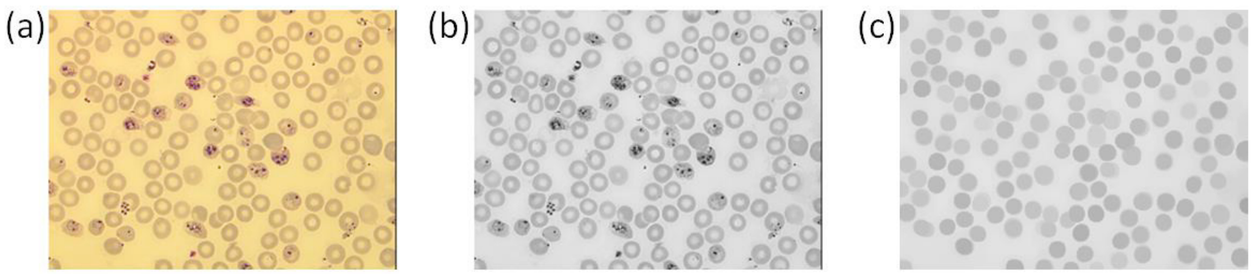

Fig. 1. (a) Original Image; (b) Grayscale Image; (c) Closed Image that will be processed by Annular Ring Ratio process

of morphological closing with structuring elements adapted to match the average size of a cell and the results are as shown in Fig 1.

2.2. Annular Ring Ratio Algorithm

This 2-part process is based on the morphology of cells and parasites. This paper will focus on cell detection but it can be adaptable to parasites by reducing radius values. In case it is used for parasites, the second pre-processing method won't be computed. The idea is to detect the contrast between the background and cells. This contrast can be detected by the differentiation of values in the grayscale spectrum.

2.2.1. Annular Ring Ratio Transform (ARRT)

For each coordinate on the image, it uses the shape of cells to determine 2 surrounding areas in which average grayscale colour will be computed. These areas are: a ring to detect background colour (in red in Fig. 2.a) and a circle to detect cell colour (in blue in Fig 2.a). These areas will be computed by the following equation for each pixel to create a new image.

$$\forall_{x,y \in image} ARRT(x,y) = \max(\frac{A_{Rxy}}{A_{Bxy}} - 1, 0) = \max(\frac{S_{Rxy}.N_{Bxy}}{N_{Rxy}.S_{Bxy}} - 1, 0) \qquad (2)$$

Where A_{Rxy} and A_{Bxy} are the average pixel value in red and blue areas surrounding (x,y) coordinates respectively. S_{Rxy}, S_{Bxy}, N_{Rxy} and N_{Bxy} are the summation of pixel values and number of pixels in red and blue areas respectively.

2.2.2. Annular Ring Ratio Find Peak (ARRFP)

This part is used to detect cell center in Annular Ring Ratio Transform images. For each coordinate it is searching if the pixel value is the local maximum in a circle surrounding area.

$$\forall_{x,y \in image} ARRFP(x,y) = \begin{cases} 1 & \forall_{i,j \in circlearea} ARRT(x,y) \geq ARRT(i,j) \\ 0 & else \end{cases} \qquad (3)$$

3. Main Hardware complexity

The ARRT method has been described on a Software platform (Matlab) and is 95% accurate. Hardware implementation of this operation is way more complex to realize with many constraints and considerations that must be taken into account.

3.1. Controller

This process uses neighborhood operations for each pixel to pick up surround pixel values in the image. In this case, a morphological operation architecture can be used. Researchers developed 4 different methods to optimize morphological operations in hardware: Partial Results Reuse [5], Delay line architecture [6], Systolic array

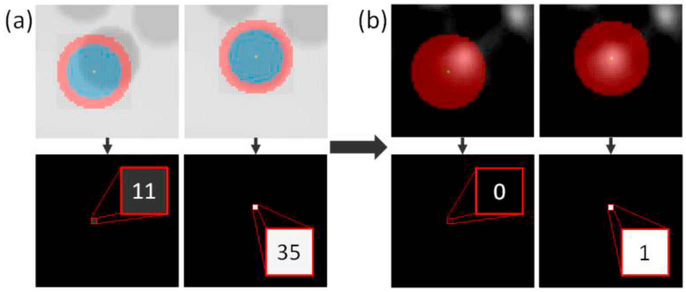

Fig. 2. (a) Annular Ring Ratio Transform: calculation of ratio between red and blue areas leading to the evaluation of cell center proximity. The more the structuring element matches a cell, the more the value is increasing. (b) Annular Ring Ratio Find Peak: evaluation if processed pixel is a local maxima for its surroundings.

architecture [7] and Pipelined architecture [8]. With these architectures, it will be possible to design an efficient controller that will be able to reuse pixel values, ratio estimation and local maximums.

3.2. Computation

In terms of computation, equation (2) is difficult to implement in hardware because it performs a division with variable sizes depending on cell sizes and pixel values. For example, a cell size radius of 12 pixels will require a division with a dividend and a divisor of 26-bits long. For a radius cell size of 50 pixels these values will grow to 34-bits. As a result, a divider adapted for a certain size of cell will be created and it will use a lot of resources. In this paper, the focus will be on the computational part and the efficient implementation of the process.

4. Proposed modifications and designs

Equation (2) has been described for software purpose. This complex division can be replaced with different methods to perform equivalent operation. These proposals have been designed in VHDL and synthetized in a FPGA.

4.1. Multiplication and subtraction

One solution would be to use a comparison and a subtraction between the dividend value and the divisor value as shown in Fig. 3.a. Equation (2) will be replaced by:

$$\forall_{x,y\in image} ARRT(x,y) = \begin{Bmatrix} S_{Rxy}.N_{Bxy} - N_{Rxy}.S_{Bxy} & if\,(S_{Rxy}.N_{Bxy} > N_{Rxy}.S_{Bxy}) \\ 0 & else \end{Bmatrix} \quad (4)$$

In this case, $S_{Rxy}.N_{Bxy}$ - $N_{Rxy}.S_{Bxy}$ will be stored in a memory (image). This solution has a good resolution but will need memory management because this subtraction can result in big values that will need subsequent storage.

4.2. The use of shifting

The solution to the issue in section 4.1 can be solved by using shifters. Shifters can help reduce the variable value in our equation. It can be used at different stage of the equation: on the summation of pixels, on the count of pixels, on the result of multiplication, and on the final result. Shifting would be a good and easy to implement solution for this application, however, discarding the bits thereby might lead to losing a lot of critical information. In addition, it is very sensitive to images taken under different acquisition set ups.

4.3. Mean architecture

Some researchers already described hardware architectures to perform an average. The architecture of [9] has been studied and evaluated for this process. This architecture computes an iterative average with an error result of 2% with a small amount of hardware resources that can be found in [9]. For the ARR process, two architectures like this one have been implemented to get the average value of the blue area and the red area. With these averages, it is possible

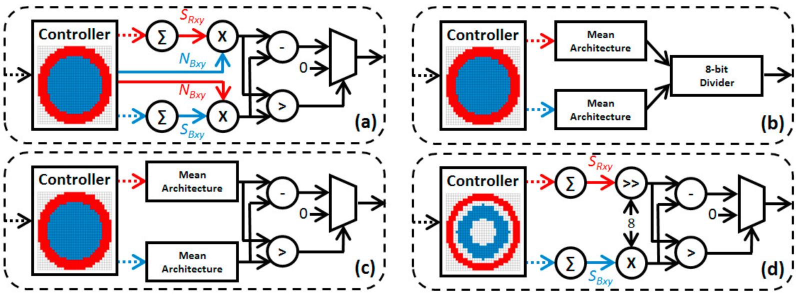

Fig. 3. (a) Division replaced by multiplication then subtraction (b) Use of mean architecture and 8-bit divider (c) Use of mean architecture and subtraction (d) Selective 256 values picked up by the controller, shifting by 8 to the right and subtraction

to use a division with 8-bits as shown in Fig. 3.b or a subtraction as shown in Fig. 3.c. A division will have a better resolution then a subtraction but will output values with a fractional part for the rest of the process.

4.4. Selective choice of pixels

The last solution would be to select a certain amount of pixel in our structuring element to compute the average using powers of two. It will sum every pixel and shift the result with a power of two as shown in Fig. 3.d. The advantage of this solution is that it is significantly simpler to implement, however, the difficulty is moved to the controller to pick up only a certain amount of pixels in the structuring element. This method has the disadvantage to be less relevant on the edge of the image when only a part of the structuring element is used.

5. Results

Table 1. Method specifications.

Method	Output data type (signed)	Theoretical Range	Resolution	Number of different values	Experimental Range (12 pixel cell radius)
(1) Multiplication and division	fraction	[0- $255.N_{Bxy}$]	$1/(255*N_{Rxy})$	$255^2.N_{Rxy}.N_{Bxy}$	[0 – 0.2498]
(2) Multiplication and subtraction	integer	[0 – $255.N_{Rxy}.N_{Bxy}$]	1	$255.N_{Rxy}.N_{Bxy}$	[0 – 6083704]
(3) Multiplication and shifting and subtraction	integer	[0 – $255.N_{Rxy}.N_{Bxy}/2^{shift}$]	1	$255.N_{Rxy}.N_{Bxy}/2^{shift}$	[0 – 185] (Shift = 15)
(4) Mean and division	fraction	[0 – 255]	0.00392	65025	[0 – 0.2609]
(5) Mean and subtraction	integer	[0 – 255]	1	255	[0 – 42]
(6) Selective Pick up	integer	[0 – 255]	1	255	[0 – 48]

Table.1 and Fig.4 shows the pros and cons of these methods. In terms of final results, methods (1) and (2) are the best while **method (3) provides the worst result**. A good resolution is provided on a huge range of values leading to very precise ratio and cell center identified on one pixel as can be seen in Fig. 4.a and b. For Hardware implementation, **method (1) must be avoided**. For memory purpose, **method (2) must be avoided**. The last solutions provide almost the same results. Then **method (5) is preferable to method (4) for hardware implement ability** while **method (6) is perfect to use less hardware when the controller is well designed**.

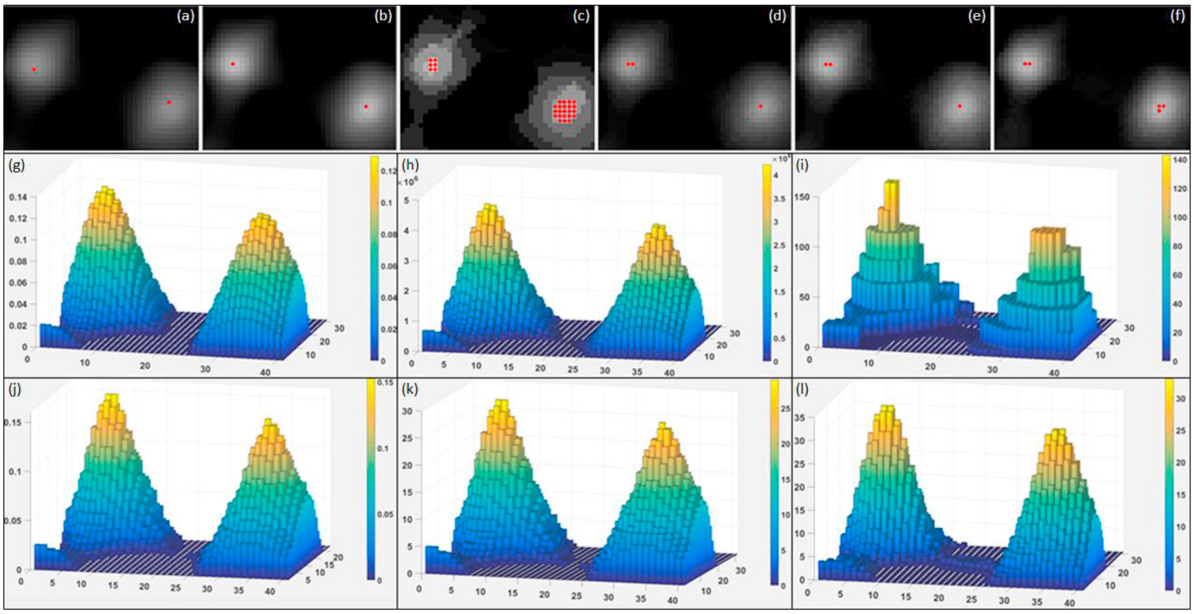

Fig. 4. (a,b,c,d,e,f) ARRT images from methods (1,2,3,4,5,6) respectively. Red dots on the images represent the output of ARRFP process. (g,h,i,j,k,l) shows value distributions in the image with methods (1,2,3,4,5,6). Methods (1) and (2) provide very precise distributions as can be seen in (g,h) leading to the best ARRFP results. Method (3) is much discretized in (i) causing the worst ARRFP result in (c). (4,5,6) shows good distributions in (j,k,l) but are not enough precise to get only 2 maximal values with ARRFP process in (d,e,f).