AUTOMATIC CELL COUNTING FROM MICROCHANNEL IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İZZET FATİH ERTÜRK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2022

Approval of the thesis:

AUTOMATIC CELL COUNTING FROM MICROCHANNEL IMAGES

submitted by **İZZET FATİH ERTÜRK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** ______________

Prof. Dr. Gözde Bozdağı Akar
Supervisor, **Electrical and Electronics Eng. Dept., METU** ______________

Prof. Dr. Haluk Külah
Co-supervisor, **Electrical and Electronics Eng. Dept., METU** ______________

Examining Committee Members:

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU ______________

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU ______________

Prof. Dr. Haluk Külah
Electrical and Electronics Engineering, METU ______________

Prof. Dr. Ziya Telatar
Electrical and Electronics Engineering, Ankara University ______________

Assoc. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz
Electrical and Electronics Engineering, METU ______________

Date: 09.02.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: İzzet Fatih Ertürk

Signature :

ABSTRACT

AUTOMATIC CELL COUNTING FROM MICROCHANNEL IMAGES

Ertürk, İzzet Fatih
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Gözde Bozdağı Akar
Co-Supervisor: Prof. Dr. Haluk Külah

February 2022, 98 pages

Cell counting is an important method used in many fields such as disease diagnosis in medicine. For example, the number of white blood cells in the blood gives information that there is a disease in the body. There are manual and automatic methods developed for cell counting. Although manual methods take a lot of time and are prone to error, they are preferred in many areas because they are less expensive than other methods. However, automated cell counting methods such as flow cytometry have been developed to achieve faster and more accurate results. The common feature of flow cytometry methods is that the fluid containing cells is passed through a microchannel one by one. These methods enable automatic cell counting as well as obtaining the physical properties of cells such as size. In cases where different cell types are examined together, cells can be classified by using the cell size and thus, the information about how many cells there are in each group can be obtained. For example, the type of disease can be determined by counting the types of white blood cells. The electrical impedance spectroscopy method, which is one of the flow cytometry methods, analyzes the effect of cell migration on the electric field applied to the microchannel, enabling the detection, counting and physical properties of cells. However, this method is quite expensive due to the devices that must be used for the method. In this thesis, it is aimed to develop an alternative method for counting the cells passed through the microchannel and classifying them according to their size. The flow of cells through the microchannel was recorded with a camera, and the images were processed with the proposed method, enabling the cells to be automatically detected, followed and counted, and classified according to their size. Image processing methods such as filtering, segmentation, opening and closing are used for automatic detection of cells, and Kalman filter and Hungarian assignment algorithm are used for tracking and counting. Classification according to their size was done with the K-medoids method and the ratio of different types of cells was found. The proposed method has been tested with various experiments and it has been concluded that cells can be detected correctly with image processing methods and cell counts can be successfully performed up to a certain flow rate.

Keywords: cell counting, object tracking, microchannel

ÖZ

MİKROKANAL GÖRÜNTÜLERİNDEN OTOMATİK HÜCRE SAYIMI

Ertürk, İzzet Fatih
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar
Ortak Tez Yöneticisi: Prof. Dr. Haluk Külah

Şubat 2022 , 98 sayfa

Hücre sayımı tıpta hastalık teşhisi gibi bir çok alanda kullanılan önemli bir yöntemdir. Örneğin kandaki akyuvar hücrelerinin sayısı vücutta hastalık bulunduğuna dair bilgi vermektedir. Hücre sayımı için geliştirilmiş manuel ve otomatik yöntemler bulunmaktadır. Manuel yöntemler, çok zaman almasına ve hataya açık olmasına rağmen diğer yöntemlerden daha az pahalı olması sebebiyle bir çok alanda tercih edilmektedir. Ancak, daha hızlı ve daha doğru sonuçlar elde etmek için akış sitometrisi gibi otomatik hücre sayma yöntemleri geliştirilmiştir. Akış sitometrisi yöntemlerinin ortak özelliği hücreler içeren sıvının bir mikrokanaldan tek tek geçirilmesidir. Bu yöntemler, otomatik hücre sayımının yapılmasının yanında hücrelerin boyutu gibi fiziksel özelliklerinin de elde edilmesini sağlamaktadır. Farklı hücre türlerin bir arada incelediği durumlarda hücre boyutu kullanılarak hücreler sınıflandırılabilir ve böylelikle hangi türden ne kadar hücre olduğuna dair bilgi elde edilebilir. Örneğin akyuvar hücrelerinin çeşitleri sayılarak hastalığın türü tespit edilebilmektedir. Akış sitometrisi yöntemlerinden biri olan elektriksel empedans spektroskopisi yöntemi ise hücre geçişinin, mikrokanala uygulanan elektrik alanına etkisini analiz ederek hücrelerin tespitini, sayılmasını ve fiziksel özelliklerinin elde edilmesini sağlar. Ancak yöntem için kullanılması gereken cihazlardan dolayı bu yöntem oldukça pahalıdır. Bu tez çalışmasında, mikrokanaldan geçirilen hücrelerin sayılması ve boyutlarına göre sınıflandırılması için alternatif bir yöntem geliştirilmesi amaçlanmıştır. Hücrelerin mikrokanaldan akışı bir kamera ile kaydedilmiş ve önerilen yöntem ile görüntüler işlenerek hücrelerin otomatik tespit edilmesi, takip edilerek sayılması, ve boyutlarına göre sınıflandırılması sağlanmıştır. Hücrelerin otomatik tespiti için filtreleme, segmentasyon, açma ve kapama gibi görüntü işleme yöntemleri, takip edilerek sayılmaları için ise Kalman filtre ve Hungarian atama algoritması kullanılmıştır. Boyutlarına göre sınıflandırma işlemi ise K-medoids yöntemi ile yapılmış ve farklı tip hücrelerin oranı bulunmuştur. Önerilen yöntem çeşitli deneyler ile test edilmiş ve görüntü işleme yöntemleri ile hücrelerin doğru şekilde tespit edilebildiği ve belirli bir akış hızına kadar hücre sayımının başarıyla yapılabildiği sonucuna ulaşılmıştır.

Anahtar Kelimeler: hücre sayımı, nesne takibi, mikrokanal

To my family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. Gözde Bozdağı Akar and my co-supervisor Prof. Dr. Haluk Külah for their guidence, advice, encouragement and endless patience throughout the completion of the thesis.

I am especially thankful to Alper Demir from Mikro Biyosistemler company for all his help, and endless patience in the laboratory.

I would like to express my gratitude to Prof. Dr. İlkay Ulusoy, Prof. Dr Ziya Telatar, and Assoc. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz for accepting to be a member of my thesis jury and evaluate my thesis.

I am also grateful to ASELSAN Inc. for the convenience it provided during my graduate education.

I have received scholarship from the Scientific and Technological Research Council of Turkey (TÜBİTAK) under 2210A domestic graduate scholarship program.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AD	Analog to Digital
CCD	Charge Coupled Devices
CDS	Correlated Double Sampling
CMOS	Complementary Metal Oxide Semiconductor
DAPI	4’,6-Diamidino-2-Phenylindole
EIS	Electrical Impedance Spectroscopy
FITC	Fluorescein Isothiocyanate
FPS	Frame per Second
PCB	Printed Circuit Board
sCMOS	Scientific Complementary Metal Oxide Semiconductor
TxR	Texas Red

CHAPTER 1

INTRODUCTION

Cell counting is widely used and very important for all cell-related research, including biological applications, genetics, biochemistry, medical diagnostics and treatments. Also, cell counts are important for monitoring cell health and proliferation rate, seeding cells for subsequent experiments, transfection or infection, and preparing them for cell-based experiments. It is important that cell counts be accurate, consistent, and fast, especially for quantitative measurements of cellular responses [11]. For example, in medical diagnosis, the amount of leukocytes, white blood cells, in the blood shows whether there exists a disease in the body. Neutrophils, the most common subtype of leukocytes, protect the body against a bacterial disease. Similarly, another subtype, the lymphocytes, protects the body against viral diseases. The source of the disease can be determined by looking at the ratio of these white blood cells in the blood. As a different example, biologists need cell counting to examine the time-dependent increase in the number of cells in the cell culture they use in bacterial experiments. In short, cell counting techniques have been developed and used for many purposes.

The first method developed especially for accurate cell counting is hemocytometers. As seen in Figure 1.1, thick glass chambers are divided into sections of calibrated area and depth. Generally, the size of the squares is 1mm x 1mm, and further subdivided into 0.05mm x 0.05mm squares. The chamber's edges are designed to hold a special glass coverslip 0.1 mm above the grid in order to create restricted areas of known volume [12]. A small sample of cell suspension is placed into the chamber, and the cells are counted manually by using a microscope. Then, the total number of cells in a suspension is easily calculated from the counts of cells in the hemocytometer chamber. This is a manual and the least expensive cell counting method. However, it is slow, and open to human errors, and therefore the least reliable method [13].

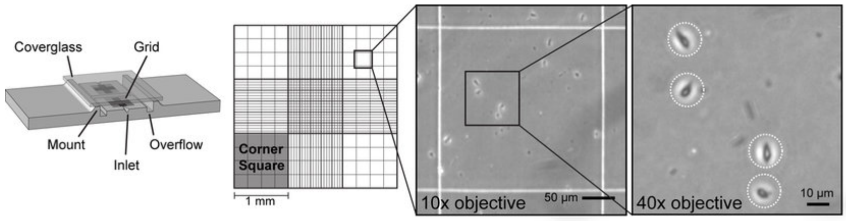
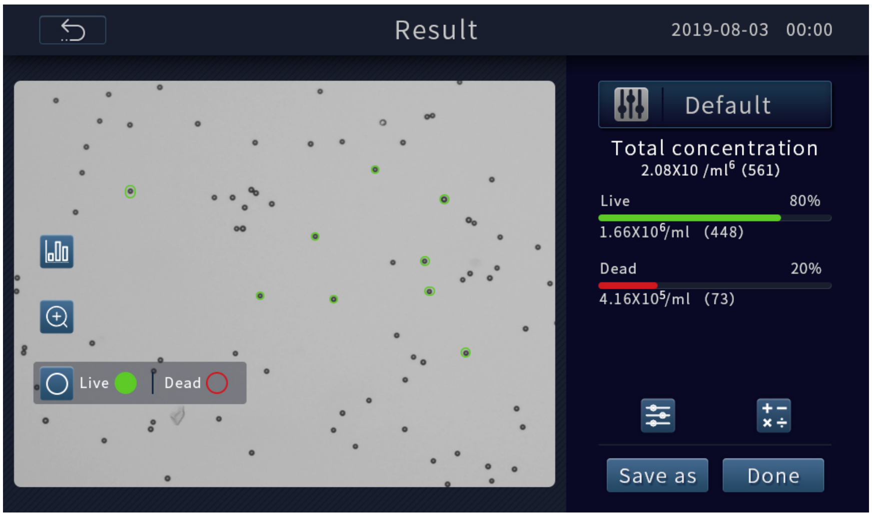


Figure 1.1: Hemocytometer cell counting method [1]

Automated cell counters are developed to provide faster, more accurate and detailed results. First method from the automated cell counters is image based automatic cell counting that has the same principle with hemocytometers, and uses the microscope image to count cells. It is designed to be faster and easier alternative to manual counting. Moreover, they can provide statistical information on the cell size depending on the camera properties. An example from an image based automated cell counter product screen is shown in Figure 1.2.



Result 2019-08-03 00:00

Default

Total concentration
2.08X10 /ml^6 (561)

Live 80%
1.66X10^6/ml (448)

Dead 20%
4.16X10^5/ml (73)

Figure 1.2: Image based automated cell counting device example [2]

There are same challenges for image processing. Firstly, the microscopic images are inherently noisy and contain optical artifacts. Moreover, image blurring and non-uniform illumination also make the cell counting more challenging [14]. Therefore, the used image processing techniques depends on the application. Generally, for a standard cell counting operation, filtering for noise reduction, segmentation of cells from the background by using global threshold, and morphological operations for separation of the cells can be used [15]. For the cases where cells are not completely visible, such as overlapping or cells in the process of dividing, some additional methods are required to accurately detect cells. For instance, adaptive histogram equalization can be used for image enhancement. In order to detect cells correctly, different methods such as edge detection, Hough transform or model based tracing can be used. For the overlapping cells, distance transform, similarity matrix generation by using template matching can be used for correct detection [14].

Cells generally have a round shape, and having a known shape has many conveniences for accurate cell detection. However, in some applications, the cells do not have a definite shape as shown in Figure 1.3. In these cases, in addition to the common image processing techniques, Laplacian of Gaussians, radial symmetry transform methods can be used for accurate separation and counting of the cells [3].

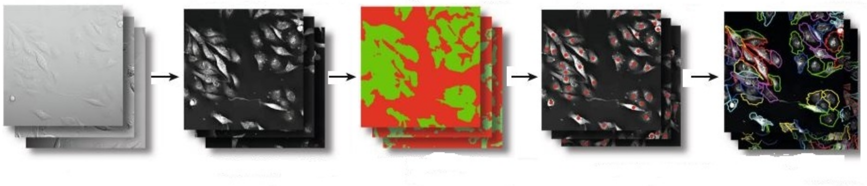


Figure 1.3: Segmentation of cells having non-uniform shapes [3]

Another type of automatic cell counters is Coulter counter method. The method is developed for fast counting and sizing of the cells, and based on the electrical impedance of the cells. It uses the principle that the electrical resistance of a conducting liquid is increased by the addition of an insulating material. The conductivity changes when a cell passes through the orifice. The resulting series of voltage pulses are electronically amplified, scaled and displayed on a suitable display. The magnitude of the voltage change is used to determine the volume of the cell. Figure 1.4 shows the schematic representation of the Coulter counter. A histogram plot is generated from the measurements, and different cell population can be divided by using an appropriate threshold.

Flow cytometers are most frequently used method for detailed cell counting and analysis, however, they are extremely expensive. In flow-based cell counting and analysis

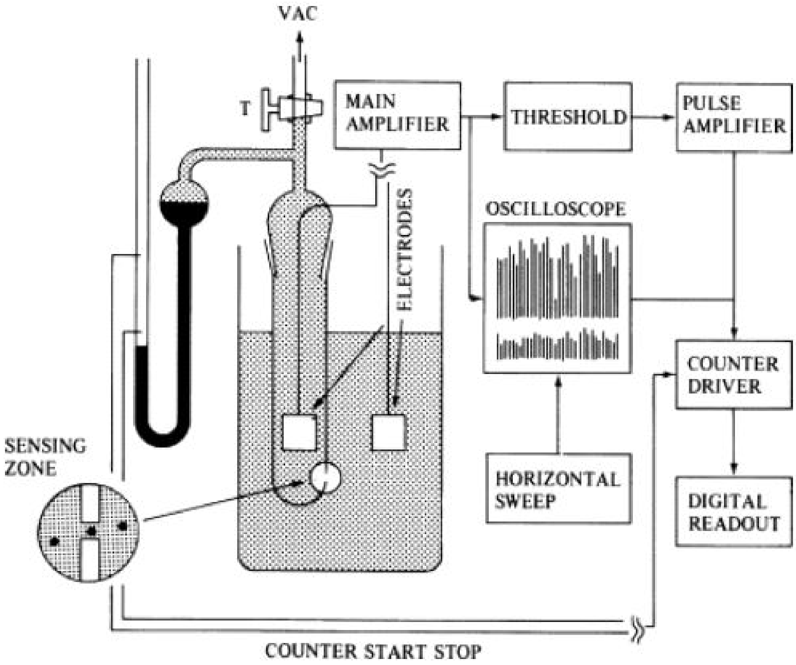


Figure 1.4: Schematic representation of Coulter counter [4]

methods, it is aimed to identify cells more easily by separating them from each other thanks to the microfluidic structure. One of the flow cytometry method is fluorescence activated cell sorting as shown in Figure 1.5. They use fluorescence detection technologies, and provide highly accurate cell counts with detailed properties. This makes them able to recognize cell types of the same size within the same sample [12]. In this method, while the liquid solution containing the cells is passed through the microchannel, the cells are exposed to the laser beam one by one, and each cell emits light. The fluorescence intensities of the inside and outside of the cell are measured with electronics light detectors. The change in the voltage value read from the detectors indicates the passage of the cell, and thus cells can be counted from these voltage changes. Moreover, it is possible to obtain more information about the cells. Forward scattered light is used to obtain biological characteristic such as size, and the magnitude of the forward scattered light is roughly proportional to the size of the cell. On the other hand, side scattered light that is caused by granularity and structural complexity inside the cell is used to measure internal structure of the cell [16]. Both measurements give a 1-D histogram for cells. In order to group the cells more effectively, two measurements are used to create a 2-D scatter plot, and therefore, grouping of the cells become easier.

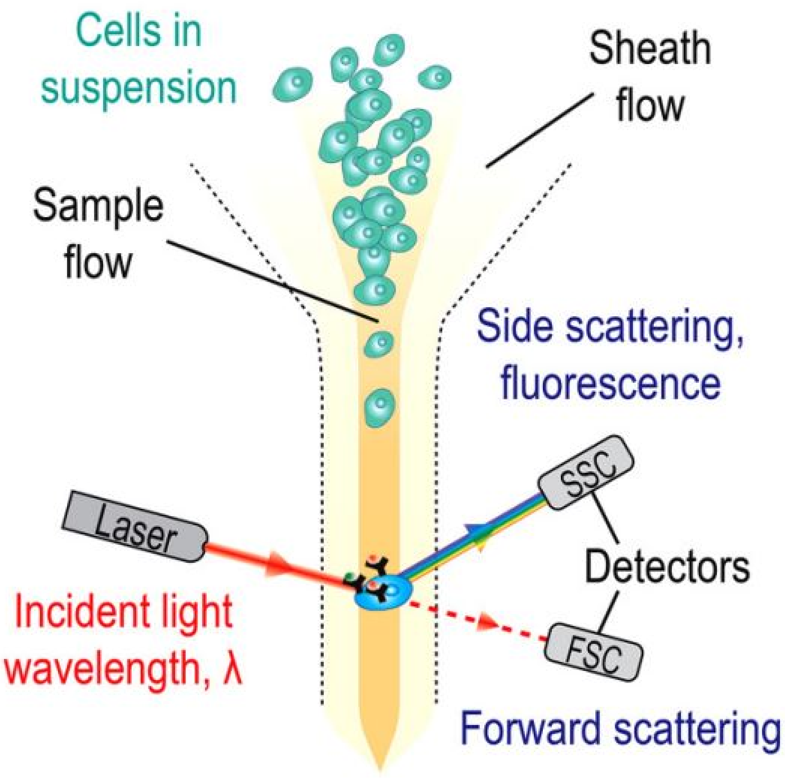


Figure 1.5: Fluorescence activated cell sorting method [5]

Another method of the flow cytometry is electrical impedance spectroscopy. A cell can be modelled electrically as in Coulter method when it is applied to an alternating current electric field as in Figure 1.6, and an electrical impedance equation can be obtained. Impedance causes phase and amplitude changes in the alternating current circuits.

In this method, as shown in Figure 1.7, while the solution containing the cells is passed through a micro-channel, an electric field is applied with the electrodes and the impedance of the cell creates a difference between the input and output signal at the electrodes. Figure 1.8 shows an example of the current change between electrodes when a cell passes. By detecting these changes, number of cells can be obtained. Moreover, some characteristics of the cell can be determined by examining the phase and amplitude changes.

Impedance measurements over a wide frequency range provide different information

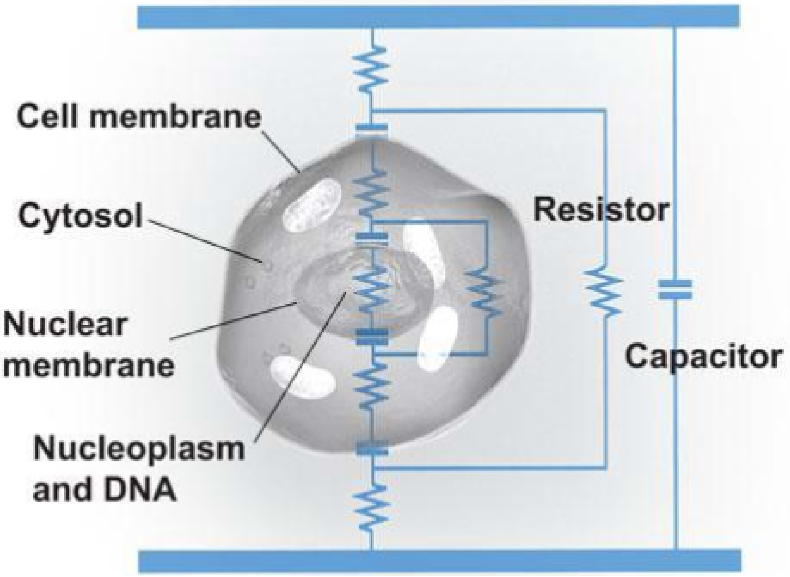


Figure 1.6: Electrical cell model [6]

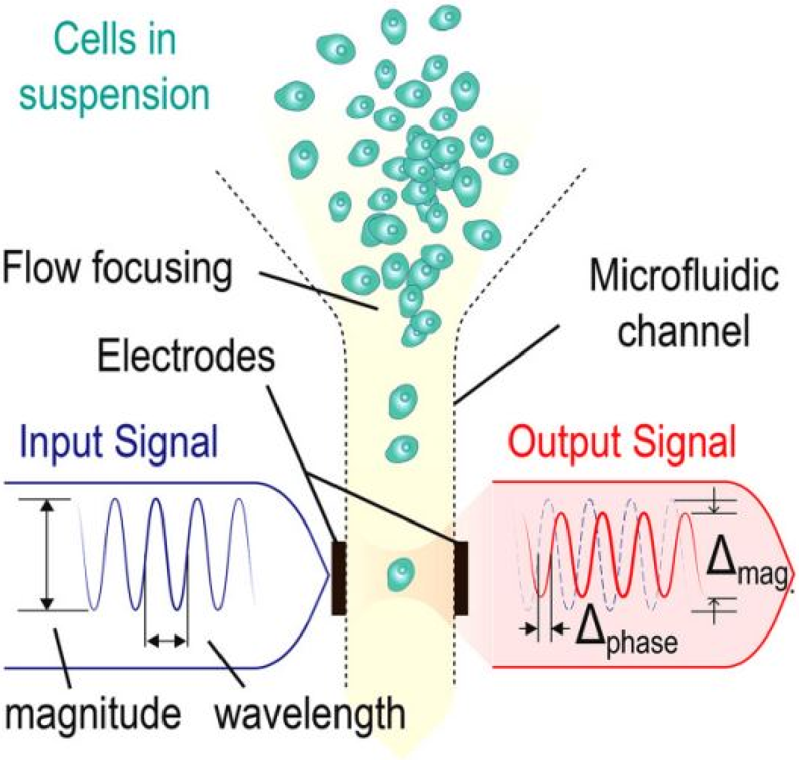


Figure 1.7: Electrical impedance spectroscopy method for cell counting [5]

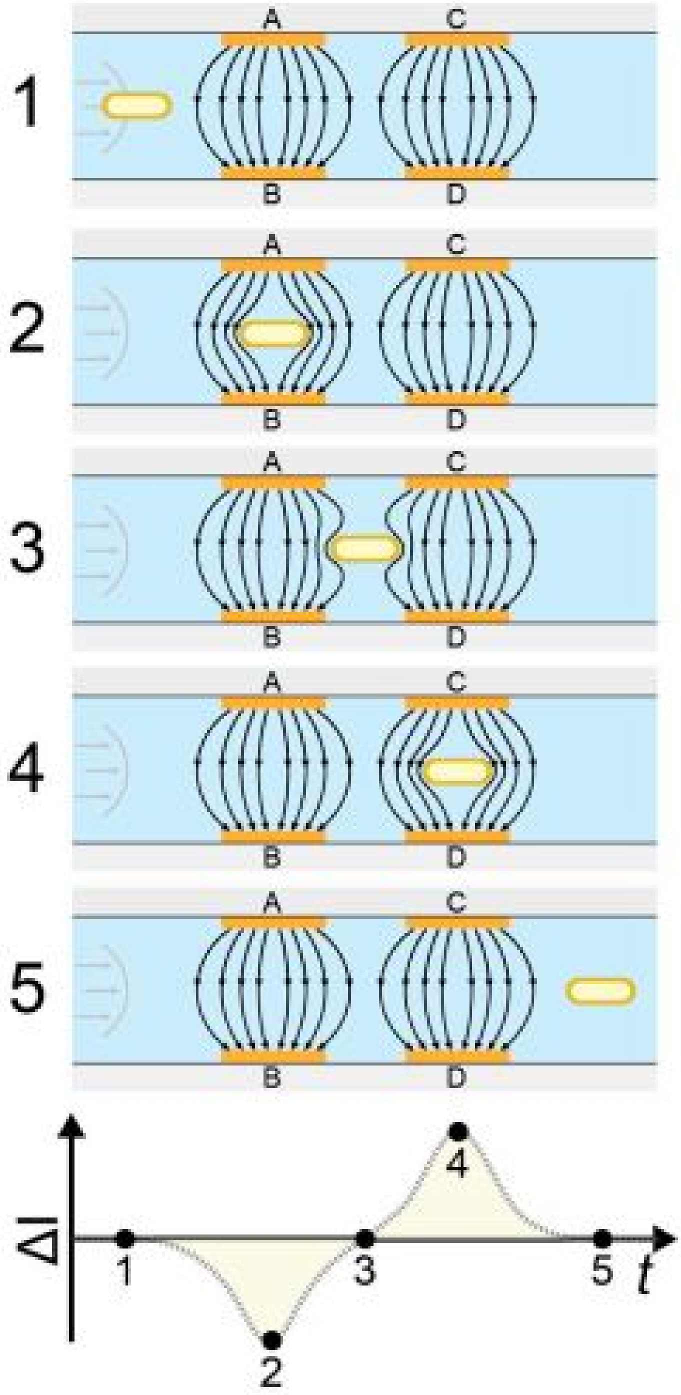


Figure 1.8: Current change example while a cell is passing between electrodes [7]

about cells such as cell size, membrane capacitance, and cytoplasm conductivity as function of frequency. Obtaining different measurements, amplitude and phase, at different frequencies allows to obtain these features and divide the cells accurately.

For example, when low frequency signal is applied, measured amplitude is used to determine the cell size. When signal has intermediate frequencies, measurements give information about the plasma membrane properties. At high frequencies, measurements give information about the dielectric properties of the cell interior [17]. Different information increases the dimension of cell properties and provides more accurate grouping of the cells.

Although flow cytometry methods make it easier to determine the number of cells from different groups by giving various information about the cells, they are extremely expensive. In this thesis, an effective and cheaper method that automatically detects, counts and classifies cells by size is proposed. The proposed method is based on image processing and object tracking techniques. Cells must be visible to use image processing methods. However, some cells, such as white blood cells, have no color. Therefore, they cannot reflect the light and cannot be seen easily. For these cases, staining of such cells in different colors is a frequently used method to see them under the microscope. By using this method, the cells can be visible under fluorescent light and recorded easily by the camera. The working system of the proposed method is as follows. As used in flow cytometry methods, the cell containing solution is passed through the microchannel. The microchannel is viewed with a microscope, and images are taken and saved by a high speed camera. The recorded images are processed with image processing techniques such as noise filtering, segmentation and morphological operations, and cells are detected automatically. A cell is displayed multiple times during the flow. In this way, instead of taking measurements at one location in the microchannel as in flow cytometer methods, multiple images are obtained for cell counting and analysis. Although flow cytometry methods try to ensure the passage of cells one by one, this is not always possible. Since detection and tracking is done along the microchannel, it enables more accurate counting of cells that move together and separate in a part of the microchannel. Automatically detected cells are tracked in the image sequence by using an object tracking algorithm so that they can be followed in all frames. Finally, measured cell sizes are classified by a clustering algorithm. Moreover, the speed of the cell is also measured from the images in order to create a control parameter for the flow rate. As a result, the number of cells passing through the microchannel, and the size of the cells can be determined by using image processing techniques and object tracking algorithm. The proposed method can be easily applied in many laboratories because the required devices are usually found in cell research laboratories.

This thesis contains 5 chapters. In Chapter 2, the background information about object detection and tracking methods is given. In Chapter 3 the experimental setup for image acquisition is described. In Chapter 4, proposed algorithms for automatic cell detection, tracking and size classification algorithms are constructed and explained with examples. In Chapter 5, the performance of the algorithm is tested with the recorded images by performing various experiments. Finally, in Chapter 6, the conclusion and future work of the thesis study is provided.

CHAPTER 2

BACKGROUND ON OBJECT DETECTION AND TRACKING

2.1 Introduction

In this chapter, background information for the proposed cell counting and analysis algorithm is given. The proposed cell counting and analysis algorithm is a kind of tracking algorithm that consists of detection, data association, and data classification parts. The aim of the algorithm is to find the number of cell passing through the microchannel, classify them according to sizes, and find the flow speed of the cells from the frames taken by the camera. The proposed algorithm contains three main topics such as detection, tracking and classification. This chapter starts with a general information about object tracking, and then the possible algorithms for each topics are provided.

2.1.1 Object Tracking

Object tracking is a popular topic nowadays, and it is used for many different purposes such as video surveillance, activity analysis, human-computer interaction, traffic control and robotics, motion-based pattern recognition, video indexing and vehicle navigation [18]. Depending on the application, one or more objects can be tracked. Generally, object tracking is composed of two main components: observation model and tracking process. The observation model is the descriptor of objects defining its appearance, velocity and location. Tracking is a process to find out the motion trajectory of an object while the video frames progresses along with the time, and identify the object position in every frame of the video [19]. There are some challenges such as pose variation, shape deformation, scale change and rotation of the object. In many instances, the target object may disappear [20]. These challenges are handled with different algorithms in the literature.

Since object tracking includes both observation model and tracking process, there are several object tracking methods containing different observation models or methods that are used for tracking process. In the case of objects that have a similar appearance and can be distinguished by small details, such as people or vehicles, it is important to obtain special features so that correct matching of the similar objects between different frames can be achieved. These features can be color, gradient, edges, texture, optical flow or biological features [19]. More detailed feature extraction can be done with convolution neural network or other deep learning methods [21]. Geometrical shapes are also an observation model and can be used for simple object tracking problems. As a summary, it is possible to find an object tracking algorithm for different needs in the literature.

Multiple object tracking algorithms can be divided into broad groups. One of the grouping method is initialization method, detection-based tracking and detection-free tracking. Detection-based tracking includes a detection part, and uses detected objects to link into trajectories. On the contrary, detection-free tracking requires manual initialization of a fixed number of objects in the first frame, and then localizes these objects in subsequent frames [22]. An alternative way to group tracking algorithms is grouping according to processing mode, online and offline tracking. Online tracking is also called as sequential tracking. This method uses the only the past information available up to current frame while offline tracking needs all frames during the motion, and therefore offline tracking is limited by the memory [22]. Another classification of multiple object tracking divides algorithms into two, deterministic and probabilistic. Deterministic method means that there is no randomness in the system. As a consequence, this method is problematic for complex movements, velocity change and noisy measurements. Other one is the statistical method that includes uncertainty factors such as noise in measurements and random complication in the motion. Therefore, it is more robust model, but it is more complicated [23]. Motion model is an important parameter in multiple object tracking algorithms, and it is used for estimation of possible position of an object in the future frames. There are two groups as linear and non-linear. Linear motion model includes constant velocity or constant acceleration motion models. If the motion of an object cannot be described with a linear equation, it is called as non-linear motion [22].

2.2 Image Processing Algorithms for Object Detection

In many cases, objects need to be detected that means they need to be separated from the background. For this purpose, many different methods have been developed according to the problem and the environment. In this section, only a few algorithms that are generally used in object detection are briefly explained.

2.2.1 Image Noise and Denoising

CMOS image sensors suffer from different noise sources such as thermal noise, AD conversion noise, pixel noise, amplifier noise and readout noise[24]. The noisy parts prevent to obtain more information from the image. Therefore, they must be reduced or eliminated. Filtering is a well-known technique for noise reduction and image enhancing. Although main purpose of this filters is noise reduction, these filters can be used to emphasize or remove some features in the image. There are several methods to remove noise in image, and they are explained briefly in the following sections.

2.2.1.1 Mean Filter

One of the simple linear filter is the arithmetic mean value or average filter. It takes the average of pixels in a region with determined size [25]. It removes short tailed noise and smooths out the impulses in the image at the cost of blurring image. The size of the region to be averaged is important and have significant effect on the blurring [26]. Since each pixel has equal weight, implementation of mean filter is easier, and needs less computation time than other methods.

2.2.1.2 Gaussian Filter

Gaussian filter is a linear and non-uniform low pass filter that is similar to mean filter, but it uses a different kernel. It is used to reduce noise, and the outputs are blurry images. A Gaussian filter is defined as in the Equation 2.1, and the only parameter that control the filter is variance, σ^2. Mean of the filter is assumed to be 0.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

Gaussian filter is more effective in smoothing images, and it is basically used in human visual perception system [26]. Although weights are different for pixels, the filter has symmetrical coefficients and this makes it easy to apply.

2.2.1.3 Median Filter

Median filter is a non-linear digital filtering method to remove noise from the image. It is widely used since it preserves the edges while removing the noise, and it is really effective at removing salt and pepper type noise [26],[25]. It takes the pixels in a region with determined size, sorts them in order from smallest to largest or largest to smallest, and picks the middle value called as median value as the output. Implementation of median filter is not simple due to sorting in each step, especially for larger filter size that causes high computational cost.

2.2.1.4 Wiener Filter

Wiener filter is an important technique to remove distortions especially for the distortions originated from linear motion or unfocused optics [26] [27]. It also called as inverse filtering and aims to reduce mean square error. It is very sensitive the additive noise and works in frequency domain. It generally outputs blurry image and computational cost is larger due to working in frequency domain.

2.2.2 Background Modelling

Background modelling is very important part for the obtaining foreground objects, and has many application area such as intelligent visual surveillance [28], intelligent visual observation of animals and insects [29], optical motion capture, human machine interaction, and content based video coding [30]. There are different methods for different requirements such as detection of moving objects from stationary background or detection of a material on a table. The background modelling methods for foreground moving object detection generally use similar algorithm. A background model is generated from first frames, and then current frame is compared with modelled background. Finally, algorithm updates the background [31].

Several difficulties can be encountered in background modelling, and four main challenges are explained shortly. The first challenge is the illumination change that has significant effect of the background model especially in outdoor scenes [32]. A good background model is expected to be robust against gradual or sudden changes of the light in the environment. Dynamic background is another challenge in background modelling. That means the background contains some movement, it does not contain only static objects. An excellent background model should effectively identify irregular movement of the object [33]. The other challenge is the shadows of the foreground objects that moves together with the objects. They needed to be separated from foreground objects. The final challenge is video noise. That challenge does not contain only the sensor noise, but also includes camera shaking, lens aging and other things as in [31][32].

There are many background modeling methods developed for different situations. These methods can be classified in the following categories [33].

- **Basic Background Modeling:** This group includes the background modelling by using average, median or histogram analysis [34].
- **Statistical Background Modeling:** This group contains background models that is modelled by a single Gaussian or a mixture of Gaussians or a Kernel Density Estimation. Statistical variables are used to classification of pixels.
- **Fuzzy Background Modeling:** This group is dedicated to the critical situ-

ations generating imprecision and uncertainties in the background modelling [32]. The background is modeled using a fuzzy running average or Type-2 fuzzy mixture of Gaussians.

- **Neural Network Background Modeling:** This group includes the background modelling methods that represent the background by mean of the weights of a neural network trained on clean frames. The neural network learns the classification.
- **Wavelet Background Modeling:** This group makes the background modelling by using dicrete wavelet transform in the temporal domain.
- **Background Clustering:** This group uses 2 class clustering methods to classify pixels. K-mean algorithm or Codebook methods are in this group.
- **Background Estimation:** A filter such as Kalman or Wiener or Tchebychev is used for background modelling. If the pixel value deviates significantly from the estimation, it is declared as foreground.

2.2.3 Segmentation

Segmentation can be described as process of partitioning an image into component regions or objects. There are different segmentation techniques for different cases. Segmentation methods can be classified as follows [35]:

- **Region Based:** A region is a pixel group having similar properties and region based segmentation divide image into regions. This technique is generally better for noisy images.
- **Edge Based:** Edges are found by analyzing discontinuities in the pixel intensities. This method uses edges to find the objects.
- **Threshold:** Thresholding method is a simple, fast and powerful technique for image segmentation. The separation of objects from the background is done by selecting a value T. There are two threshold methods, global and local. In the case of global thresholding, T value is constant. On the other hand, multiple

threshold values are used in local thresholding. A pixel is segmented as object if the pixel intensities are larger than the threshold, T [36].

- **Feature Based Clustering:** This method is more complicated and it uses features of the objects. It is the process of grouping objects that have similar properties. There are different algorithms for feature extraction and clustering such as K-mean and Fuzzy C-Means.

2.2.4 Morphological Operations

Morphological image processing is a nonlinear image processing technique used for filtering, enhancement and feature extraction [37]. It is a shape based algorithm especially for binary and grayscale images [38]. There are some imperfections due to segmentation. Some pixels between the background and the objects are labeled as a part of the objects. Moreover, there can be small white pixel regions in the image that can be called as noise. In order to remove these imperfections and noisy parts, morphological image processing techniques is used [39]. There is a structuring element with a desired shape, size and origin. Image is scanned with this structuring element, and mathematical operations are performed according to morphological operation. The structuring element is the same as convolution mask in linear image filtering. Dilation and erosion are basic morphological operations, and build many algorithms in morphological image processing. In a spatial domain over the basis n_1 and n_2, let A be the image, B be a structuring element. Dilation operation satisfies the Equation 2.2, and erosion operation satisfies the Equation 2.3.

$$A \oplus B = \begin{cases} 1, & \text{for } (n_1, n_2) \text{ where } A(n_1, n_2) * B(n_1, n_2) \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

$$A \ominus B = \begin{cases} 1, & \text{for } (n_1, n_2) \text{ where } A(n_1, n_2) * B(-n_1, -n_2) = \sum_{n_1,n_2} B(n_1, n_2) \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

Dilation operation increases the size of the object by adding pixels to the boundary elements. It can be said that it is analogous to smoothing spatial low pass filter in linear filtering. Erosion operation is the complement of the dilation operation, and decreases the object size from boundary elements. Since erosion operation removes structures lesser in size than the structure element, it can be used to remove noisy connection between objects. It is analogous to sharpening high pass filter in linear filtering [8]. Figure 2.1 and 2.2 shows the visual examples for dilation and erosion operations with a rectangular structure element, respectively.

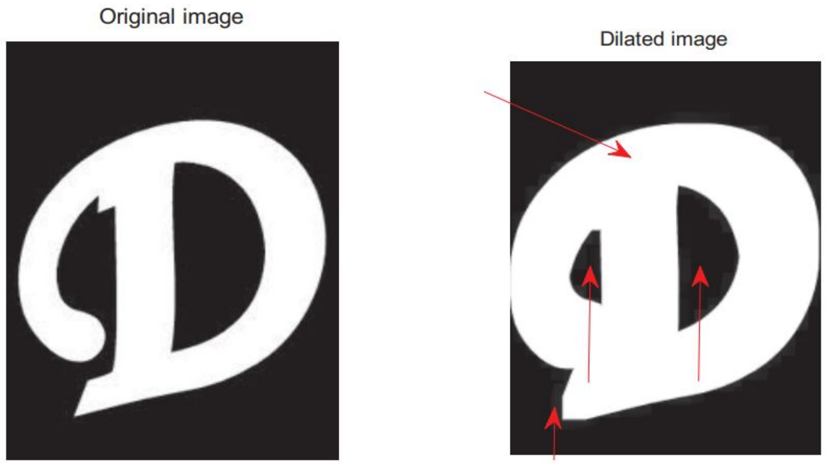


Figure 2.1: An example for dilation operation [8]

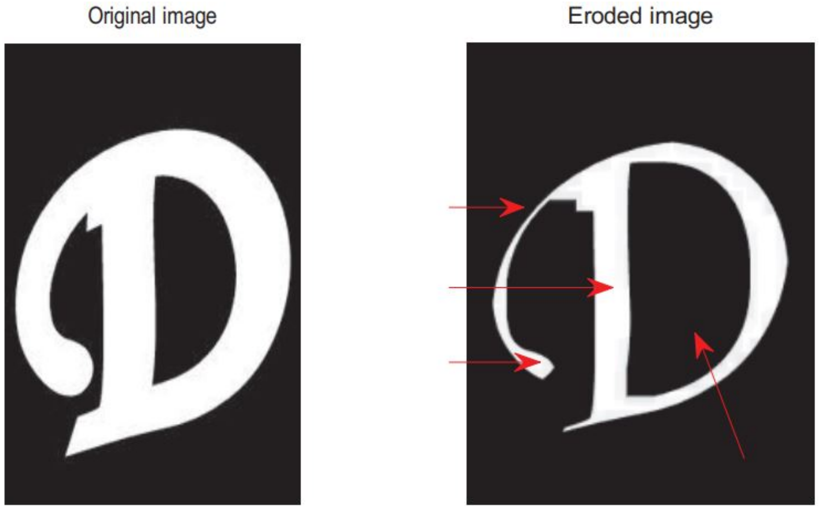


Figure 2.2: An example for erosion operation [8]

There are secondary operations in morphological image processing such as opening and closing. Opening is an algorithm that is useful for smoothing the contour, eliminating narrow protrusions, and breaking narrow passages without changing the size of objects too much. It is also useful for removing of unwanted noise, small regions in the image [39]. It performs first erosion, and then dilation, $A \circ B = (A \ominus B) \oplus B$. Likewise, closing is another algorithm that is useful for smoothing sections of contour, elimination small holes, and filling the gaps in that contour. It performs first dilation, and then erosion that are the basic morphological operations, $A \bullet B = (A \oplus B) \ominus B$. Figure 2.3 and 2.4 shows the visual examples for opening and closing operations with a rectangular structure element, respectively.

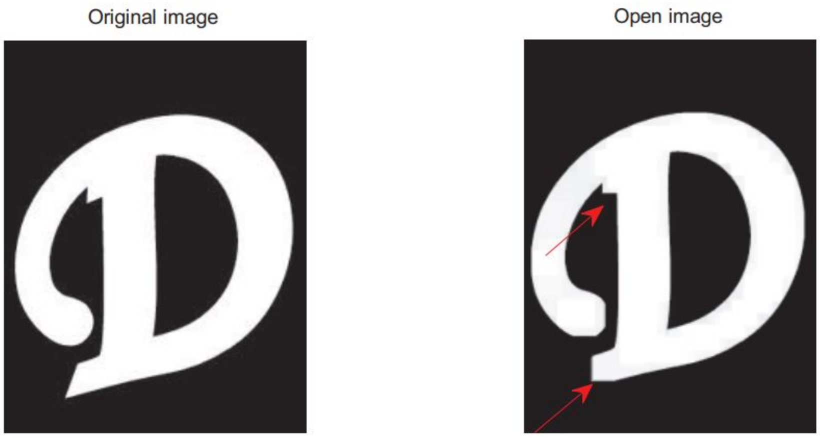


Figure 2.3: An example for opening operation [8]

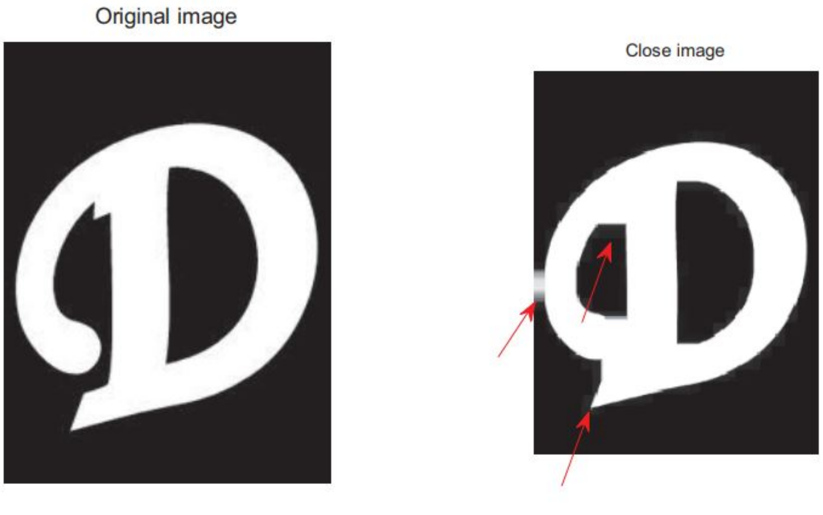


Figure 2.4: An example for closing operation [8]

2.2.5 Connected-Component Labeling

Connected component labeling algorithm aims to find and label connected pixel regions in the binary image with center of mass and size information. For pixel $b(x, y)$, the four pixels $b(x-1, y), b(x, y-1), b(x+1, y), b(x, y+1)$ are called the 4-neighbors of the pixel, and the four-neighbours together with the $b(x-1, y-1), b(x+1, y-1), b(x-1, y+1)$, and $b(x+1, y+1)$ are called the 8-neighbours of the pixel. Two pixels p and q are called as 4-connected or 8-connected if there is a path that contains 4 neighbours or 8 neighbours between p and q [40]. Figure 2.5 illustrates these concepts. Center pixel is shown in orange, and its neighbors are colored by light yellow. For 4-connectivity, only top, bottom, left and right pixels are counted as connected. For 8-connectivity, all pixels around the center are counted as connected.

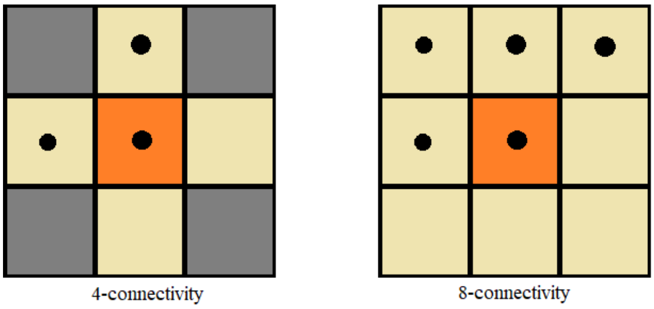


Figure 2.5: 4-connectivity and 8-connectivity

There are different connected component labeling algorithms that can be grouped in two parts. Firstly, the algorithms based on the label propagation search an unlabeled object pixel, and label it with a new label firstly. In later processing, they propagate the same label to connected pixels [40]. There are different search methods for different algorithms such as counter tracing labeling and hybrid object labeling. Secondly, algorithms based on label equivalent resolving process a binary image in raster scan in two steps. In the first step, they assign a label to each object pixel. In the second step, they integrate the equivalent labels [40]. There are many algorithms for label equivalent resolving method such as multi-scan, four-scan and two-scan algorithms. Two scan algorithm with 8-connectivity definition is used for connectivity check and labeling due to its easy implementation. Pixel with black dots are used in connectivity check operation. At the first step, image is scanned pixel by pixel from the top-left pixel to bottom-right pixel. Labeling algorithm is the following.

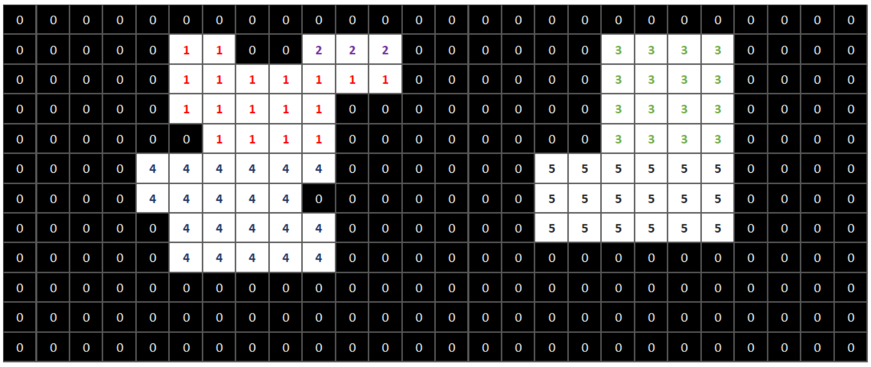
At the first pass,

- **Step 1:** Check the value of pixel.
- **Step 2:** If it is 0, it belongs to background and unlabeled, otherwise it belongs to object and need to be labeled.
- **Step 3:** Check left pixel whether it is labeled or not.
- **Step 4:** If it is labeled, assign the same label to current pixel, and go to the next pixel, Step 1. If it is not labeled, check top-left pixel.
- **Step 5:** If it is labeled, assign the same label to current pixel, and go to the next pixel, Step 1. If it is not labeled, check top pixel.
- **Step 6:** If it is labeled, assign the same label to current pixel, and go to the next pixel, Step 1. If it is not labeled, check top-right pixel.
- **Step 7:** If it is labeled, assign the same label to current pixel, and go to the next pixel, Step 1. If it is not labeled, assign a new label to the current pixel, and go to Step 1.

Figure 2.6 shows the result of first pass for an example image. Although there are 2 different connected regions, the first pass result in 5 different labeling. The second pass algorithm will solve this problem. This time, labeled image scanned pixel by pixel from the top-left pixel to bottom-right pixel. Labeling algorithm is the following.

At the second pass,

- **Step 1:** Check the pixel whether it is labeled or not.
- **Step 2:** If it is not labeled, it belongs to background, go to the next pixel, Step 1.
- **Step 3:** It it is labeled, check the top pixel whether it is labeled with a different label.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	2	2	2	0	0	0	0	0	0	3	3	3	3	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	3	3	3	3	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	3	3	3	3	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	3	3	3	3	0	0	0	0
0	0	0	0	4	4	4	4	4	4	0	0	0	0	0	0	5	5	5	5	5	5	0	0	0	0
0	0	0	0	4	4	4	4	4	0	0	0	0	0	0	0	5	5	5	5	5	5	0	0	0	0
0	0	0	0	0	4	4	4	4	4	0	0	0	0	0	0	5	5	5	5	5	5	0	0	0	0
0	0	0	0	0	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.6: Labeling example after the first pass

- **Step 4:** If it is not labeled with a different label, go to the next pixel, Step 1. If it is labeled with a different label, mark these two labels as equivalent. Go to the next pixel, Step 1.

After all pixels are scanned, equivalent label set is obtained as in Table 2.1. There are two different equivalent label sets and equivalent labels counted as a single label. At the end, 2 different connected regions are detected and labeled successfully. Labeling result after second pass is shown in Figure 2.7

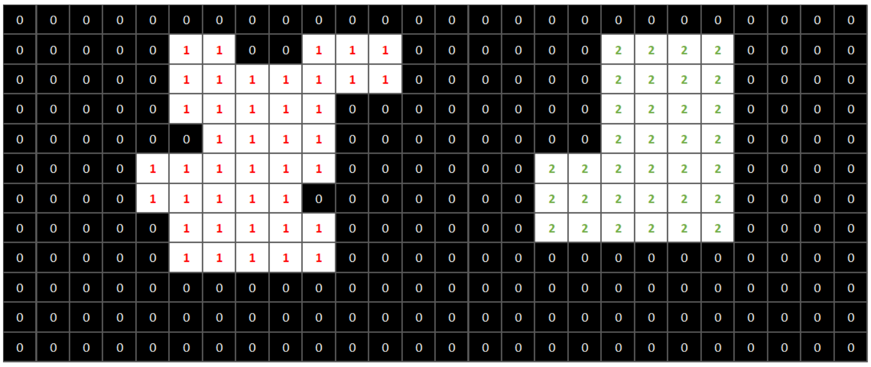
Table 2.1: Table of equivalent labels

Labels	Equivalents
1	1,2,4
2	1,2,4
3	3,5
4	1,2,4
5	3,5

Another necessary parameters are the center of mass and size information of the labeled regions. Centroid, center of mass, is calculated by using Equation 2.4.

$$Centroid_l(i,j) = \frac{1}{P_l} \sum_{i,j \in R_l} (i,j) \text{ for } l = 1, ...max\ label\ no \tag{2.4}$$

where P_l is total number of pixel in label l, and R_l is the region of label l.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	2	2	2	2	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	2	2	2	2	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	2	2	2	2	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	2	2	2	2	0	0	0	0
0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	2	2	2	2	2	2	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	2	2	2	2	2	2	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	2	2	2	2	2	2	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.7: Labeling example after the second pass

Width is the number of pixels of the object on the horizontal axis, and width of the labeled region can be found easily by $max(j) - min(j) + 1$ formula, where j is the column index in the labeled region. Likewise, length is the number of pixels of the object on the vertical axis, and the length of the labeled region can be found by $max(i) - min(i) + 1$ formula, where i is the row index in the labeled region. By using centroid, width and height information of labeled regions, a bounding box can be drawn to labeled objects.

2.3 Tracking Methods

The tracking part is the second part, and responsible for linking objects between detections from consecutive frames. It is used to follow an object during its motion. At this point, the motion model is an important parameter to link detections between consecutive frames. If the motion model is known, next location of the object can be estimated by using a mathematical model. When a new frame is taken, detections and estimations are compared, and they are linked if they are close enough to each other. However, measurements and motion model may not be exactly true, that means measurements can include noise, and time difference between frames may not be the same that causes an error in the mathematical motion model. Moreover, each object can move at different speeds. Another major challenge is that a detection become invisible for a few frames or when two detections move together and seen as a single object during a part of the movement. Due to these challenges, a tracker for each object must be used for making prediction of the next location of the object. This tracker must model the motion of objects between consecutive frames for next position estimations, and must be robust for speed changes, invisible frames, noise in the measurements and the processes. Since there are several algorithms in the literature, only suitable algorithms for the proposed method are described.

As stated, a tracker is necessary in order to make estimation for future location of an object, and it must be robust to different problems. Therefore, it must be probabilistic method. There are two well-known probabilistic tracking method, Kalman filter and Particle filter, and they use only the inforamation from past frames. Kalman filter can be used for linear or linearized processes and measurement system, and the uncertainty of Kalman filter is restricted to Gaussian distribution. On the other hand, Particle filter can be used for nonlinear systems, and deal with non-Gaussian noise distribution. Moreover, Particle filter consumes longer computation time [41]. In the case of a linear system with Gaussian-distributed object states, Kalman filter is proved to be optimal estimator [42]. As a result, Kalman filter is more appropriate method for proposed cell tracking algorithm since motion model of the cells in the microchannel is linear, and it can be considered as constant velocity motion due to constant flow rate.

2.3.1 Kalman Filter

Kalman filter is a statistical and region based tracking method, and appropriate for multi-target tracking especially small number of objects [20]. In the case of a linear system with Gaussian-distributed object states, Kalman filter is proved to be optimal estimator [42]. Mathematically, Kalman filter is an estimator to predict and correct the states of linear processes [43]. Kalman filter has a predict-correct model as shown in Figure 2.8. That means, Kalman filter firstly makes an estimation for the future frame, and then by using measurements, it updates itself, and make new prediction more accurately according to error between estimation and measurements including noises.

There are 4 sets of equations that form the basis of the Kalman filter. x_k is the state vector, and represents the dynamic behaviour of the object. Subscript k is the discrete time indicator. The aim is to estimate state vector from measurement vector, z_k. The

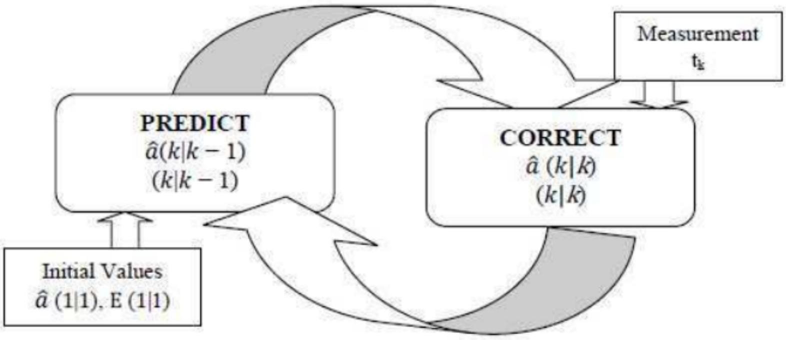


Figure 2.8: The Kalman Filter predict/correct model [9]

first equation is the process equation and given in Equation 2.5.

$$x_k = Ax_{k-1} + w_{k-1} \tag{2.5}$$

where A represents the state transition matrix, and w_{k-1} is the Gaussian process noise with normal probability distribution, 0 mean and variance Q. The second equation is the measurement equation that is given in Equation 2.6 .

$$z_k = Hx_k + v_k \tag{2.6}$$

where H is the measurement matrix and z_k is the observed measurement. v_k is the Gaussian measurement noise with normal probability distribution, 0 mean and variance R. Process and measurement equations describe linear model. The third equation set is time update equations given in Equations 2.7 and 2.8. State vector, x_k, is not directly measured, instead, measured z_k is used to update unknown states of x_k. Apriori estimate of state $\hat{x}_k^-$ and covariance error $\hat{x}_k$ estimate is obtained for the next time step k.

$$\hat{x}_k^- = A\hat{x}_{k-1} + w_k \tag{2.7}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.8}$$

where P is the initial state estimation error covariance matrix. Until now, given equations are related with the feedback part of the system. The final equation set is measurement update equations and given in Equations 2.9, 2.10 and 2.11. The objective is to estimate aposteriori estimation of x_k that is linear combination of the apriori estimate and the new measurement.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{2.9}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \tag{2.10}$$

$$P_k = (1 - K_k H) P_k^- \tag{2.11}$$

K_k is the Kalman filter gain and computed by Equation 2.9. After $\hat{x}_k$ and P_k is computed by the z_k, time and measurement equations are calculated recursively with previous aposterior estimates to predict new aprior estimate. The recursive behaviour of estimation is one of the highlights of the Kalman filter [44]. These equations are general Kalman filter equations. Vector and matrix sizes may change according to number of object to be tracked and motion model.

2.3.2 Assignment Algorithm

At many points in daily life, situations that require the assignment of available resources to needs may be encountered. For example, 4 postmen in the field have to deliver to 4 different addresses. To achieve an efficient result, all postmen should go to the nearest location and the overall distance traveled should be minimized. Similarly, detections need to be matched with the closest estimates in object tracking algorithms. One of the methods developed for this purpose is the Hungarian algorithm. Before making the assignments, distance parameters between all resources and needs must be determined. There can be different definitions for the distance between two points. A well-known distance type is Euclidean distance that represents the shortest distance between two points. In a 2-D plane, the Euclidean distance, d, between point $A = (p_1, p_2)$ and $B = (q_1, q_2)$ can be formulated as in Equation 2.12.

$$d = \sqrt{(p_1 - p_2)^2 + (q_1 - q_2)^2} \tag{2.12}$$

Manhattan Distance is the another distance type and formulated for the same points as in Equation 2.13.

$$d = |p_1 - q_1| + |p_2 - q_2| \tag{2.13}$$

Another distance formulation that is especially appropriate for Kalman filter applications is given in Equation 4.3. It takes into account the Euclidean distance between the points, and also includes the confidence of the prediction that is the covariance of the predicted state and the process noise of Kalman filter.

$$d(z) = (z - Hx)^T \Sigma^{-1} (z - Hx) + ln|\Sigma| \tag{2.14}$$

where $\Sigma = HPH^T + R$ and $|\Sigma|$ is the determinant of Σ.

After obtaining a distance matrix including distance between points, assignments must be done according to total minimum distance. Hungarian algorithm can be used for the assignment problem. Hungarian algorithm is explained with an example for the detections and tracker assignment case. Let n be the number of detection to be assigned trackers with the same number. It is not possible to assign a detection to multiple trackers, and a tracker to the multiple detections. There are nxn distance values, and $n!$ combinations for assignments. The following example shows how the Hungarian algorithm gives the minimum distance assignment for $n = 4$ [45]. Let A be the 4x4 cost matrix.

$$A = \begin{bmatrix} 80 & 40 & 50 & 46 \\ 40 & 70 & 20 & 25 \\ 30 & 10 & 20 & 30 \\ 35 & 20 & 25 & 30 \end{bmatrix}$$

- **Step 1:** Apply row reduction, subtract the smallest element in a row from all elements in that row.

Let B be the resultant matrix after Step 1.

$$B = \begin{bmatrix} 40 & 0 & 10 & 6 \\ 20 & 50 & 0 & 5 \\ 20 & 0 & 10 & 20 \\ 15 & 0 & 5 & 10 \end{bmatrix}$$

- **Step 2:** Apply column reduction, subtract the smallest element in a column from all elements in that column.

Let C be the resultant matrix after Step 2.

$$C = \begin{bmatrix} 25 & 0 & 10 & 1 \\ 5 & 50 & 0 & 0 \\ 5 & 0 & 10 & 15 \\ 0 & 0 & 5 & 5 \end{bmatrix}$$

- **Step 3:** Cover all 0s with minimum number of vertical and horizontal lines.

For C matrix, three lines, drawn to row 2,4 and column 2 covers all 0s.

- **Step 3:** If number of lines covering the all 0s are equal to row or column size, go to Step 5. Else, go to Step 4.

- **Step 5:** Start assignments by using 0s starting with the row that has minimum number of 0s.

Let D be the resultant matrix after Step 4.

$$D = \begin{bmatrix} 24 & 0 & 9 & 0 \\ 5 & 51 & 0 & 0 \\ 4 & 0 & 9 & 14 \\ 0 & 1 & 5 & 5 \end{bmatrix}$$

For D matrix, there are 4 lines to cover all 0s.

- **Step 5:** If the minimum number of lines covering all 0s is not equal to row or column size of the matrix, find the smallest element, s, in the matrix that is not covered by the lines. Subtract s from uncovered elements, and add s to the elements that has two lines. Go to Step 3.

When the algorithm continues with Step 3, assignments can be done as in $D*$ with minimum total cost.

$$D^* = \begin{bmatrix} 24 & 0 & 9 & 0^* \\ 5 & 51 & 0^* & 0 \\ 4 & 0^* & 9 & 14 \\ 0^* & 1 & 5 & 5 \end{bmatrix}$$

If the cost matrix is not a square matrix, a dummy zero row or column is added to the matrix so that the matrix become a square matrix. Then, the same algorithm is applied. However, no assignment is done by using that dummy row or column.

2.4 Classification of Unlabeled Samples

Classification or clustering is an unsupervised learning method to divide the data points into a number of groups so that data points in the same group are more similar to each other and less similar to those in the other group. Among the best-known classifiers are K-means and ISODATA [46],[47]. ISODATA means iterative self organizing data classification technique, and mostly used in remote sensing. It creates a predefined number of classes, and requires several parameters to control the number of clusters and the iterations. On the other hand, K-means is a statistical and data mining method. It divides n observation to k clusters based on the Euclidean mean value. This method is simple and fast if the number of clusters is known [46]. One of the main advantage of K-means algorithm is time complexity. Time complexity of K-means is $O(NKdT)$, where T is the number of iterations, and d is the number of features, K is the number of clusters, and N is the data count. Since K, d, and T are usually much less than N, the time complexity is approximately linear. Therefore, K-means is very desirable for large data sets [48]. Moreover, K-means and ISODATA have similar clustering performance while K-means requires lower computational time [49].

K-means algorithm finds a partition of n measurements into k clusters, and each measurement belongs to a cluster with the closest mean. K-means algorithm needs initial k value that specifies the number of clusters, and then make assignments to minimize the squared error [50]. K-means algorithm is explained step by step. Let $A = a_1, a_2, ..., a_n$ represents the set of data points and $C = c_1, c_2, ...c_k$ represent centers.

- **Step 1:** Specify k cluster centers randomly from the data set.
- **Step 2:** Compute the Euclidean distance between cluster centers and each data point.
- **Step 3:** Place the data into the cluster with least distance.
- **Step 4:** Recompute the new center, mean value, for all clusters.
- **Step 5:** Recompute the distance between each data and cluster centers.
- **Step 6:** If any data changes the cluster, redo from Step 3, else stop.

Since the number of clusters is predetermined, having a wrong measurement, that generates outlier, affects the classification significantly. That is, K-means algorithm is sensitive to outliers. As a solution, instead of using the mean value of the objects in a cluster as a reference point, medoids are used [51]. This algorithm is called as K-Medoids, and it is a good alternative to the K-mean algorithm. The partitioning method is performed based on the minimizing the sum of dissimilarities between each object and corresponding medoid. The dissimilarity is defined as absolute difference between an object and medoid. K-medoids algorithm is performed in following steps [52].

- **Step 1:** Specify k random points as medoids from data set a
- **Step 2:** Assign each object to the nearest medoid by using any distance metric methods.

- **Step 3:** Randomly select a non-medoid object.
- **Step 4:** Compute the total cost of swapping the medoid with a new random point.
- **Step 5:** If the cost is negative, use the new random point as the medoid, and repeat from step 2 until no change occurs.

CHAPTER 3

EXPERIMENTAL SETUP AND IMAGE ACQUISITION

3.1 Introduction

In this chapter, the experimental setup that is required for the automatic cell counting from microchannel images is demonstrated, and image acquisition is explained. In Section 3.2, the experimental setup is demonstrated with brief information about components. Since the imaging is an important part due to providing inputs to the proposed algorithm, some basic information about CMOS imaging is given initially, and then detailed information about the image acquisition with ORCA-Flash4.0 LT+ scientific CMOS camera is explained in Section 3.3.

3.2 Experimental Setup

A setup consisting of important components is needed for automatic cell tracking from microchannel images. For this purpose, the laboratory and apparatus currently used in the Mikro Biyosistemler Inc. are used. Mikro Biyosistemler company creates solutions for biomedical applications such as circulating tumor cells detection systems. They especially works on the micro electro mechanical systems and microfluidics technologies. One of the current projects carried out in this company is cell counting and analysis using the electrical impedance spectroscopy method that is explained briefly in Chapter 1. The tools used in Mikro Biyosistemler laboratory are those that can be easily found in the cell research laboratories, and no additional material was needed for the proposed method.

In the experiments, instead of real and live cells, floresbrite beads are used with different sizes and colors. They have similar electrical and physical characteristic with real cells, and they glows under fluorescent light. Therefore, they are visible for the cameras. These beads are inside the phosphate-buffered saline solution that has similar physical and electrical properties with the plasma where the cells live in the body. Therefore, the solution is suitable for cell culture, and used widely in biological experiments. Two types of beads modelling two types of cells are used in the experiments. The first bead type has 6.1μm diameter, and the second bead type has 9.8μm diameter. These beads are named as cells in the thesis.

The first and the main component of the setup is the microchip that contains the microchannel and electrodes. The microchannel is the channel through which the fluid containing the cells is passed, and has a cross-sectional area of 30μmx30μm with a length of 2mm. The electrodes are used to apply alternating current signals to be used impedance measurements in electrical impedance spectroscopy method. The lab-on-chip including the microchannels is shown in Figure 3.1.

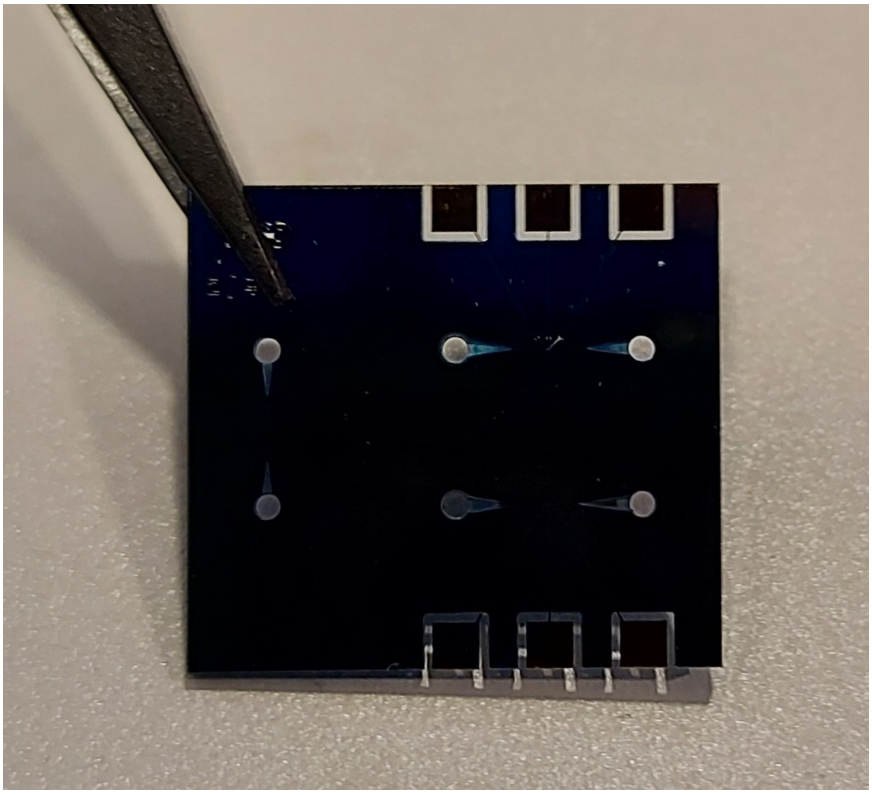


Figure 3.1: The microchip including the microchannels and electrodes

A chip holder is used to fix the microchip and make connections for electrical measurement and fluid flow. Figure 3.2 shows the chip holder with connection cables. The connection cable is used to electrical connection of the electrodes in the chip to the other necessary components.

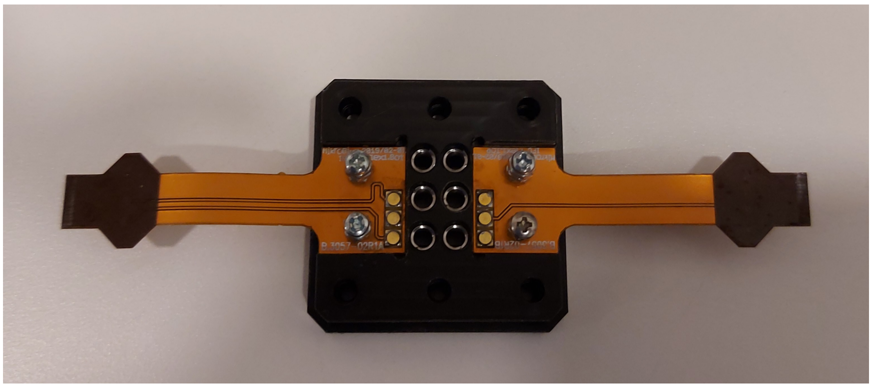


Figure 3.2: The chip holder with connection cables

The microchip that is inside the chip holder is connected to a PCB with the connection cable, and connections for fluid flow is done as shown in Figure 3.3.

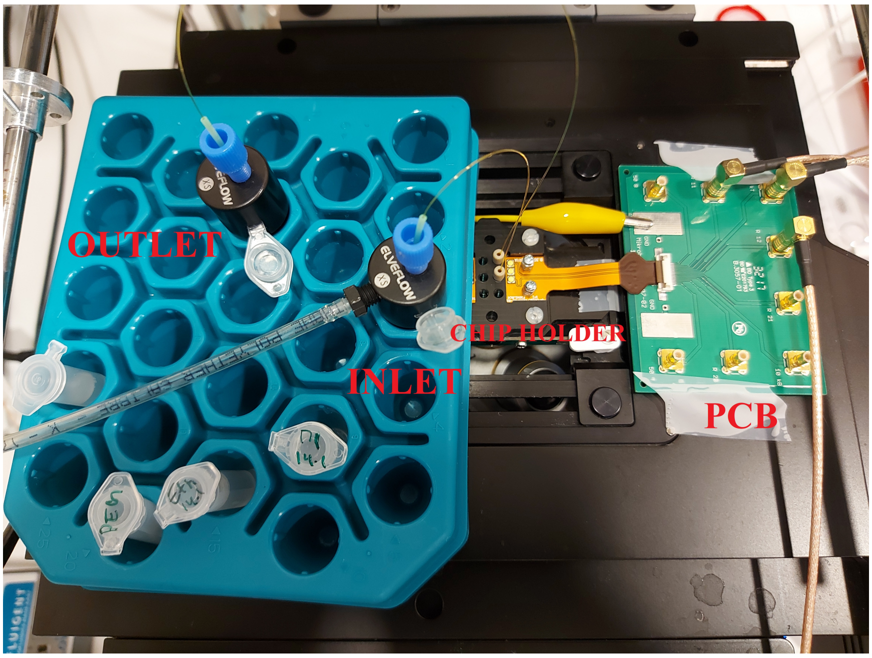


Figure 3.3: Combined setup elements on the microscope

PCB is used for extending the electrical lanes, and connecting them to the connectors for impedance measurement. There are two tubes connected to the microchannel. First one is the inlet, that includes the phosphate-buffered saline solution containing cells, and the other tube is the outlet tube where the solution passed through the microchannel is collected. The part including microchip, chip holder, PCB and tubes are fixed to the microscope for viewing the microchannel with the help of the microscope.

The flow of the liquid is provided by Flugient Flow EZ that is a liquid pressure generator and controller. The pressure is controlled by the computer software. This device is connected to inlet tube of the system. The solution in the inlet tube is pushed into the microchannel by the pressure generated by the flow generator. The connectors on the PCB is connected to the Zurich HF2IS that is a 50MHz Impedance Spectroscope. The impedance spectroscope generate signals at different frequencies, and this signals are applied to the microchannel initially. Then, it senses and measures the output voltage of the microchannel. By using phase and amplitude differences, an impedance is calculated and saved to the computer. The measured impedance values have peaks at some points that indicates the effect of the cell. By this way, number of cells and properties can be obtained. Although the impedance spectroscope is a key component for the electrical impedance spectroscopy, it is not used in the proposed method.

Leica Dmi 8 microscope is used to view the microchannel and the cells in 5x zoom mode. It has C-Mount adapter that provides external camera connection to the microscope. A fluorescent light source was also used to illuminate the microscope. There is a light filter set that filters the incoming fluorescent light. The filter set includes DAPI, FITC and TxR filters and any combination can be used. These filters are used to make different stained cells appear together or individually. In the experiments, all filters are used to make all types of cells are visible. Moreover, only visible things are the cells under the fluorescent light.

An ORCA-Flash4.0 LT+ scientific CMOS camera is mounted the microscope, and the images taken by the camera are transferred to the computer. The camera in the setup is only used to observe the microchannel to see the possible situations easily. The camera software is used for live view of the microchannel. Therefore, most of the function of the camera is not used, and no images are saved. However, the camera is the key component of the proposed method, and images are needed to be saved at high fps rate. For this reason, appropriate selections are made in the camera settings, the images are recorded in a special format, and then the images are read by the written MATLAB code. Camera functions and appropriate settings for the proposed method are explained in the Image Acquisition part. Note that, although the microscope and the camera is not used in electrical impedance spectroscopy method, they are necessary components used for the control and observation of the operation of the system. Figure 3.4 shows the setup with all components.

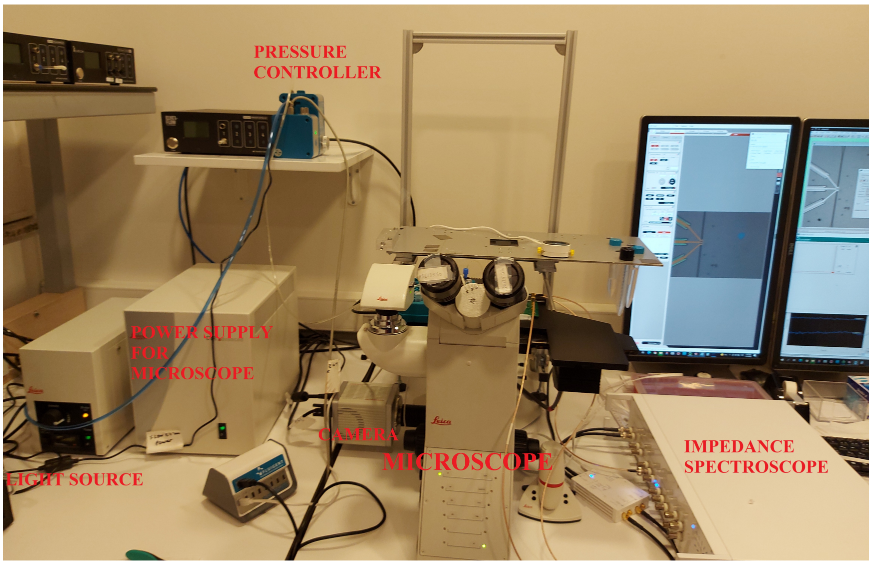


Figure 3.4: The experiment setup with all components

3.3 Image Acquisition

Image acquisition is the most important part of the algorithm since it provides inputs to the algorithm from a sensor, and any problem on this part causes failure in the algorithm. Therefore, it is important to understand the possible effects of the image acquisition part on the system. In the following section, some basic information is given about the CMOS camera sensor and imaging parameters, and then technical details of ORCA-Flash4.0 LT+ scientific CMOS camera, and appropriate settings for cell detection and tracking are provided.

3.3.1 CMOS Camera Basics

A camera is an optical device that is used for image capturing, and the sensor is the most important part of a camera. The sensor generates images from photons, to electrons, and finally to grey levels. In modern microscopy digital cameras, there are 2 main categories, CCD and sCMOS. CCD is the first digital camera and have been used for scientific imaging for decades [53]. It is well suited to high-light applications. However, it lacks sensitivity and speed. Readout of the pixels is done sequentially. Therefore, frame rates are limited. On the other hand, sCMOS cameras has different readout mechanism. Readout process is done in parallel. Therefore, sCMOS sensors operate in higher speed, and frame rates. Moreover, sCMOS cameras provide larger dynamic range meaning that there are more gray levels between black and white [54]. Both types can provide same pixel sizes. As a result, sCMOS camera is chosen for microscopy image capturing of moving cells because of lower power dissipation, higher frame rates, larger dynamic range and lower cost [55],[24].

Basically, a CMOS camera consists of an 2-D array of CMOS sensors, and each sensor includes a photo-diode and an amplifier. Incoming light is converted to electrical charge by photo-diode, and amplifiers convert charges to voltage. Each pixel is converted to a digital value by using CDS circuit and AD converter [24]. Then, readout process is done in parallel for columns. An example of CMOS image sensor is given in Figure 3.5. In this CMOS sensor type, each column has two CDS circuits and AD converter. Number and location of CDS circuit and AD converter may differ for different types of CMOS sensors. Figure 3.5 shows an example sensor similar to the sensor used in ORCA-Flash4.0 LT+ scientific CMOS camera.

Exposure time(shutter time) is the length of time that the photo-diode inside the digital camera is exposed to light in order to create a pixel value. Long exposure time provides more energy to create pixel value with less noise, however, that may cause undesired blurry results for moving objects.

Readout time is the necessary time to digitization and readout of pixel values in a frame. Readout time depends on the readout methodology that determines the reading order and algorithm of pixels.

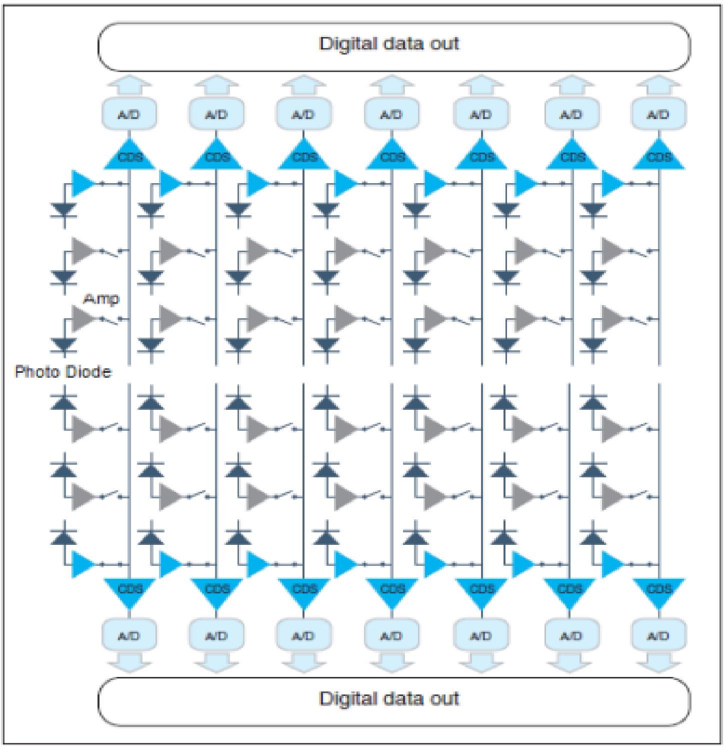


Figure 3.5: The structure of the CMOS Image Sensor [10]

Frame rate is a key parameter for cameras. It generally expressed as frame per second, fps, and represent number of consecutive frames captured by the camera in a second. Equation 3.1 is a general fps equation, however, fps value may be dominated by frame acquisition time or frame read time depending on the camera and parameter selection.

$$fps = \frac{1}{\text{Frame Acquisition Time} + \text{ Frame Read Time}} \tag{3.1}$$

where frame acquisition time is the total time including shutter opening, exposure, and shutter closing time, and dominated by the exposure time.

3.3.2 Image Acquisition with ORCA-Flash4.0 LT+ sCMOS Camera

There is a digital CMOS camera that is connected to the microscope in order to take images from the microchannel during the cell flowing. It is a ORCA-Flash4.0 LT+ scientific CMOS camera that is designed for microscope imaging by Hamamatsu Photonics Company. It is equipped with an advanced CMOS imaging sensor FL-400 that provides low noise and high resolution [10]. Table 3.1 shows some technical details of the camera.

Table 3.1: Technical Details of ORCA-Flash4.0 LT+ scientific CMOS camera

Effective area	13.312mm (H) x 13.312mm(V)
Effective number of pixels	2048(H)x2048(V)
Cell size	6.5μm x 6.5μm
Sensor readout time	33ms (Normal Mode)
Sensor readout time	10ms (Rapid Rolling Mode)
Exposure time	3ms to 10s (Normal Mode)
Exposure time	1ms to 10s (Rapid Rolling Mode)
Frame rate (2048x2048)	30 frames/s
AD conversion	16 bit
Interface	USB3.0

The FL-400 CMOS image sensor consist of 2048x2048 pixels that are composed of a photo-diode and an amplifier. Incoming light is converted to charge by the photodiode and then converted to voltage by the amplifier. The voltage that is converted in each pixel is output by the switch one by one. FL-400 imaging sensor has an on-chip CDS circuit that provides low noise. The output of the CDS circuit is sampled by a 16bit AD Converter, and digital data is obtained. There are two split readout schemes at the top and bottom halves of sensor to provide independent and fast readout [10].

The CMOS image sensor has rolling shutter exposure and readout method. In this mode, the exposure and readout are done by line by line instead of taking a snapshot. That means, parts of the images are not recorded exactly at the same time. However, this method is not a problem even if the objects are moving during the exposure since rolling shutter time is very small. Figure 3.6 shows exposure timing of rolling shutter method.

The camera reads out the CMOS image sensors from the center to the top and the bottom simultaneously. Figure 3.7 shows the readout method.

There are several readout methods such as Normal, Binning, Sub-array and Rapid

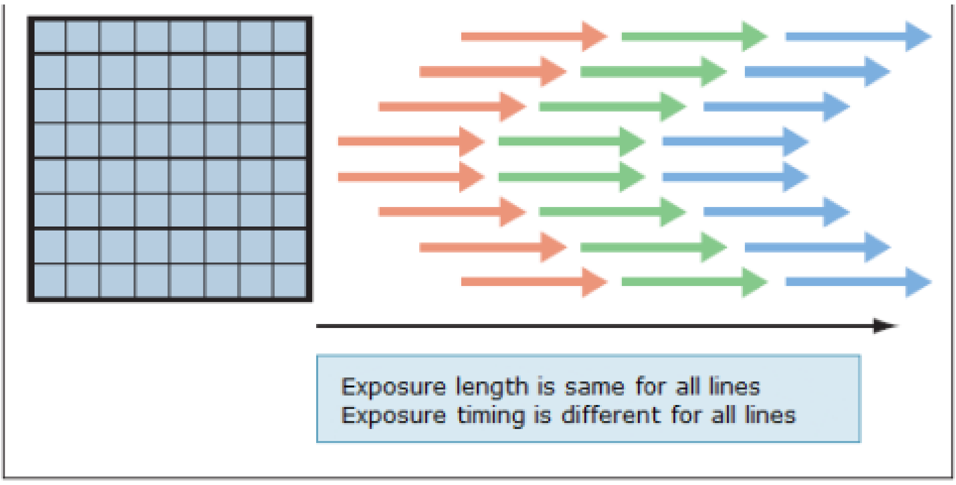

Exposure length is same for all lines
Exposure timing is different for all lines

Figure 3.6: The exposure timing of rolling shutter method [10]

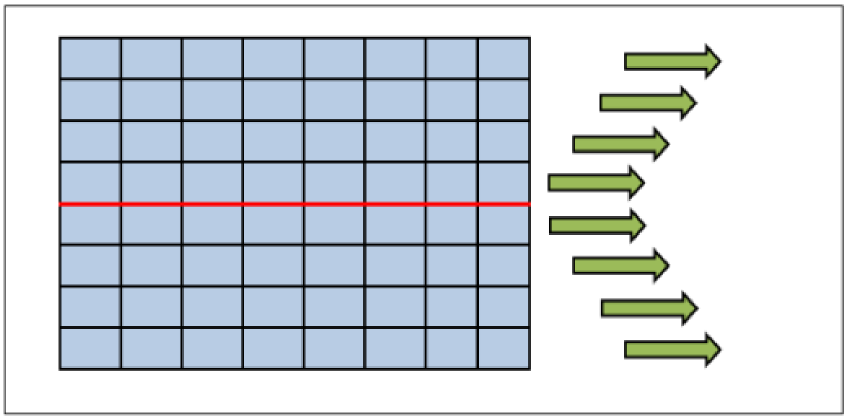


Figure 3.7: Readout method of the camera [10]

Rolling Mode. In normal mode, charge readout is performed for all pixels individually. In Binning mode, 2x2 or 4x4 binning modes are available. In this mode, adjacent 2 or 4 pixels are added that causes decreasing the resolution. In sub-array mode, a region of interest can be selected to readout. The last method is Rapid Rolling and it is preferred to acquire images of fast moving objects in order to minimize distortion effect of rolling shutter mechanism. Operation modes of the camera is shown in Figure 3.8.

In free running mode, the exposure and readout timing settings are set by software commands and controlled by an internal microprocessor. Normal readout mode is used when the exposure time is longer than readout time. On the contrary, if the

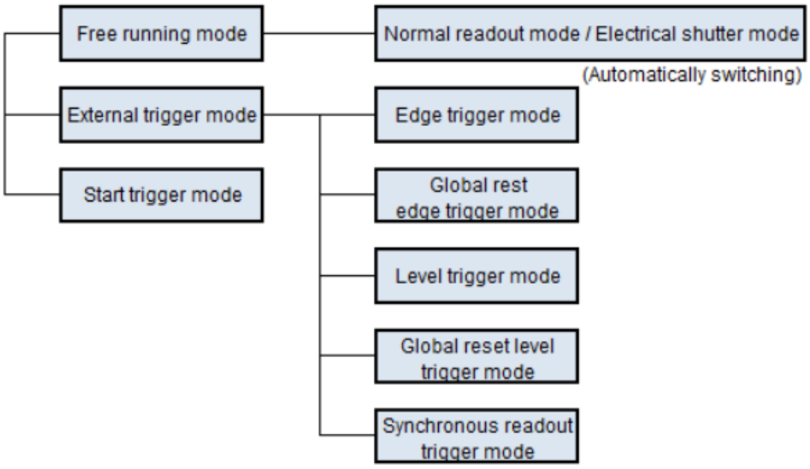

Free running mode
Normal readout mode / Electrical shutter mode
(Automatically switching)
External trigger mode
Edge trigger mode
Global rest edge trigger mode
Level trigger mode
Global reset level trigger mode
Synchronous readout trigger mode
Start trigger mode

Figure 3.8: Readout method of the camera [10]

exposure time is shorter than readout time of 1 frame, electrical shutter mode is used. The mode selection is done automatically according to exposure time selection. Table 3.2 shows the some possible fps values for free running mode based on readout time.

Table 3.2: fps values based on readout time for free running mode

Horizontal Width	Vertical Width	fps
2048	2048	30
	1024	60
	512	120
	256	240
	128	481

The captured frames that consist of 2048x2048 pixels, each with a 16 bit digital value transfer to the computer via USB3.0 interface. Figure 3.9 shows the timing chart for free running-normal readout mode of the camera.

Figure 3.10 is taken by the ORCA-Flash4.0 LT+ scientific CMOS camera at full resolution (2048x2048). The microchannel is located only a small part of the frame. Therefore, only the part that covers microchannel is selected at 2048x256 resolution instead of full resolution. In this way, readout time is decreased to 3.75ms from 30ms.

The exposure time is set to 10ms.The exposure time can be extended to collect more

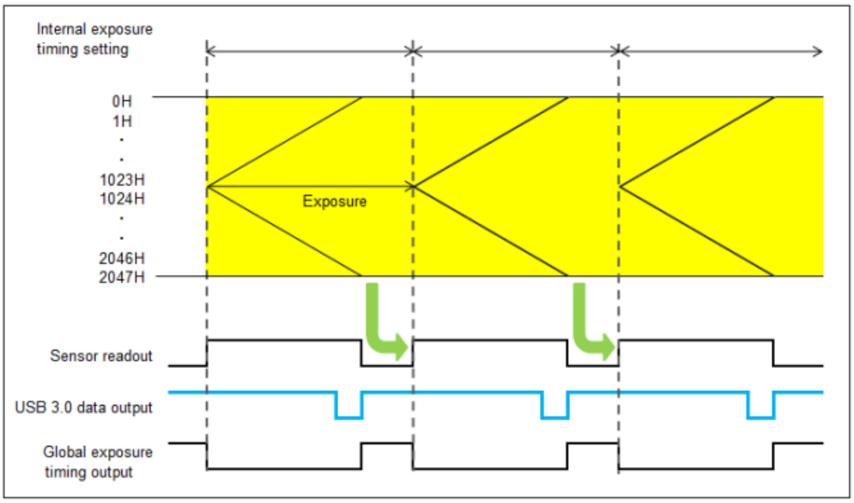


Figure 3.9: Timing chart for the free running, normal readout mode [10]

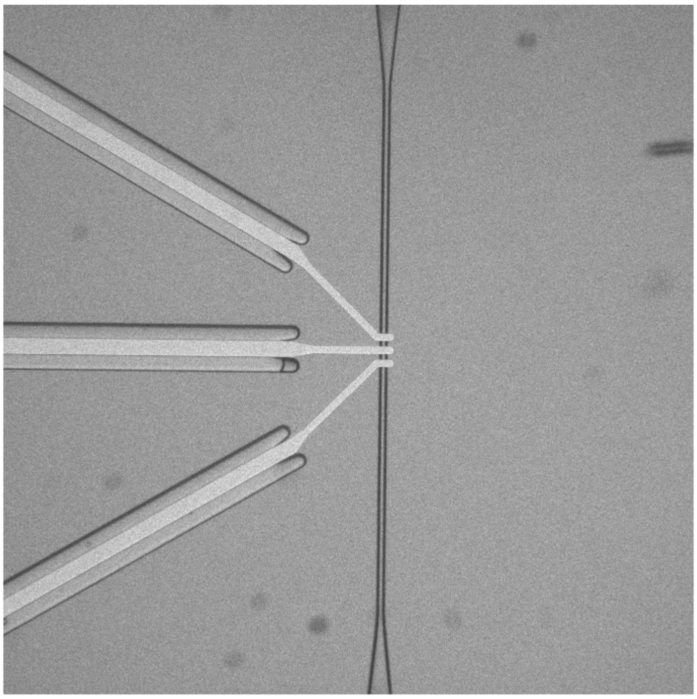


Figure 3.10: Full resolution (2048x2048) frame of the microchannel

photons, and increase the signal-to-noise ratio, however, moving objects may cause motion blur. Nevertheless, low exposure time causes low intensity values without motion blur. 10ms is an appropriate value in order to increase signal to noise ratio of the pixel with a small motion blur. Due to the motion of the cells in the exposure time, the cells appear longer than they are, that allows easier detection of the cells. According to timing chart shown in Figure 3.9, fps rate is directly related to exposure time since it is larger than the readout time. As a result, fps is calculated as 100 for 10ms exposure time.

Captured images are transferred to the computer via USB3.0 interface. In order to transfer all captured images, transfer rate of USB3.0 must be larger than the total image size in a second. Otherwise, some frames will be lost. Calculation of the total size of frames in a second with 16 bit pixel resolution is given in Equation 3.2. The Equation 3.2 gives 100 MB/sec that is less than transfer rate of USB3.0 (4.8Gb/sec) for 2048x256 frames taken at 100 fps.

$$TotalSize(MB/sec) = \frac{2(B/pixel) * H * V(pixels/frame) * fps(frame/sec)}{2^{20}} \tag{3.2}$$

Hokawo Imaging Software that is developed by HAMAMATSU is used for saving frames to the computer's hard disk by using streaming mode. In streaming mode, incoming frames are written to computer disk directly. In case the writing speed of the hard disk is lower than the data transfer rate, although a buffer is used for the data waiting to be written, data loss may occur, and the fps may decrease. This situation was experienced in the setup conditions used, and therefore fps value was measured about 97.5 fps. Captured consecutive frames at 97.5 fps are saved in HIS file format. HIS file is a special file type, and details about the HIS file format is given in Hokawo User Manuel Document [10]. In this file format, two header types are explained. The first header given in Table 3.3 is the header of the HIS file, and the second header given in Table 3.4 is the header of images within the sequence. After the HIS file is opened, the first header is read initially, and the necessary information for reconstruction of the images are taken. These information contains image width and height, total number of frames in the file, file type, and time stamp value that is the time tag of the camera for each frame. File type shows the number of bits representing each pixel, and time stamp value is high resolution time tag of each frames. Comment length value is also important for correct read of the file. The first frame is read according to file type after the first header, and reshaped according to height and width value. Then, consecutive headers and images are read until the total number of frames that is given in the first header are read.

Table 3.3: Header of HIS file

Bytes	Content
0-1	Character IM
2-3	Comment length in Bytes (CL)
4-5	Image width in pixels(W)
6-7	Image height in pixels(H)
8-9	X-offset
10-11	Y-offset
12-13	File Type (1: 8bit, 2: 16bit, 11: 24bit RGB)
14-17	Number of images in the file
18-19	0
20-21	0
22-29	Time stamp of image 0 (double)
30-33	Marker
34-63	Additional information
64 to 64+CL-1	Comment Area
64+CL to 64+CL+DL-1	Image Data

Table 3.4: Header of images within sequence

Bytes	Content
0-1	Character IM
2-3	Comment length in Bytes (CL)
4-5	Image width in pixels(W)
6-7	Image height in pixels(H)
8-9	X-offset
10-11	Y-offset
12-13	File Type (1: 8bit, 2: 16bit, 11: 24bit RGB)
14-17	0
18-19	0
20-21	0
22-29	Time stamp of image the next image(double)
30-33	Marker
34-63	Additional information
64 to 64+CL-1	Comment Area
64+CL to 64+CL+DL-1	Image Data (DL = H*W*Bytes from file type)

CHAPTER 4

PROPOSED CELL COUNTING AND ANALYSIS ALGORITHM

4.1 Introduction

In Chapter 3, the microchannel images are captured by the ORCA-Flash4.0 LT+ scientific CMOS camera in 2048x256 pixels sizes with 16 bit pixel resolution as in Figure 4.1.

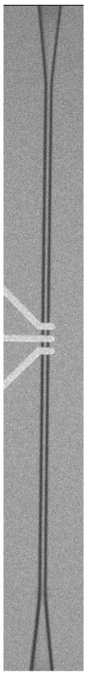


Figure 4.1: Cropped frame (2048x256) that covers the microchannel

Exposure time is set to 10msec that corresponds to 100 fps, and 97.5 fps is obtained experimentally. Captured frames are transferred to the computer via USB3.0 interface. By using Hokawo software, frames are saved to hard disk as a HIS file. The file is opened and frames are read according to HIS file format that is given in Table 3.3 and Table 3.4.

In this chapter, the proposed cell counting and analysis algorithm is explained. Cell counting and analysis algorithm is a tracking algorithm that consists of detection and data association parts. The aim of the algorithm is to find the number of cell passing through the microchannel, classify them according to sizes, and find the flow speed of the cells from the frames taken by the camera. The chapter is divided into three main sections. Section 4.2 is Automatic Cell Detection that is about image processing algorithms that are used for automatic detection of cells while they are flowing. The output of Automatic Cell Detection Section is input to the Section 4.3, Cell Tracking that is about tracking of cells by applying a data association algorithm. Finally, width and speed of counted cells are found, and clustering according to the size is done in Section 4.5, Cell Analysis. Figure 4.2 shows the flowchart of the algorithm.

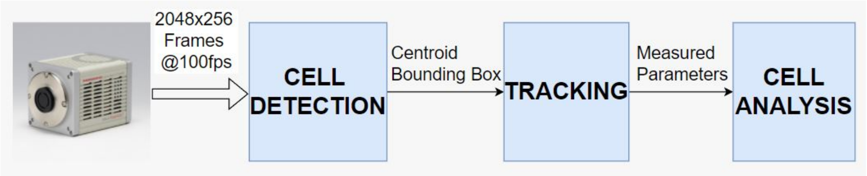


Figure 4.2: Flowchart of cell counting and analysis algorithm

4.2 Automatic Cell Detection

Initialization method of the cell counting and analysis algorithm is detection based meaning that objects to be tracked are found automatically from the detections. The automatic cell detection algorithm creates the observation model of the cell counting and analysis algorithm, and it finds the cells automatically. Automatic cell detection block takes frames from the camera, and detect the cells with their centroids and bounding boxes. As the first step, saved frames are read one by one from the HIS file that has special format as in Table 3.3 and 3.4. Initially, the format, size, timestampt and the total number of the frames are obtained from the headers. Then, a frame is read according to size and format information. At each iteration, a frame is read from the file with its header that includes timestampt value of that frame. At the end, the image is ready for image processing algorithms.

The proposed detection algorithm includes five steps. Initially, a low pass filter is applied to the frames in order to reduce the noise, and then the background is modelled. A constant threshold value is determined for cell segmentation. The segmentation results are processed with morphological operations. Finally, centeroid, length, and width of the blobs will be found and a bounding box is created by connected component labeling algorithm. Figure 4.3 shows the flow chart of the cell detection algorithm.

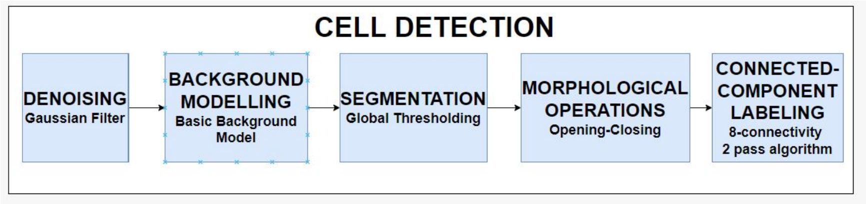


Figure 4.3: Flowchart of the cell detection algorithm

In this section, appropriate algorithms are selected for each part, and the results are showed by using the example microchannel images.

4.2.1 Image Noise and Denoising

CMOS image sensors suffer from different noise sources such as thermal noise, AD conversion noise, pixel noise, amplifier noise and readout noise[24]. The noisy parts prevent to obtain more information from the image. Therefore, they must be reduced or eliminated. Filtering is a well-known technique for noise reduction and image enhancing. Different methods are explained in Chapter 2. Gaussian filter is selected from explained filters since it has easier implementation and lower computational cost. Moreover, some of the noise source of image such as thermal, pixel circuit AD conversion has approximately Gaussian distributed characteristic that makes Gaussian filter more appropriate image filtering [56]. A Gaussian kernel with standard deviation 1 is created to suppress the noise, and the frames are filtered with that Gaussian kernel. The created Gaussian kernel with standard deviation 1 is given in

Figure 4.4.

	0.02	0.24	1.07	1.77	1.07	0.24	0.02
	0.24	2.92	13.07	21.55	13.07	2.92	0.24
	1.07	13.07	58.58	96.58	58.58	13.07	1.07
$\frac{1}{1000}*$	1.77	21.55	96.58	159.24	96.58	21.55	1.77
	1.07	13.07	58.58	96.58	58.58	13.07	1.07
	0.24	2.92	13.07	21.55	13.07	2.92	0.24
	0.02	0.24	1.07	1.77	1.07	0.24	0.02

Figure 4.4: Gaussian Kernel for low pass filtering

The statistical information can be helpful to understand filtering effect on the images. For this purpose, mean value, standard deviation, maximum and minimum values of the 10 frames from the microchannel images are used. Mean value and standard deviation is calculated by using Equation 4.1 and Equation 4.2, respectively.

$$\mu = \frac{1}{10}\frac{1}{2048}\frac{1}{256}\sum_{i=1}^{10}\sum_{j=1}^{2048}\sum_{k=1}^{256} Pixel_i(j,k) \tag{4.1}$$

$$\sigma = \sqrt{\frac{1}{10}\frac{1}{2048}\frac{1}{256}\sum_{i=1}^{10}\sum_{j=1}^{2048}\sum_{k=1}^{256}(Pixel_i(j,k) - \mu)^2} \tag{4.2}$$

In Equation 4.1, average value of the pixels is found. The standard deviation found by Equation 4.2 is a measure of the amount of variation of pixels from the mean. Maximum and minimum are helpful to determine impulsive changes on the pixel intensities.

Before filtering, the histogram plot of about 5 million pixels from 10 frames showing background that means there is no cell in the image is shown in Figure 4.5. Mean value is calculated as 104.9 with 4.78 standard deviation. Maximum pixel value is 185, and minimum pixel value is 50.

After filtering the frames with Gaussian filter given in Figure 4.4, a histogram plot shown in Figure 4.6 is obtained. Mean value remains same since Gaussian kernel has

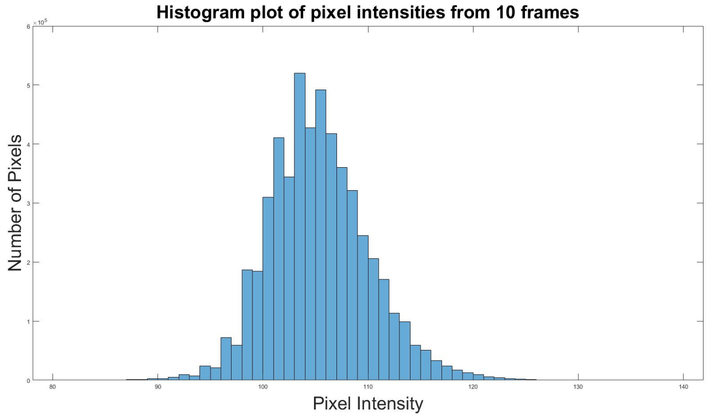


Figure 4.5: Histogram plot of background frames

zero mean, however standard deviation is decreased to, 1.495. Maximum pixel value is 118, and minimum pixel value is 95. These results shows that filtering decreases the variance, therefore the power of the noise, and eliminate sharp changes in the pixel intensities.

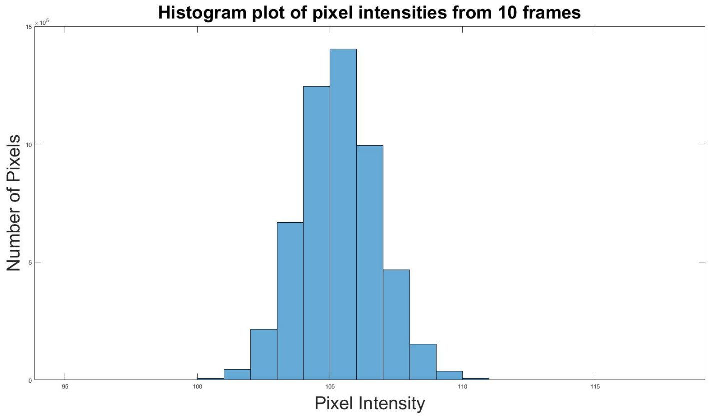


Figure 4.6: Histogram plot of filtered background frames

As explained in 2.2.1.2 part, Gaussian filter outputs a blurry image. Figure 4.7 is an example of original and blurry filtered image parts that contains cells. For more visible images, only some part of the frames are shown.

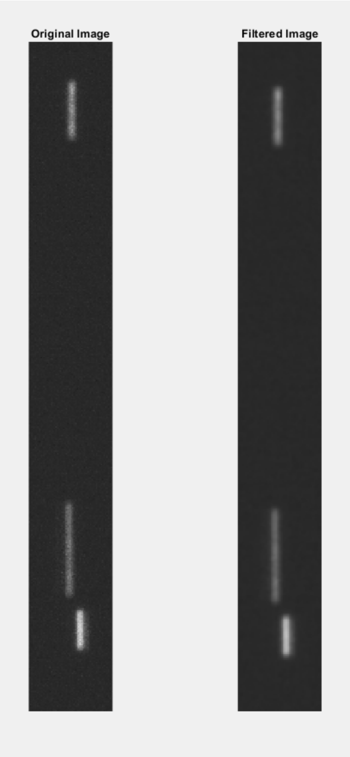


Figure 4.7: Original and filtered images containing cells

4.2.2 Background Modelling

Figure 4.1 shows the microchannel under daily light. However, cells are colored in order to become visible under a fluorescent light. Under this light, the only visible things are cells, and they need to be separated from background. Therefore, a background model must be created.

As explained in Chapter 2, there are different types of methods containing different algorithms. They are robust to different challenges. Therefore, an appropriate method should be chosen to be used under the setup conditions. There is a stable fluorescent light source, therefore, there is no illumination change problem. The only moving things are the cells, and there is no other object to causes small movements meaning that there is no dynamic background problem. Shadows can be seen for some cells, and video noise is only originated from the camera. Moreover, the camera is stationary, that provides no challenge due to movement of the camera. As a result, choosing a basic background model method is enough to detect cells, and provides low computational cost. Shadows can be eliminated in segmentation step by determining appropriate threshold. Table 4.1 shows the parameters obtained from the histogram of about 5 million pixels taken from 10 background frames.

Table 4.1: Parameters of the background model

Mean	104.9
Standard Deviation	1.495
Mod	105
Median	105
Maximum	118
Minimum	95

4.2.3 Segmentation of Cells

Segmentation can be described as process of partitioning an image into component regions or objects . There are different segmentation techniques for different cases as explained before. In the previous part 4.2.2, the background model was obtained as in Table 4.1. For segmentation of cells, global thresholding method is used due to its simplicity and effective results for the experiment. This method is used for similar cases in [57] and [58]. A threshold value between the background and the cells must be determined. Therefore, the intensity of the cell is analyzed for different cases. There is no exact range for cell intensities since cells can flow from any depth within the microchannel that causes different intensities at different level. Figure 4.8 shows an example for intensity values for 3 different cells in a frame. Brighter cells have edges with higher pixel values than other cells body. The selected threshold value must separate each cell from the background and ensure that brighter cells do not appear wider. Moreover, it should eliminate the shadows.

As a result, 160 is selected as threshold, experimentally. After segmentation, a binary image is obtained that contains only 0 or 1. The pixels with value less than threshold value belong to the black background, 0, and remaining pixels segmented as white cells, 1. Figure 4.9 shows filtered image and segmentation result.

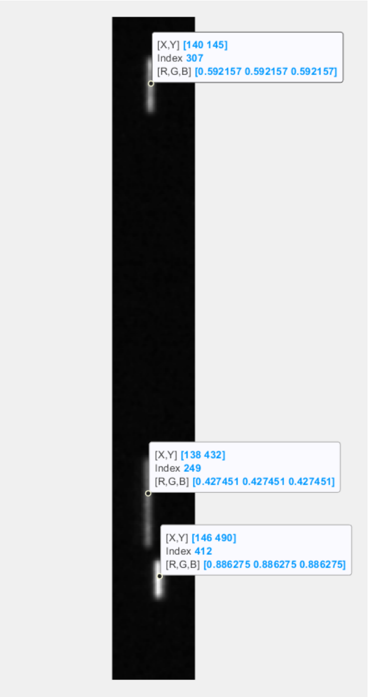

[X,Y] [140 145]
Index 307
[R,G,B] [0.592157 0.592157 0.592157]
[X,Y] [138 432]
Index 249
[R,G,B] [0.427451 0.427451 0.427451]
[X,Y] [146 490]
Index 412
[R,G,B] [0.886275 0.886275 0.886275]

Figure 4.8: Intensity values for different cells in a frame. Index values, 307,249, and 412 are the pixel intensities.

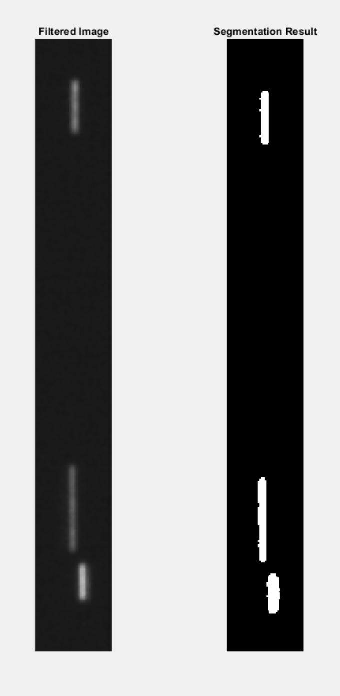


Figure 4.9: Filtered image and segmentation result

4.2.4 Morphological Operations

There are some imperfections due to segmentation. Some pixels between the background and the cells are labeled as a part of cells. Moreover, there can be small white pixel regions in the image that can be called as noise. In order to remove these imperfections and noisy parts, morphological image processing techniques is used [39]. A rectangle structuring element is selected with the size of 10x2 for opening operation since moving cells look like a rectangular shape. By using opening operation, small regions less than 10x2 sizes are eliminated, and imperfections of the segmentation is eliminated. Figure 4.10 shows the image after segmentation and opening result.

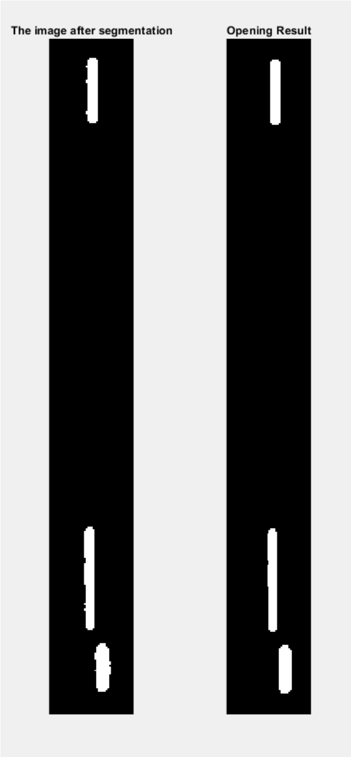


Figure 4.10: The image after segmentation and opening result

As it seen in Figure 3.10, there are electrodes in the microchip, and some part of the microchannel is under the electrodes. Those electrodes cause invisible region in the microchannel. While cells are flowing in the microchannel, they observed in a rectangular shape due to exposure time. When they go under the electrodes, some part of the cells become invisible. Therefore, cells may be divided into 2 different cells in some frames, and that causes wrong input for the next algorithms. 2 parts of the cells are merged with closing operation with a rectangular structuring element with the size 30x4. Figure 4.11 shows an example for that case. There are 3 consecutive frames in Figure 4.11. The first frame shows the cell before it passes under the electrode. The second frame shows the cell while it is passing under the electrode. The last frame shows the cell after it has passed under the electrode. Figure 4.12 shows the result of closing operation for the same case in Figure 4.11. As a result, cell division is prevented by using closing operation from morphological image processing.

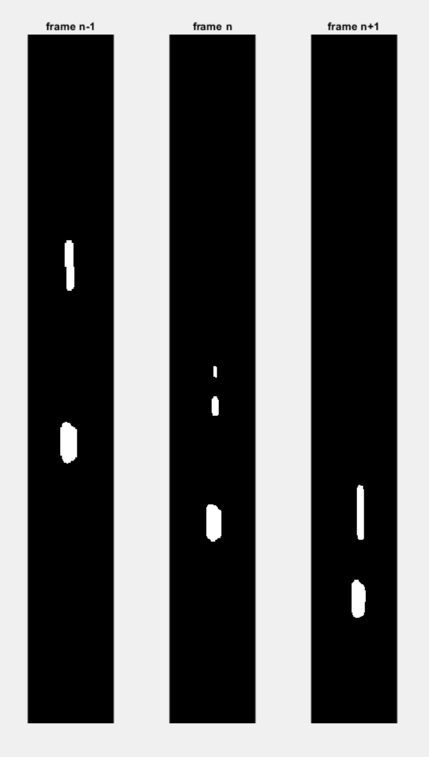


Figure 4.11: 3 consecutive frames that show the effect of invisible region. Frame n-1: before cell passes under the electrode. Frame n: while the cell is passing under the electrode. Frame n+1: After the cell passed under the electrode.

4.2.5 Connected-Component Labeling

Final step before the tracking section is finding connected components, blobs. and labeling them. Connected component labeling algorithm aims to find and label connected pixel regions in the binary image with center of mass and size information.

The explained connected-component labeling method, 2 pass algorithm with 8 con-

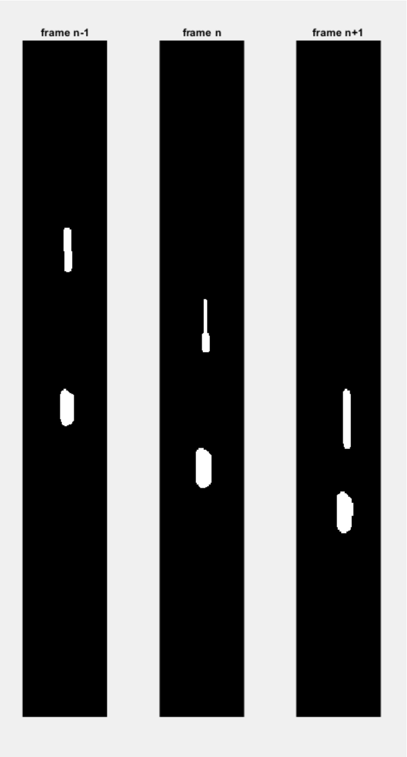


Figure 4.12: 3 consecutive frames that show the result of closing for invisible region. Frame n-1: before cell passes under the electrode. Frame n: while the cell is passing under the electrode. Frame n+1: After the cell passed under the electrode.

nectivity in Chapter 2, is used. Figure 4.13 shows an example of labeling and bounding box result. Cells are covered by green bounding box with label id. Moreover, centroid of each cell marked with a blue star. These parameters are the necessary inputs for the tracking part, Section 4.3. The region having less than 21 pixels are not considered as the cell. This is the last precaution taken in addition to denoising, and morphological operations to avoid erroneous detection.

4.3 Cell Tracking

Cell tracking part is the second main part of the cell counting and analysis algorithm. This part takes the input, centroid and bounding box of each labeled components from the cell detection part for each frame. These labeled components, cells, are followed with a unique ID during the flow in the microchannel. The parameters such as start-

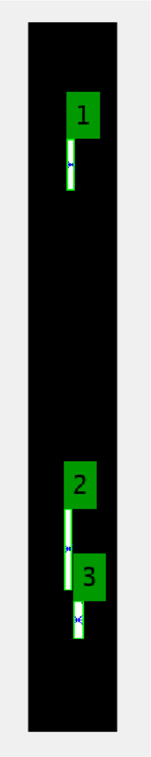


Figure 4.13: An example result for connected-component labeling

end time and location of a cell in the microchannel, width of the cell are used in the cell analysis part.

As stated previously, the main purpose of the cell tracking part is to follow each labeled component with a unique identifier, ID, during the flowing process. To achieve the aim, detections between frames should be linked correctly. At this point, the motion model is an important parameter to link detections between consecutive frames. If the motion model is known, next location of the object can be estimated by using a mathematical model. When a new frame is taken, detections and estimations are compared, and they are linked if they are close enough to each other. However, measurements and motion model may not be exactly true, that means measurements can include noise, and time difference between frames may not be the same that causes an error in the mathematical motion model. Moreover, each object can move at different speeds. Another major challenge is that a detection become invisible for a few frames when it passes under the electrodes, or when two detections move together and seen as a single object during a part of the flow. Since there are many challenges to follow each detection while they are flowing, a tracker for each object is used for making prediction of the next location of the object. This tracker must model the motion of objects between consecutive frames, and must be robust for speed changes, invisible frames, noise in the measurements and the processes. Kalman filter that is explained in Chapter 2 is used to next position estimations.

4.3.1 Cell Tracking Algorithm Flow

Automatic cell detection block provides centroid and bounding box information of all labeled components. The first step of the tracking algorithm is predicting new locations by using Kalman filter. The second step is generating a cost matrix that contains a special distance between detections and predictions. This cost matrix is solved by Hungarian assignment algorithm with a non-assignment cost parameter. This algorithm provides prediction and detection matches if it is possible. On the other hand, a prediction may not be assigned any detection or a detection may not be matched with a prediction. In the case of unassigned detection, a new tracker is created. For the case of unassigned prediction, predicted value is saved to be used in the next cost matrix. After assignment part, Kalman filter corrects its estimation with measurements. If a cell leaves the microchannel, its tracker is deleted so that it is not used in assignment problem.

The flow of the algorithm is as follows:

- Take detections from cell detection part.
- Take the predictions from tracker.
- Generate a cost matrix by using distance between predictions and detections.
- Solve the cost matrix to link detections and predictions.
- If there is a prediction that is not assigned to any detection, update tracker with last prediction.
- If there is a detection that is not assigned to any prediction, create a new tracker.

- Update location parameter of the tracker by using assignments from the cost matrix.
- Update related parameters of unassigned trackers.
- Delete the tracking if it satisfies invisibility condition.

The tracking algorithm is also represented with the flow chart as in Figure 4.14.

4.3.1.1 Tracker Structure

A structure called as tracker is created for each object. This structure includes all necessary information to be used in tracking. Table 4.2 shows the tracker parameters.

Table 4.2: Parameters of tracker structure

ID
Bounding box
Centroid
Kalman filter
Time stampt
Age
Consecutive Invisible Count
Width

ID is the unique identifier for a tracker and it is different for all objects. ID is assigned when a tracker is generated and never changes. Bounding box is the rectangle shape drawing that includes whole object. It is updated at each step and useful for visualization. If the object detected in the new frame, new bounding box is assigned to this part. If it is not detected, previous bounding box is preserved to show estimated position of the object while it is not visible.

Centroid shows the last position of the object and updated at each step. If the object is detected in the new frame, centroid is updated with new position. On the other hand,

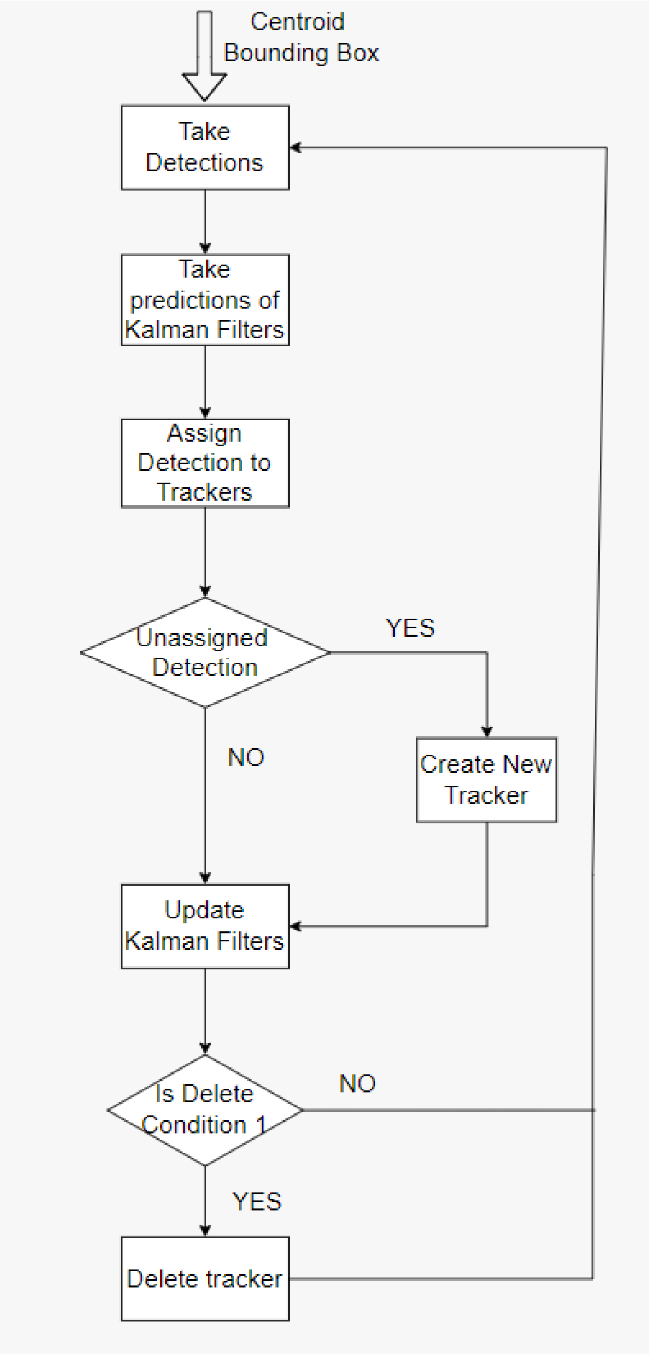


Figure 4.14: Flow chart of the tracking algorithm

centroid is updated with the estimation of Kalman filter. This parameter is used for speed analysis of the object, and important for delete condition of the track.

Kalman filter is the tracker method that is used for estimating next position of the object. Each tracker has its own Kalman filter, and multiple object tracking is done by using single trackers. The Kalman filter is created with centroid of the object when it is detected firstly. Kalman filter predicts the next position of the object and this estimation is used in detection-tracking assignment part. After detection-tracking part, Kalman filter is updated with new measurements. If it is not updated manually due to non-assignment, it uses last prediction for the next prediction.

Timestampt parameter comes from the frame, and shows the timer of the camera when the frame is taken. It is updated at each frame, and used for high accuracy speed analysis of the object since time difference between consecutive frames are not constant always, and fps can be changed due to different reasons.

Age parameter shows the number of frames that the tracker exists. Consecutive invisible count is a critical parameter and shows the number of frames that the tracker is not matched with any detection. It is used in deleting condition of the tracker.

Width is an important parameter for classification of the cells with respect to sizes. Width component of the tracker is an array of width values measured in all frames that the tracker exists.

4.3.1.2 Detection to Tracking

When a new frame comes with detections, these detections should be matched with the estimations of trackers so that objects can be followed correctly. At this step, a cost matrix is constructed that shows distance between each tracker and detection to solve assignments of detections to tracks. Cost matrix includes distance values between estimations and detections. Hungarian assignment algorithm is used to solve cost matrix with the minimum cost.

Distance is calculated by Equation 4.3, and it takes into account the Euclidean distance between the predicted centroid of the tracker and the centroid of the detection. It also includes the confidence of the prediction that is the covariance of the predicted state and the process noise of Kalman filter.

$$d(z) = (z - Hx)^T \Sigma^{-1} (z - Hx) + ln|\Sigma| \tag{4.3}$$

where $\Sigma = HPH^T + R$ and $|\Sigma|$ is the determinant of Σ.

There are 4 different possible cases for detection-tracking matching. The first case is that the number of detections and trackers are the same, and these detections belong to these trackers. This case is straightforward and detections are assigned to trackers by solving the cost matrix for the minimum total cost. The second case is that the number of detections are larger than trackers due to first detection of objects. In this case, the cost matrix is solved for the minimum total cost, and new trackers are created for unassigned detections. The third case is that there are more trackers than detections due to invisible moments. In this case, the cost matrix is solved for the minimum total cost, and unassigned trackers are updated with predicted values. The final case is that the number of detections and trackers are the same, however, some object may detected firstly, and some trackers may wait for deleting. In such cases, the distance values are large. The cost matrix is solved for the minimum total cost, and assignment is done with a distance value that is larger than the non-assignment cost value, and assignment is ignored. As a result, instead of matching new detections with the dead trackers, new trackers can be created for new objects, and trackers waiting to be deleted preserved with predicted values.

4.3.1.3 Deleting A Tracker

A tracker is created with the first detection of an object, and used until the object leaves the microchannel. The consecutive invisible count parameter is used for deciding to delete a tracker or not. If consecutive invisible count is equal to 3, and the location of the object near the end of the microchannel, tracker is deleted so that it is not used in detection to tracking assignment problem. Note that, the location of the object is also important since an object may not be visible in the microchannel for several reasons, and tracker must be preserved at these moments. Otherwise, when an object seen again, it would take a new ID that causes wrong cell count.

4.4 An Example For Tracking Algorithm

In this part, an example of tracking algorithm including all possible assignment cases described in detection to tracking part is given. At a random time, while frame 200 is taken, there are 2 detections with their trackers as shown in Figure 4.15 and Figure 4.16.

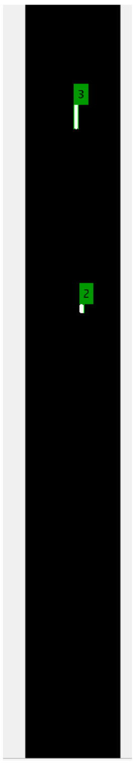
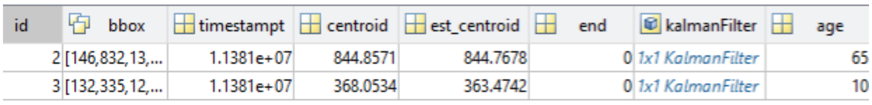


Figure 4.15: Frame 200 having 2 cells



id	bbox	timestampt	centroid	est_centroid	end	kalmanFilter	age
2	[146,832,13,...	1.1381e+07	844.8571	844.7678	0	1x1 KalmanFilter	65
3	[132,335,12,...	1.1381e+07	368.0534	363.4742	0	1x1 KalmanFilter	10

Figure 4.16: Existing trackers when frame 200 is taken

Algorithm starts with taking the estimations from Kalman filters of each trackers. Kalman filter estimations are listed in Table 4.3, and detections from frame 200 are listed in Table 4.4.

Distance between each detection and tracker is calculated by using Equation 4.3, and

Table 4.3: Kalman filter estimations when frame 200 is taken

ID	column	row
2	152	845
3	138	363

Table 4.4: Detections from frame 200

column	row
138	368
151	845

cost matrix, C is obtained as follows.

$$C = \begin{bmatrix} 418.37 & 12.66 \\ 12.70 & 426.32 \end{bmatrix}$$

Since the 2 cells are far from each other, each tracker has one small and one large cost value. When the cost matrix is solved, tracker 1 with ID 2 is assigned to the second detection, and tracker 2 with ID 3 i assigned to the first detection. Trackers are updated with new measurements. Cell with ID 3 has larger length that means it has larger speed. Therefore, after 10 frames, 2 cells overlap and appear as one connected component as in Figure 4.17.

Algorithm starts with taking the estimations from Kalman filters of each trackers. Kalman filter estimations are listed in Table 4.5, and detection from frame 211 is listed in Table 4.6..

Table 4.5: Kalman filter estimations when frame 211 is taken

ID	column	row
2	154	1068
3	145	1073

Distance between each detection and tracker is calculated by using Equation 4.3, and

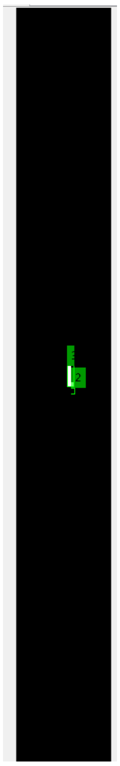


Figure 4.17: Frame 211 having overlapped cells

Table 4.6: Detection from frame 211

column	row
146	1073

cost matrix, C is obtained as follows.

$$C = \begin{bmatrix} 15.20 \\ 12.66 \end{bmatrix}$$

Since position of two cells are close, cost values are closer. When the cost matrix is solved, tracker 2 with ID 3 is assigned to the detection. Tracker 2 with ID 3 is updated with new measurement while tracker 1 with ID 2 is updated with the estimated value. After 2 frames, the cell with ID 3 exceeds the cell with ID 2, and become detectable as a single object again as in Figure 4.18.

Algorithm starts with taking the estimations from Kalman filters of each trackers. Kalman filter estimations are listed in Table 4.7, and detections from frame 214 are

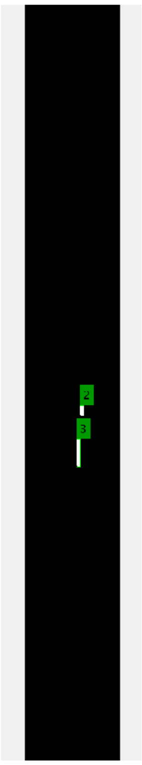


Figure 4.18: Frame 214 when 2 cells separate

listed in Table 4.8.

Table 4.7: Kalman filter estimations when frame 214 is taken

ID	column	row
2	154	1174
3	148	1177

Table 4.8: Detection from frame 214

column	row
145	1211
154	1097

Distance between each detection and tracker is calculated by using Equation 4.3, and

cost matrix, C is obtained as follows.

$$C = \begin{bmatrix} 19.20 & 18.25 \\ 14.80 & 24.06 \end{bmatrix}$$

When the cost matrix is solved, tracker 2 with ID 3 is assigned to the first detection, and tracker 1 with ID 2 is assigned to the second detection. Trackers are updated with new measurements. After 10 frames, a new cell is enter the microchannel. Estimations before frame 224 is listed in Table 4.9, and detections from frame 224 are listed in Table 4.10.

Table 4.9: Kalman filter estimations when frame 224 is taken

ID	column	row
2	156	1292
3	151	1861

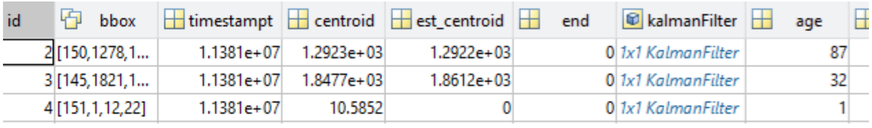
Table 4.10: Detection from frame 224

column	row
150	1874
155	1292
156	10

Distance between each detection and tracker is calculated by using Equation 4.3, and cost matrix, C is obtained as follows.

$$C = \begin{bmatrix} 563.18 & 12.66 & 2942 \\ 12.98 & 589.87 & 6121 \end{bmatrix}$$

Since position of two cells are close, and the new cell is far away from other two cells, distance values for detection 3 are larger. When the cost matrix is solved, tracker 2 with ID 3 is assigned to the first detection, and tracker 1 with ID 2 id assigned to the second detection. Trackers are updated with new measurements. However, the third detection is not assigned to any tracker. Therefore, a new tracker with ID 4 is created as in Figure 4.19. After new tracker is created, Figure 4.20 is obtained.



id	bbox	timestampt	centroid	est_centroid	end	kalmanFilter	age
2	[150,1278,1...	1.1381e+07	1.2923e+03	1.2922e+03	0	1x1 KalmanFilter	87
3	[145,1821,1...	1.1381e+07	1.8477e+03	1.8612e+03	0	1x1 KalmanFilter	32
4	[151,1,12,22]	1.1381e+07	10.5852	0	0	1x1 KalmanFilter	1

Figure 4.19: Trackers after frame 224 is processed

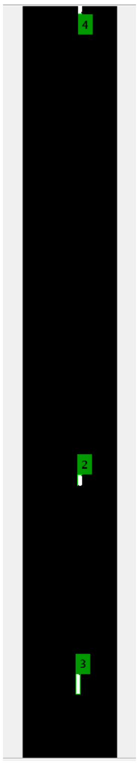


Figure 4.20: Frame 224 when a new cell is entered to the microchannel

After 9 frames, cell with ID 3 leaves the microchannel. In that frame, Kalman filter estimations are listed in Table 4.11, and detections from frame 233 are listed in Table 4.12.

Table 4.11: Kalman filter estimations when frame 233 is taken

ID	column	row
2	157	1470
3	148	2060
4	144	256

Distance between each detection and tracker is calculated by using Equation 4.3, and

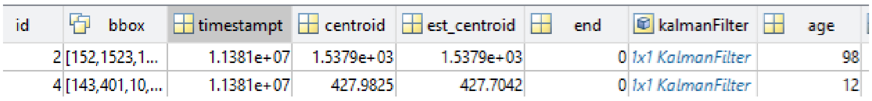
Table 4.12: Detection from frame 233

column	row
146	271
158	1470

cost matrix, C is obtained as follows.

$$C = \begin{bmatrix} 2578 & 12.65 \\ 5721 & 632.78 \\ 12.91 & 2631 \end{bmatrix}$$

When the cost matrix is solved, tracker 3 with ID 4 is assigned to the first detection, tracker 1 with ID 2 is assigned to the second detection. However, tracker 2 with ID 3 is not assigned to any detection. After 3 frames, tracker with ID 3 is deleted since that cell left the microchannel. Remaining trackers are shown in Figure 4.21, and Figure 4.22 shows frame 233.



id	bbox	timestampt	centroid	est_centroid	end	kalmanFilter	age
2	[152,1523,1...	1.1381e+07	1.5379e+03	1.5379e+03	0	1x1 KalmanFilter	98
4	[143,401,10,...	1.1381e+07	427.9825	427.7042	0	1x1 KalmanFilter	12

Figure 4.21: Trackers after frame 233 is processed

4.5 Cell Analysis

Cell analysis part consists of classification of cells according to sizes, and speed analysis to interpret effect of flow rate on the cells. Cells appear circular when stationary, but appear larger than their actual size as they glow under fluorescent light. The width, that is described as the maximum number of pixels belongs to the object on the horizontal axis, of the cells can be used for size classification since cells are circular. There are 2 types of cells, and their actual sizes are about 6μm and 10μm. Blurry edges are also a challenge for correct width measurement. For an effective width classification, width measurements must be correct.

When the cells move in the microchannel, they lengthen in the direction of movement,

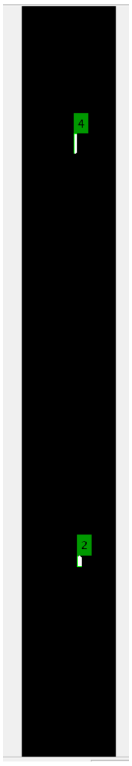


Figure 4.22: Frame 233 when a cell left the microchannel

and proportional to velocity due to the exposure time, and they appear as a rectangle. This time, the effect of the light scattered from the cells on the width decreases depending on the velocity. It is difficult to model how much the width decreases since each cell can have a different velocity. Therefore, it is not easy to estimate the actual size correctly from the measured width of the cells in motion. As a solution to this situation, a width value was created by using the width values measured in the area where the cells appear circular before they start moving in the microchannel. However, in this case, partial visibility of the cell before entering the frame also causes erroneous measurement. As a solution to all problems, the width values measured between 25 and 250 horizontal pixels of the image were kept, and their averages were used as the width of that cell.

Although rare, some cells enter the microchannel together and a new cell is detected when they are separated in the microchannel. However, its position has already moved out of the width measurement area. Moreover, since it is in motion, the measured width value will not reflect reality. In such cases, the first detected width value of the cell is saved and used. A flag is also kept indicating that this value is unreliable. In the following sections, classification method used for clustering the cells according to their widths, and speed measurement is explained in detail.

4.5.1 Classification of Cells

An important part of the automatic cell counting algorithm is classification of the cells to determine how many cells are in which size class such as 6μm or 10μm. The cells must be classified according to the measured width. In order to obtain statistical information of measured pixel width of each class, measurements are taken from the cells from each class. Figure 4.23 shows the histogram plot of cells' width when 6μm cells are used only. In this case, 545 cells are counted, and most of the cells are measured with widths between 7.5 and 10 pixels. However, there are a few number of cells with larger widths. Likewise, Figure 4.24 shows the histogram plot of cells' width when 10μm cells are used only. 322 cells are counted, and most of the cells are measured with widths between 13 and 17 pixels, and there are some cells with too small or large widths. As seen in Figures 4.23 and 4.24, most of the measurements are centered around a certain value, however, few cells measured much larger or much smaller. Larger width measurements may result from adherent cells, and smaller widths may result from movement of cells in the width measurement area. These values may classified wrongly. Since their effect on the total ratio is little, this error has negligible effect on the results.

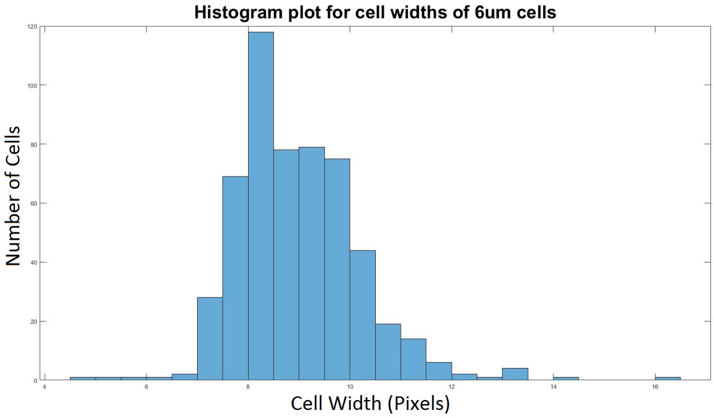


Figure 4.23: Histogram plot of cell widths for 6μm cells

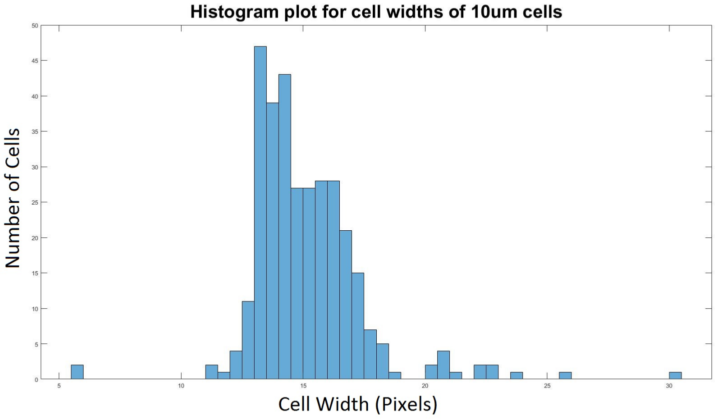


Figure 4.24: Histogram plot of cell widths for 10μm cells

In the real case, cells from both size classes exist in the solution, and they are counted together. Most of the measurements will center around 2 different pixel values, 9 and 15. However, there will be an intersection area where it cannot be determined which measurement belongs to the class. In order to divide measurements into 2 class, an unsupervised clustering method, K-medoids, is used.

Figure 4.25 shows the histogram plot of cell width that is the union of the widths in Figure 4.23 and 4.24. Classification result is also shown in Figure 4.25 with different colors.

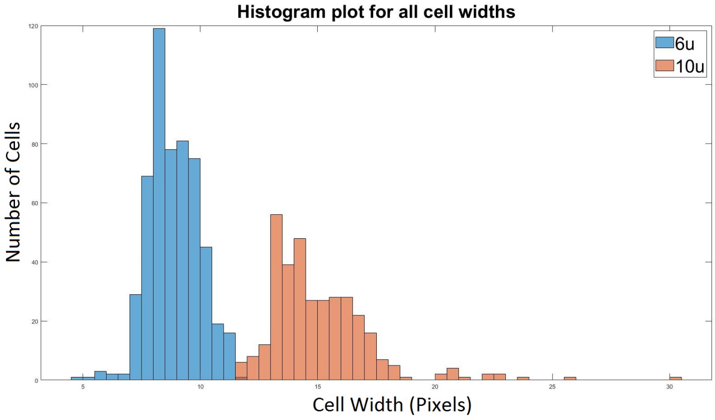


Figure 4.25: Histogram plot of cells width created by combining the 6-10μm measurements

It is known that, there are 545 cells with the mean value 9 from the 6μm sample set, and 322 cells with the mean value 15 from the 10μm sample set. Algorithm outputs 2 clusters with 536 and 331 elements by determining the 8.75 and 14.63 as the medoids. Originally, 62.86% of the total cells belong to the 6μm sample set, and 37.14% belong to the 10μm sample set. According to K-medoids classification, 61.82% of the total cells belong to the 6μm sample set, and 38.18% belong to the 10μm sample set. It is expected to obtain different classification result due to erroneous measurements and widths in the intersection area. Nevertheless, K-medoids algorithm provides a good classification result. Note that this algorithm does not depend on the actual values, but rather on the distribution of width measurements. Therefore, it can be used to separate any 2 groups of cells of different sizes. It can also be used not only to classify into 2 but also to divide into more groups. However, it should not be forgotten that the total number of groups must be known and the cell sizes must be sufficiently separated from each other.

4.5.2 Speed Analysis

Speed analysis is necessary part for the detecting relationship between liquid pressure and flow rate in the microchannel. For this purpose, average speed is calculated by using Equation 4.4.

$$v = \frac{x_1 - x_2}{t_1 - t_2} \tag{4.4}$$

where x_1 is the last position of the cell observed at time t_1 and x_2 is the first position of the cell observed at time t_2 . The start position of the microchannel is around 200 pixel bottom from the top of the frame. Moreover, end position of the microchannel is around 1800 pixel bottom from the top of the frame. Therefore, start time of each tracker is saved when the current location is about 200 pixels bottom from the top of the the frame, and end time of each tracker is saved when the current location is about 1800 pixels bottom from the top of the frame as shown in Figure 4.26.

As it explained in Chapter 3, pixel width is 6.5μm corresponding 1.3μm in 5x zoom mode. Therefore, the difference between start and end pixel location of the cell is multiplied with the 1.3μm, and real amount of the displacement is calculated. Time information is taken from the timestampt parameter. The timestampt is read from the header of the frame, and it is the time tag of the camera. Figure 4.27 shows an

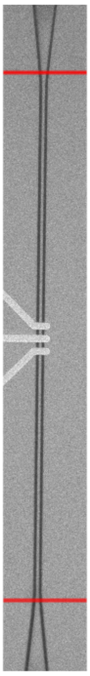


Figure 4.26: The microchannel with red horizontal lines that shows start and end positions for the speed analysis

example histogram plot of cell velocities from 623 cells. Mean value of the velocity is measured as 7.13μm/msec.

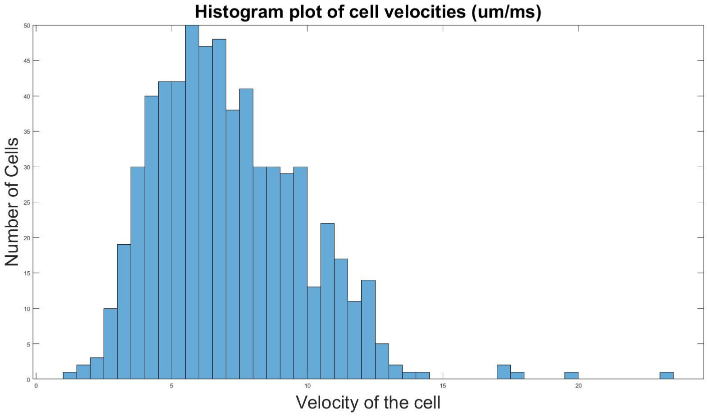


Figure 4.27: Velocity histogram of 623 cells in μm/msec unit

CHAPTER 5

EXPERIMENTAL RESULTS

In order to test the proposed cell counting and analysis algorithm explained in Chapter 4, various tests were carried out under the setup conditions described in Chapter 3. For this purpose, image sequences were taken and processed in various situations on 2 different days. On the first day, images were recorded for the case where only one type of cell was passed through the microchannel initially. By using these images, cell counting accuracy of the algorithm was tested. Afterwards, images in which cells of different sizes were passed through the microchannel were recorded, and both the number of cells and their classification according to their size were tested. On the second day, a solution containing cells from both dimensions was prepared, and images were recorded while different pressure values are applied. By using these images, both the number of cells and their classification according to their size were tested. Moreover, effect of the pressure on the algorithm is also tested. The proposed algorithm also applied without Gaussian filter to show effect of noise in the images. In addition, the proposed algorithm without morphological operations is applied to all recorded sequences. In this Chapter, the test results obtained on cell counting, classification by size, and velocity analysis are shown and evaluated. In all experiments, each image sequence contains 5974 frames that are recorded for 1 minute. Each HIS file size is 5.81GB. The proposed algorithm is implemented in MATLAB, and run by Intel(R) Core(TM) i7-6700 CPU @3.40GHz. Computation times are provided with each result.

5.1 Experiment 1

On the first day experiment, initially, the solution containing 500.000 cells with 6μm size per milliliter are passed through the microchannel, and 7 sequences of images were recorded. The cells in the images were counted both by the proposed algorithm and carefully by eye. In order to show the importance of Gaussian filtering and morphological operations, the proposed algorithm applied without them. In Table 5.1, the numbers obtained by counting the cells by eye, by proposed algorithm, by proposed algorithm without Gaussian Filter and by the proposed algorithm without morphological operations are shown. In addition, wrong counts between the proposed algorithm and the eye count is shown in the table.

Table 5.1: Comparison of cell counts in 7 image sequences for 6μ case

Seq.	Prop. Alg.	Eye	Error	W/o Gaussian Filt.	W/o Morph. op.
1	99	99	0	99	137
2	127	126	1	128	146
3	117	118	1	117	154
4	57	57	0	57	67
5	69	69	0	69	95
6	41	41	0	42	56
7	42	42	0	42	59
Total	**552**	**552**	**2**	**554**	**714**

Results shows that 552 cells are counted by both method, proposed algorithm and eye. Although the total cell counts are the same, there are 2 wrong counts. When the cause of the wrong cell count was investigated, it was observed that some cells moved together along the microchannel and therefore could not be counted as separate cells. The desired pressure value of the pressure controller device is set to 2 mbar. However, it is observed on the application user interface of the pressure controller device that the device does not always provide constant pressure. This situation results in cells to move at different speeds. Speed plot of the cells is given in Figure 5.1 that demonstrates that cells have different speeds. Average speed of the cells are calculated as 4μm per millisecond.

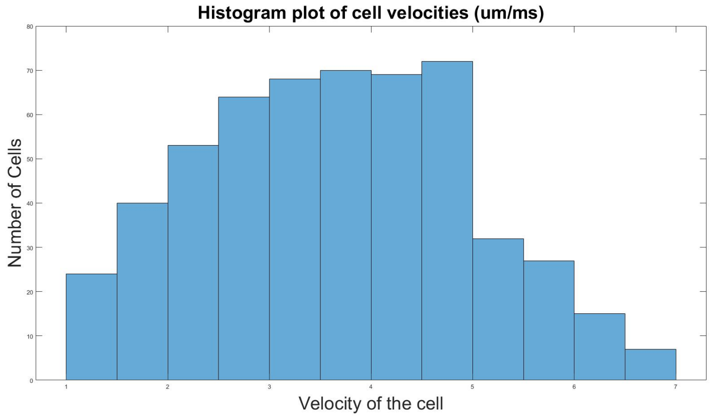


Figure 5.1: Speed plot of the cell with 6μm size

In the second case, the solution containing 500.000 cells with 10μm size per milliliter are passed through the microchannel, and 7 sequences of images were recorded. The cells in the images were counted both by the proposed algorithm and carefully by eye. In order to show the importance of Gaussian filtering and morphological operations, the proposed algorithm applied without them. In Table 5.2, the numbers obtained by counting the cells by eye, by proposed algorithm, by proposed algorithm without Gaussian Filter and by the proposed algorithm without morphological operations are shown. In addition, wrong counts between the proposed algorithm and the eye count is shown in the table.

Table 5.2: Comparison of cell counts in 7 image sequences for 10μ case

Seq.	Prop. Alg.	Eye	Error	W/o Gaussian Filt.	W/o Morph. op.
1	13	13	0	13	28
2	171	171	0	175	216
3	80	81	1	83	131
4	37	37	0	37	66
5	6	6	0	6	8
6	10	10	0	10	15
7	14	14	0	14	26
Total	**331**	**332**	**1**	**338**	**490**

Results shows that 331 cells are counted by the algorithm and 332 cell are counted by eye. The difference between the results was originated from the cells moving together along the microchannel. The desired pressure value of the pressure controller device is set to 2 mbar, again. The histogram of cell velocities is given in Figure 5.2 that demonstrates that cells have different speeds. Average speed of the cells are calculated as 4.5μm per millisecond.

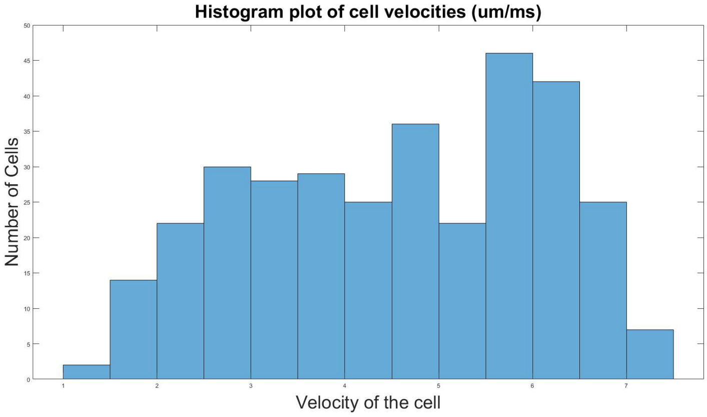


Figure 5.2: Speed plot of the cell with 10μm size

As the final experiment of the first day, the solution containing 250.000 cells with 6μm size and 250.000 cells with 10μm size per milliliter are passed through the microchannel, and 7 sequences of images were recorded. The cells in the images were counted both by the proposed algorithm and carefully by eye. In order to show the importance of Gaussian filtering and morphological operations, the proposed algorithm applied without them. In Table 5.3, the numbers obtained by counting the cells by eye, by proposed algorithm, by proposed algorithm without Gaussian Filter and by the proposed algorithm without morphological operations are shown. In addition, wrong counts between the proposed algorithm and the eye count is shown in the table.

Results shows that 359 cells are counted in both methods. The desired pressure value of the pressure controller device is set to 2 mbar. The histogram of cell velocities is given in Figure 5.3 that demonstrates that cells have different speeds. Average speed of the cells are calculated as 4μm per millisecond.

Width of the cells are measured, and reliable results are classified as in Figure 5.4. Note that the distribution of cell width is more important for cell classification than actual widths. Even if cells are measured larger or smaller in different conditions, it

Table 5.3: Comparison of cell counts in 7 image sequences for 6-10μ case

Seq.	Prop. Alg.	Eye	Error	W/o Gaussian Filt.	W/o Morph. op.
1	91	91	0	94	120
2	64	64	0	64	84
3	64	64	0	63	87
4	23	23	0	23	36
5	50	50	0	45	72
6	45	45	0	50	77
7	22	22	0	22	33
Total	**359**	**359**	**0**	**361**	**509**

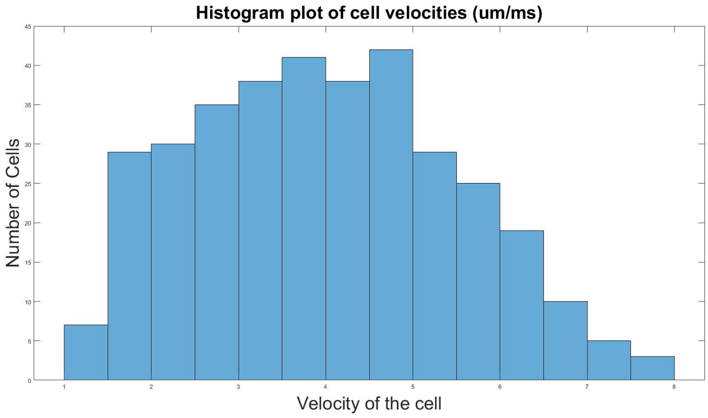


Figure 5.3: Speed plot of the cells in 7 image sequences

has no effect on classification. Classification results shows that there are two center point for the cell witdth that are 8 and 14.4. The ratio of the cells with 6μm size to the all counted cell is 66%.

5.2 Experiment 2

On the second day experiment, besides the classification of cells according to the number and size, the effect of applied fluid pressure was also observed. The solution containing 250.000 cells with 6μm size and 250.000 cells with 10μm size per milliliter are passed through the microchannel, and 10 sequences of images were

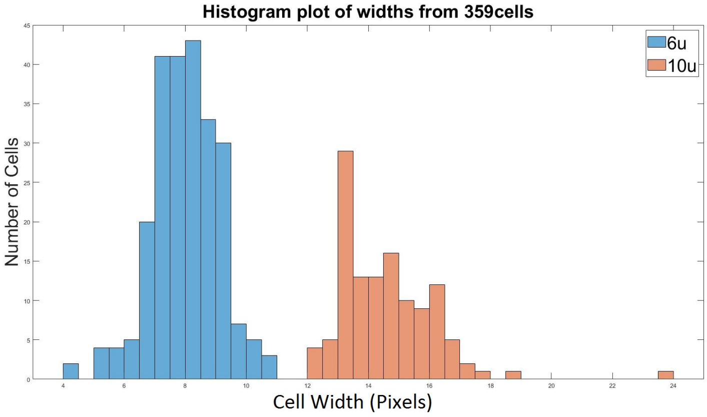


Figure 5.4: Histogram plot of the cell width for 7 image sequences

recorded. The cells in the images were counted both by the proposed algorithm and carefully by eye. In order to show the importance of Gaussian filtering and morphological operations, the proposed algorithm applied without them. In Table 5.4, the numbers obtained by counting the cells by eye, by proposed algorithm, by proposed algorithm without Gaussian Filter and by the proposed algorithm without morphological operations are shown. In addition, wrong counts between the proposed algorithm and the eye count is shown in the table.

Table 5.4: Comparison of cell counts in 10 image sequences for 6-10μ case

Seq.	Prop. Alg.	Eye	Error	W/o Gaussian Filt.	W/o Morph. op.
1	65	65	0	65	96
2	61	61	0	63	93
3	87	88	1	87	122
4	130	129	1	134	208
5	63	64	1	63	83
6	68	68	0	69	101
7	55	56	1	55	72
8	91	91	0	92	135
9	64	64	0	64	86
10	64	64	0	65	90
Total	**748**	**750**	**4**	**757**	**1086**

Results shows that 748 cells are counted by the algorithm although there were 750. However, error is not only 2. 4 cells are counted incorrectly. The desired pressure value of the pressure controller device is set to 2 mbar. The histogram of cell velocities is given in Figure 5.5 that demonstrates that cells have different speeds. Average speed of the cells are calculated as 4.3μm per millisecond. Width of the cells are measured and classified as in Figure 5.6.

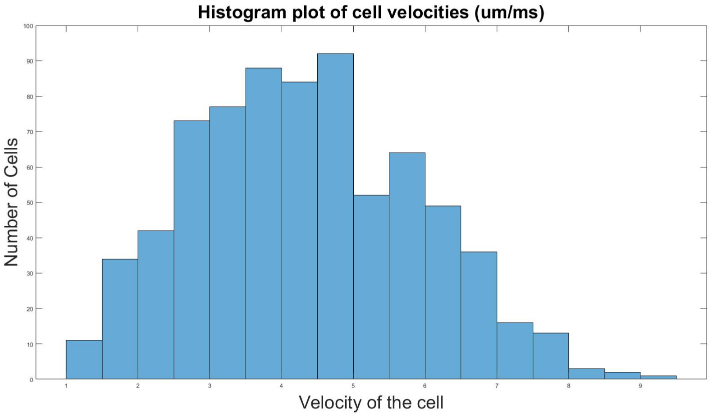


Figure 5.5: Speed plot of the cells for 10 image sequences

Width of the cells are measured, and reliable results are classified as in Figure 5.6. Classification results shows that there are two center point for the cell width that are 14.8 and 24.5. The ratio of the cells with 6μm size to the all counted cell is 71%. As explained before, the distribution of cell width is more important for cell classification than actual widths as shown in this case. These center points are different than the previous experiment results due to changing setup conditions such as intensity or angle of the fluorescent light source. Since the effect of this situation on the cells is similar, the width distributions did not change significantly.

Under the same conditions, desired pressure value is set to 6 mbar, and 12 sequences of images were recorded. The cells in the images were counted both by the proposed algorithm and carefully by eye. In order to show the importance of Gaussian filtering and morphological operations, the proposed algorithm applied without them. In Table 5.5, the numbers obtained by counting the cells by eye, by proposed algorithm, by proposed algorithm without Gaussian Filter and by the proposed algorithm without morphological operations are shown. In addition, wrong counts between the proposed

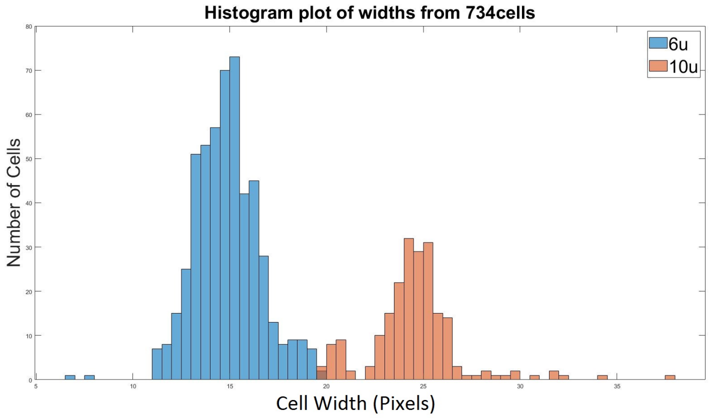


Figure 5.6: Histogram plot of the cell width for 10 image sequences

algorithm and the eye count is shown in the table.

Table 5.5: Comparison of cell counts in 12 image sequences for 6-10μ case

Seq.	Prop. Alg.	Eye	Error	W/o Gaussian Filt.	W/o Morph. op.
1	140	140	2	151	287
2	124	124	0	134	229
3	116	116	0	119	187
4	126	125	1	143	285
5	123	125	2	134	234
6	145	146	1	165	297
7	113	113	0	128	237
8	139	141	4	151	310
9	90	89	1	97	153
10	124	126	2	132	282
11	136	136	4	154	314
12	111	110	1	115	200
Total	**1487**	**1491**	**18**	**1623**	**3015**

Results shows that 1487 cells are counted by the algorithm although there were 1491. However, error is not only 4. 18 cells are counted incorrectly. Since the desired pressure value of the pressure controller device is set to 6 mbar, speed of the cells are measured with larger values. The histogram of cell velocities is given in Figure

5.5 that demonstrates that cells have different speeds. Average speed of the cells are calculated as 7.3μm per millisecond.

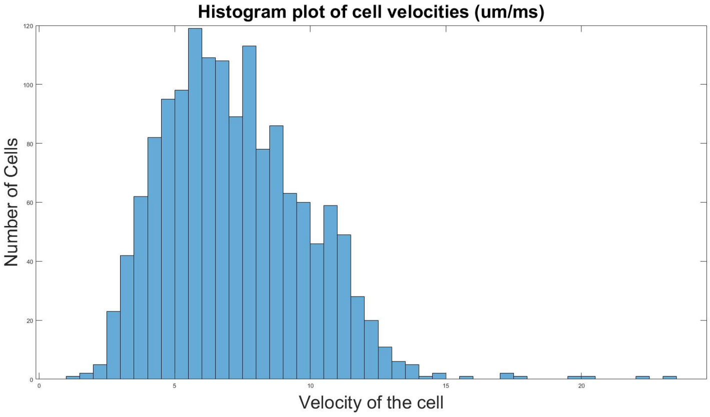


Figure 5.7: Speed plot of the cells for 12 image sequences under 6 mbar pressure

Width of the cells are measured, and reliable results are classified as in Figure 5.8. Classification results shows that there are two center point for the cell width that are 12.8 and 23.4. The ratio of the cells with 6μm size to the all counted cell is 69%.

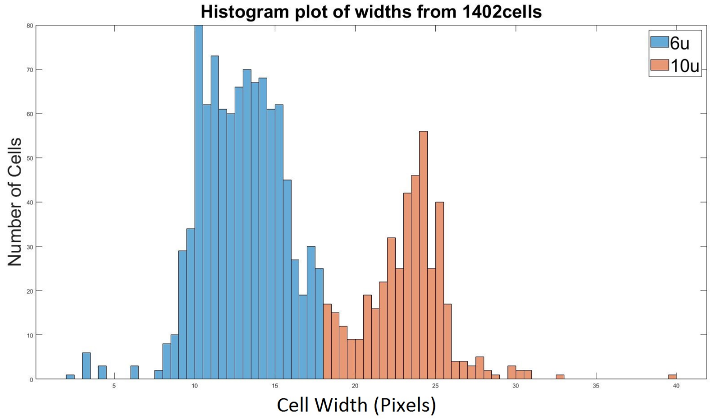


Figure 5.8: Histogram plot of the cell width for 12 image sequences

5.3 Computation Analysis

The computation part of the sub parts of the algorithm is measured as in Figure 5.9. As seen in the Figure 5.9, most of the computation time is consumed on morphological operations. In summary, 75% of the total time is used for morphological operations, 9% is used for reading frames from HIS file, 8% is used for cell segmentation, 4% is used for Gaussian filtering for noise elimination, 3% is used for connected component labeling, 1% is used for tracking algorithms, and almost 0% is used for clustering part.

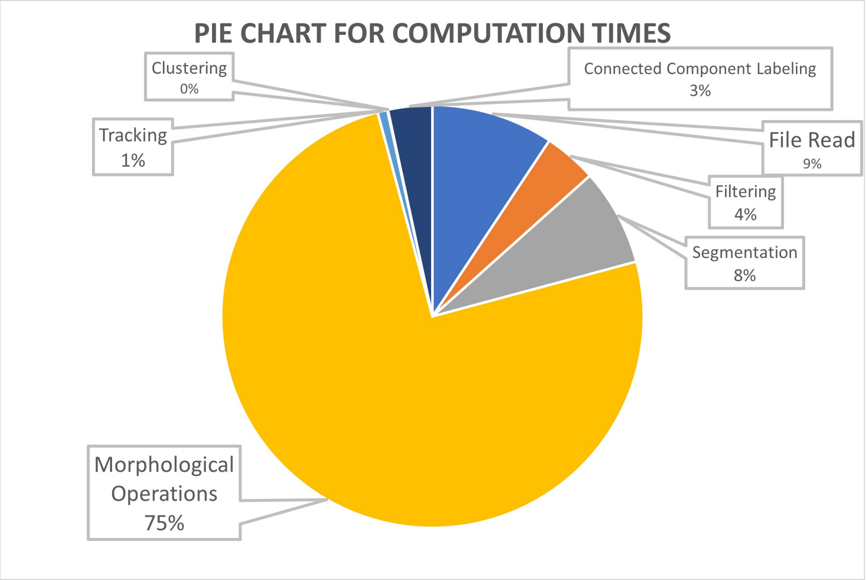


Figure 5.9: Pie chart for the computation time of each part of the proposed algorithm

The total computation time of the algorithm is measured from CPU time. Table 5.6 and 5.7 that are located at the end of the chapter shows the measured computation time of the proposed algorithm in seconds for experiment 1 and 2, respectively. The average computation time for 5974 frames is 350 seconds, and average processing time is measured as 64.4 msec for a frame.

5.4 Comments on Results

Test results show that cell counting is done successfully for all cases. Width measurements are classified by using the distribution of the widths. Although a comparison could not be made with a reference result in the classification of cells according to their width, gathering the measurements around 2 main values and classifying them according to their distribution gives logical results. It was observed that the applied fluid pressure had a significant effect on the velocity of the cells, and the increase in cell velocity caused more errors in cell counting. When cells move faster, the number of frames in which cells are detected and tracked decreases and the cells appear longer. The change on the location of the detections are larger and affected by the noise more. Further increase in fluid pressure may result in inaccurate counting of cells. As a result, the test results show that the algorithm counts the cells correctly and make a valid classification according to sizes if the liquid pressure is less than or equal to 6 mbar.

The proposed algorithm includes many measures such as Gaussian filter, morphological operations, and minimum area in connected component labeling for accurate detection. When the Gaussian filter is removed and the results are obtained again, it is seen that the wrong cell count increases. In addition, the measured cell widths increase. However, the number of errors that occur is not as great as in the removal of morphological operations. When morphological operations are removed, the error in the number of cells increases a lot. In addition, in this case, the confidence flag marked in the cell width is also lit very few times. The minimum area condition of the connected component labeling algorithm prevented thousands of false counts in the absence of morphological operations. Because without morphological operations, faulty segmentations cannot be cleaned and special problems such as electrode area are not solved. Although the most of the computation time is consumed in the morphological operations, their contribution to the proposed algorithm is vital.

Table 5.6: Computation time for experiment 1

Experiment Id	Proposed Algorithm Computation Time (seconds)
Exp 1-1-1	375
Exp 1-1-2	358
Exp 1-1-3	352
Exp 1-1-4	347
Exp 1-1-5	339
Exp 1-1-6	337
Exp 1-1-7	336
Exp 1-2-1	335
Exp 1-2-2	346
Exp 1-2-3	334
Exp 1-2-4	335
Exp 1-2-5	341
Exp 1-2-6	334
Exp 1-2-7	337
Exp 1-3-1	338
Exp 1-3-2	337
Exp 1-3-3	339
Exp 1-3-4	337
Exp 1-3-5	338
Exp 1-3-6	336
Exp 1-3-7	337

Table 5.7: Computation time for experiment 2

Experiment Id	Proposed Algorithm Computation Time (seconds)
Exp 2-1-1	398
Exp 2-1-2	373
Exp 2-1-3	367
Exp 2-1-4	355
Exp 2-1-5	350
Exp 2-1-6	350
Exp 2-1-7	344
Exp 2-1-8	346
Exp 2-1-9	350
Exp 2-1-10	364
Exp 2-2-1	366
Exp 2-2-2	363
Exp 2-2-3	350
Exp 2-2-4	378
Exp 2-2-5	368
Exp 2-2-6	365
Exp 2-2-7	362
Exp 2-2-8	357
Exp 2-2-9	348
Exp 2-2-10	349
Exp 2-2-11	340
Exp 2-2-12	334

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Cell counting is a method used in many areas of cell experiments. Various methods have been developed in order to count the cells correctly and to extract as much data about the cells as possible. Firstly, the cells counted by manual methods when the cells are stationary. Then, cells could be analyzed one by one by passing them through a microchannel with the contributions of microfluidic technology. In this thesis, it is aimed to count the cells while they are passing through the microchannel, to measure their size and velocity, and to group according to cell size if more than one cell type is desired to be counted. For this purpose, while the stained cells were passed through the microchannel, they are illuminated with fluorescent light, and microscope images are taken recorded at high frame rate with a camera. With the proposed method, cells are automatically detected by using image processing techniques, and then accurate counting was attempted by tracking them along the microchannel with the object tracking algorithm.

The proposed algorithm consists of three main topics. The first topic is the automatic detection of cells. This is achieved by using image processing algorithms such as Gaussian filtering for noise elimination, background modeling and segmentation for cell detection, morphological operations for removing segmentation faults and eliminating undesired situations caused by the setup conditions, and finally connected component labeling method for labeling each detection separately with a centroid and bounding box. The second topic is tracking of the cells along the microchannel. This is achieved by an object tracking method. Kalman filter is used to obtain next position estimates of the detections, and estimations and detections are matched by using Hungarian algorithm for the optimal solution. Finally, width and speed information is obtained by using related measurements. For cases where multiple sizes of cells are used, width measurements are grouped by K-medoids clustering algorithms.

In order to test the proposed method, experiments were carried out on the setup in the Mikro Biyosistemler company, which is currently used for other researches. According to experimental results, the proposed method successfully counted the cells and made clustering. The proposed method can be used in many different projects and places, as it requires equipment such as microscopes and cameras that generally exist in cell research laboratories. If the proposed method is compared with other methods, it can be said that the proposed method is faster and more accurate than manual counting methods, and it is ahead in terms of obtaining statistical information about cell sizes. Compared to EIS method, the proposed method cannot give that much detailed information about cells. However, EIS method requires a more complex method and more expensive equipment to provide detailed information. As a result, the proposed algorithm can be used for accurate and automated cell counting if the frame rate of the camera is high enough for the cell speed, background is stable, camera is stationary, and the motion of cells in the microchannel is modelled by a linear motion model. Differential counting of cells having different sizes is also can be done by using the proposed algorithm. Moreover, the proposed method can be used directly in cell counting as well as as a reference value for many cell counting studies under development.

The proposed method uses recorded images for now that is it is working offline. As future work, the camera images are directly taken from the serial interface of the camera, and processed online. By this way, total computation time can be decreased as the amount of file read time. In the proposed method, closing is used to eliminate a setup problem. By designing an appropriate microchannel for the proposed method, total process time can be decreased as the amount of closing operation. Velocity measurement can be made with the proposed method, but it is only used to control the flow rate. As future work, a new axis can be added for the grouping of different cells by determining the effect of cell size on velocity by considering microfluidic dynamics. The proposed method counts cells moving in groups along the microchannel as a single cell. As future work, instead of counting connected cells as a single cell, it is aimed to count cells separately by using detailed shape analysis.

REFERENCES

[1] S. Zhang and J. Kuhn, “Cell isolation and culture,” *WormBook : The online review of C. elegans biology*, pp. 1–39, 2013.

[2] RWD Life Science Inc., *C100 Automated Cell Counter*, 2019.

[3] T. Vicar, J. Balvan, J. Jaros, F. Jug, R. Kolar, M. Masarik, and J. Gumulec, “Cell segmentation methods for label-free contrast microscopy: Review and comprehensive comparison,” *BMC Bioinformatics*, vol. 20, 2019.

[4] A. Doroszkowski, “7 - particle size and size measurement,” in *Paint and Surface Coatings (Second Edition)* (R. Lambourne and T. Strivens, eds.), Woodhead Publishing Series in Metals and Surface Engineering, pp. 243–285, Woodhead Publishing, second edition ed., 1999.

[5] F. Gökçe, P. Ravaynia, M. Modena, and A. Hierlemann, “What is the future of electrical impedance spectroscopy in flow cytometry?,” *Biomicrofluidics*, vol. 15, p. 061302, 2021.

[6] M. K. Aslan, “Lens free imaging of dep manipulated cancer cells,” Master’s thesis, Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, 2017.

[7] C. Bertelsen, J. Franco, G. Skands, M. Dimaki, and W. Svendsen, “Investigating the use of impedance flow cytometry for classifying the viability state of e. coli,” *Sensors*, vol. 20, p. 6339, 2020.

[8] R. Srisha and A. Khan, “Morphological operations for image processing : Understanding and its applications,” in *National Conference on VLSI, Signal processing Communications*, pp. 17–19, 2013.

[9] E. Rani, “Kalman filter based multiple object tracking system,” *International Journal of Electronics, Communication Instrumentation Engineering Research and Development*, vol. 8, pp. 1–6, 2018.

[10] Hamamatsu Photonics, *ORCA-Flash4.0 LT+ Digital CMOS Camera C11440-42U30 / C11440-42U31 Instruction manual*, 2018.

[11] K. Ongena, C. Das, J. L. Smith, S. Gil, and G. Johnston, “Determining cell number during cell culture using the scepter cell counter.,” *Journal of visualized experiments : JoVE*, vol. 45, 2010.

[12] Logos Biosystems, *How to Count Cells, An Overview of Cell Counting Methods*, 2018.

[13] M. Absher, “Chapter 1 - hemocytometer counting,” in *Tissue Culture* (P. F. KRUSE and M. PATTERSON, eds.), pp. 395–397, Academic Press, 1973.

[14] H. A. Khan and G. Maruf, “Counting clustered cells using distance mapping,” in *2013 International Conference on Informatics, Electronics and Vision, ICIEV 2013*, pp. 1–6, 2013.

[15] M. G. Forero and A. Hidalgo, “Image processing methods for automatic cell counting in vivo or in situ using 3d confocal microscopy,” in *Advanced Biomedical Engineering* (G. D. Gargiulo and A. McEwan, eds.), ch. 10, Rijeka: IntechOpen, 2011.

[16] T. Komoda and T. Matsunaga, “Chapter 6 - biotechnological study,” in *Biochemistry for Medical Professionals* (T. Komoda and T. Matsunaga, eds.), pp. 75–92, Boston: Academic Press, 2015.

[17] K. Cheung, S. Gawad, and P. Renaud, “Impedance spectroscopy flow cytometry: On-chip label-free cell differentiation,” *Cytometry Part A*, vol. 65A, no. 2, pp. 124–132, 2005.

[18] Z. Xi, D. Xu, W. Song, and Y. Zheng, “A* algorithm with dynamic weights for multiple object tracking in video sequence,” *Optik*, vol. 126, no. 20, pp. 2500–2507, 2015.

[19] B. Deori and D. Thounaojam, “A survey on moving object tracking in video,” *International Journal on Information Theory*, vol. 3, pp. 31–46, 2014.

[20] L. Fan, Z. Wang, B. Cail, C. Tao, Z. Zhang, Y. Wang, S. Li, F. Huang, S. Fu, and F. Zhang, “A survey on multiple object tracking algorithm,” in *2016 IEEE In-*

ternational Conference on Information and Automation (ICIA), pp. 1855–1862, 2016.

[21] M. Jogin, M. Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (cnn) and deep learning," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 2319–2323, 2018.

[22] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, 2017.

[23] S. M. Ågren, "Object tracking methods and their areas of application: A meta-analysis : A thorough review and summary of commonly used object tracking methods," Master's thesis, Faculty of Science and Technology, Department of Computing Science, Umeå University, Umeå, 2017.

[24] M. Bigas, E. Cabruja, J. Forest, and J. Salvi, "Review of cmos image sensors," *Microelectronics Journal*, vol. 37, no. 5, pp. 433–451, 2006.

[25] L. Cadena, A. Zotin, F. Cadena, A. Korneeva, A. Legalov, and B. Morales, "Noise reduction techniques for processing of medical images," *World Congress on Engineering*, vol. 1, no. 5, pp. 496–500, 2017.

[26] A. Ravishankar, S. Anusha, H. K. Akshatha, A. Raj, S. Jahnavi, and J. Madhura, "A survey on noise reduction techniques in medical images," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, pp. 385–389, 2017.

[27] N. Kumar and M. Nachamai, "Noise removal and filtering techniques used in medical images," *Oriental Journal of Computer Science and Technology*, vol. 10, no. 1, pp. 103–113, 2017.

[28] L. Unzueta, M. Nieto, A. Cortes, J. Barandiaran, O. Otaegui, and P. Sanchez, "Adaptive multi-cue background subtraction for robust vehicle counting and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 527–540, 2012.

[29] S. V. Mashak, B. Hosseini, M. Mokji, and S. A. R. Abu-Bakar, “Background subtraction for object detection under varying environments,” in *2010 International Conference of Soft Computing and Pattern Recognition*, pp. 123–126, 2010.

[30] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, “Efficient moving object segmentation algorithm using background registration technique,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577–586, 2002.

[31] Y. Xu, J. Dong, B. Zhang, and D. Xu, “Background modeling methods in video analysis: A review and comparative evaluation,” *CAAI Transactions on Intelligence Technology*, vol. 1, pp. 43–60, 2016.

[32] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11-12, pp. 31–66, 2014.

[33] T. Bouwmans, “Recent advanced statistical background modeling for foreground detection - a systematic survey,” *Recent Patents on Computer Science,*, vol. 4, no. 3, pp. 147–176, 2011.

[34] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, “Extracting roadway background image: Mode-based approach,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1944, pp. 82–88, 2006.

[35] J. Kuruvilla, D. Sukumaran, A. Sankar, and S. P. Joy, “A review on image processing and image segmentation,” in *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pp. 198–203, 2016.

[36] N. M. Zaitoun and M. J. Aqel, “Survey on image segmentation techniques,” *Procedia Computer Science*, vol. 65, pp. 797–806, 2015.

[37] A. R. Weeks, *Fundamentals of Electronic Image Processing*. NJ: SPIE Press, 1996.

[38] L. Sartor and A. Weeks, “Morphological operations on color images,” *J. Electronic Imaging*, vol. 10, pp. 548–559, 2001.

[39] M. Goyal, "Morphological image processing," *IJCST*, vol. 2, no. 4, pp. 161–165, 2011.

[40] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25–43, 2017.

[41] N. Y. Ko and T. G. Kim, "Comparison of kalman filter and particle filter used for localization of an underwater vehicle," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 350–352, 2012.

[42] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[43] S. S. Pathan, A. Al-Hamadi, and B. Michaelis, "Intelligent feature-guided multi-object tracking using kalman filter," in *2009 2nd International Conference on Computer, Control and Communication*, pp. 1–6, 2009.

[44] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using kalman filter," in *The 2010 IEEE International Conference on Information and Automation*, pp. 1862–1866, 2010.

[45] D. Hernández, J. M. Cecília, C. T. Calafate, J.-C. Cano, and P. Manzoni, "The kuhn-munkres algorithm for efficient vertical takeoff of uav swarms," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–5, 2021.

[46] H. Ouchra and A. Belangour, "Satellite image classification methods and techniques: A survey," in *2021 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6, 2021.

[47] A. W. Abbas, N. Minallh, N. Ahmad, S. A. R. Abid, and M. A. A. Khan, "K-means and isodata clustering algorithms for landcover classification using remote sensing," *Sindh University Research Journal-SURJ (Science Series)*, vol. 48, no. 2, 2016.

[48] R. Xu and D. C. Wunsch, "Clustering algorithms in biomedical research: A review," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 120–154, 2010.

[49] E. F. Sirat, B. D. Setiawan, and F. Ramdani, "Comparative analysis of k-means and isodata algorithms for clustering of fire point data in sumatra region," in *2018 4th International Symposium on Geoinformatics (ISyG)*, pp. 1–6, 2018.

[50] J. Oyelade, I. Isewon, O. Oladipupo, O. Emebo, Z. Omogbadegun, O. Aromolaran, E. Uwoghiren, D. Olaniyan, and O. Olawole, "Data clustering: Algorithms and its applications," in *2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, pp. 71–81, 2019.

[51] T. Velmurugan and T. Santhanam, "Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points," *Journal of computer science*, vol. 6, no. 3, p. 363, 2010.

[52] D. Cao and B. Yang, "An improved k-medoids clustering algorithm," in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, vol. 3, pp. 132–135, 2010.

[53] J. R. Janesick, *Scientific Charge-Coupled Devices*. Washinton: SPIE Press, 2001.

[54] W. G. J. Jerome, "Practical guide to choosing a microscope camera," *Microscopy Today*, vol. 25, no. 5, pp. 24–29, 2017.

[55] J. R. Joubert, "How to choose between a ccd and scmos scientific-grade camera." 2017.

[56] R. D. Gow, D. Renshaw, K. Findlater, L. Grant, S. J. McLeod, J. Hart, and R. L. Nicol, "A comprehensive tool for modeling cmos image-sensor-noise performance," *IEEE Transactions on Electron Devices*, vol. 54, no. 6, pp. 1321–1329, 2007.

[57] Q. Gu, T. Aoyama, T. Takaki, and I. Ishii, "Simultaneous vision-based shape and motion analysis of cells fast-flowing in a microchannel," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 204–215, 2015.

[58] B. C. Arvind, S. K. Nagaraj, C. S. Seelamantula, and S. S. Gorthi, "Active-disc-based kalman filter technique for tracking of blood cells in microfluidic channels," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3394–3398, 2016.