Cell Phenotype Analyzer: Automated Techniques for Cell Phenotyping using Contactless Dielectrophoresis

Divya Chandrakant Bala

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering

A. Lynn Abbott, Chair
Devi Parikh
Eva Schmelz

April 28, 2016
Blacksburg, Virginia

Keywords: Cell Phenotype, Computer Vision, Dielectrophoresis
Copyright 2016, Divya Chandrakant Bala
ABSTRACT

Cancer is among the leading causes of death worldwide. In 2012, there were 14 million new cases and 8.2 million cancer-related deaths worldwide. The number of new cancer cases is expected rise to 22 million within the next two decades. Most chronic cancers cannot be cured. However, if the precise cancer cell type is diagnosed at an earlier, less aggressive stage then the chance of curing the disease increases with accurate drug delivery. This work is a humble contribution to the advancement of cancer research. This work delves into biological cell phenotyping under a dielectrophoresis setup using computer vision. Dielectrophoresis is a well-known phenomenon in which dielectric particles are subjected to a non-homogeneous electric field. This work is an analytical part of a larger proposed system replete with hardware, software and microfluidics integration to achieve cancer cell characterization, separation and enrichment using contactless dielectrophoresis. To analyze the cell morphology, various detection and tracking algorithms have been implemented and tested on a diverse dataset comprising cell-separation video sequences. Other related applications like cell-counting and cell-proximity detection have also been implemented. Performances were evaluated against ground truth using metrics like precision, recall and RMS cell-count error. A detection approach using difference of Gaussian and super-pixel algorithm gave the highest average F-measure of 0.745. A nearest neighbor tracker and Kalman tracking method gave the best overall tracking performance with an average F-measure of 0.95. This combination of detection and tracking methods proved to be best suited for this dataset. A graphical user interface to automate the experimentation process of the proposed system was also designed.
GENERAL AUDIENCE ABSTRACT

This work is a contribution to cancer research and can be used as a verification tool as well as a stand-alone application for analyzing cancer cell attributes. Different biological cells react differently in the presence of strong electric fields owing to their biophysical properties. Using this knowledge, isolation of cancer cells from healthy cells, amongst other applications, becomes possible. This isolation idea forms the basis for this work. With the advent of computer vision, visual traits of biological cells can be measured. In this work, videos capturing cell separation using the aforementioned isolation idea have been used for implementing the vision algorithms. The attributes measured were as follows: the number of different types of cells under experimentation, their sizes which included width, height and radii, track history of each cell, proximity between these cells. The algorithms used to determine these have been evaluated too in this work. Software to automate the processes on the isolation platform has also been implemented. An advantage of automated attribute measurement is reduction in manual efforts and better information extraction and analysis. The nature of this thesis allows it to be used for the study of other biological cells too, thus not limiting it to cancer cell research.
Acknowledgments

What started as a summer internship at a budding startup has culminated into this thesis. I would like to thank phenoCHIP LLC, a phenotype-based cell hierarchy and isolation platform company, for giving me the opportunity to work with them. I am grateful for the relentless support that they have extended towards me over the past one year. A special thanks to Mr. Elliot McAllister, general manager of phenoCHIP LLC, for educating me about dielectrophoresis, for all the candid brain-storming sessions and for showing keenness towards my work. I take this opportunity to thank my advisor Dr. Lynn Abbott for his guidance throughout this work. I thank him for his help in writing this thesis. I would like to thank Dr. Devi Parikh. She has been an inspiration to me. I thank her for motivating her students to apply the theoretical knowledge they gain to significant practical applications. I would like to thank Dr. Rafael Davalos, Jaka Cemazar and the Bioelectromechanical Systems Lab for the dataset. I would like to thank Dr. Eva Schmelz for her inputs as well as the enthusiasm that she has always showed for the use of computer vision in studying cancer cells.
4.5 Agilent function generator ........................................ 14
4.6 Graphical User Interface ........................................ 14

5 Detection .......................................................... 17
  5.1 Introduction .................................................... 17
  5.2 Connected component analysis .............................. 17
  5.3 Difference of Gaussians ....................................... 18
  5.4 Simple Linear Iterative Clustering (SLIC) ............... 19

6 Tracking .......................................................... 21
  6.1 Introduction .................................................... 21
  6.2 Nearest neighbor tracker ..................................... 21
  6.3 Kalman filter .................................................. 22
  6.4 Optical flow .................................................. 22

7 Cell Counting ..................................................... 24
  7.1 Introduction .................................................... 24
  7.2 Approach ..................................................... 25

8 Proximity Detection ................................................. 26
  8.1 Introduction .................................................... 26
  8.2 Approach ..................................................... 27

9 Dataset and Annotation ............................................. 28
  9.1 Introduction .................................................... 28
  9.2 VATIC ......................................................... 28
  9.3 MATLAB interface ........................................... 29

10 Quantitative Evaluation ........................................... 34
  10.1 Detection metrics ............................................ 34
  10.2 Tracking metrics ............................................. 36
10.3 Cell count metric ................................................................. 37

11 Qualitative Evaluation ........................................................... 39

11.1 Detection .......................................................... 39
11.2 Tracking .......................................................... 42
11.3 Proximity detection .................................................. 45
11.4 Cell characterization .................................................. 47

12 Conclusions ................................................................. 48

Bibliography ................................................................. 49
List of Figures

1.1 The process of dielectrophoretic trapping. (a)-(c) highlight images taken by a video microscope (field of view is 1.3mm x 0.6mm). (a) Two cell sub-populations, in this case the red and green cells, are flowing with the AC electric field switched off. (b) The electric field is turned on. The green cells (cells of interest) are attracted to the posts and are trapped. Posts are insulated micro-structures used to focus the AC electric field. The liquid buffer which is pumped continuously washes the untrapped red cells out. (c) The field is turned off. The trapped green cells are now released and washed away by the buffer [1] Bioelectromechanical Systems Laboratory - Virginia Tech. [Online]. Available: http://www.sbes.vt.edu/davalos/, Used under fair use, 2016. ................................................................. 4

3.1 Dielectric particle in non-homogeneous or point-plane electric field [19] M. P. Hughes, “AC Electrokinetics: Applications for Nanotechnology,” Used under fair use, 2016. ................................................................. 8

3.2 Plot of real and imaginary parts of Clausius-Mossotti factor. The solid lines indicate the real part and the dotted indicate the imaginary part. The different colors are indicative of different media conductivity (from [20] O. Castillo-Fernandez, N. Uria, F. X. Munsoz, and A. Bratov, “Cell concentration systems for enhanced biosensor sensitivity,” in Nanotechnology and Nanomaterials - ”Biosensors - Micro and Nanoscale Applications”, 2015, ch. 7, Used under fair use, 2016). ................................................................. 10

4.1 The cDEP device and associated hardware-software. This is an illustration of the proposed experimental setup (from [27] https://www.youtube.com/watch?v=qVmjcfyBocg, The illustration indicating the cDEP device is the work of Dr. R. Davalos’s lab, Used under fair use, 2016). ................................................................. 12


4.9 The GUI. The Browse button under uProcess Settings is used to run the script file. Other options can be used to control respective hardware controls like SVM, function generator and amplifier. .................................................. 16

5.1 An illustration showing difference of Gaussian filter. The filter helps to reduce high frequency components [30] [Online]. Available: http://www.olympusmicro.com/primer/java/digitalimaging/processing/diffgaussians/, Used under fair use, 2016. .................................................. 20


9.1 (a)-(d) one frame from each of the dataset videos. .................................................. 30
9.2 (a), (b) show two consecutive frames being annotated in VATIC. After annotating the first frame it was expected that the next frame would be well-annotated. However manual annotations had to be repeated for every frame as the cells were not followed close enough from the previous frame. In many scenarios the moving/resizing was minute but given the number of objects of interest, it was a tedious task.

9.3 The MATLAB UI to facilitate creation of ground truth boxes.

9.4 (a)-(c) highlight the scenarios where the detection algorithm may not perform well, requiring resizing/adding of new bounding boxes.

9.5 (a) shows the output of the detection algorithm overlaid on the frame using the detection MATLAB interface. (b) shows the ground truth boxes overlaid on the frame. These were obtained using the MATLAB UI functionality. As is seen, the ground truth boxes are more closely-fitted to the cell boundaries than the detection.

9.6 Re-labelling of labelled ground truth boxes for precise track-correspondence using tracking MATLAB UI.

11.1 Results for connected components detection.

11.2 Results for difference of Gaussian detection.

11.3 Results for DoG + SLIC detection.

11.5 Tracking results for Video 1.
List of Tables

10.1 Average precision and recall values for different videos in the dataset (Fig. 9.1) for different detection algorithms. ................................................. 35
10.2 Average precision and recall values for tracking algorithms using ground-truth detection boxes. ................................................................. 36
10.3 Average precision and recall values for tracking algorithms using DoG + SLIC detection boxes. ................................................................. 37
10.4 RMS cell-count error values for different videos in the dataset for different detection algorithms. ................................................................. 38
Chapter 1

Introduction

1.1 Motivation

Cell phenotyping is a technique used in molecular biology for classification of cells based on distinctive physical and behavioral traits of the organisms. Example traits are size, shape, metabolic activities, or patterns of movement. Cell phenotyping differs from cell genotyping, which relies on DNA for classification. Studying cancer stem cells is a highly active research area, where the answers may lie in tumor type and phenotype. A tumor is a heterogeneous population of mutant cells, all of which share some mutations but vary in specific phenotype. This interesting facet can be explored where different tumor cells would share same genotype, but different phenotypes.

Phenotyping in cancer treatment can be used for targeted therapy. Conventional chemotherapy destroys healthy as well as tumor cells. A lot of time is lost in diagnosing the patient’s exact Tumor Initiating Cell (TIC) type and prescribing the apposite drug combination to treat it. By using a phenotype profiling platform, the patient’s exact TIC type can be determined at a primitive stage. This is made possible by taking sample of blood or fluid from the patient and analysing it to identify the cancer cell based on phenotype. Efficient drug delivery would follow, leading to minimal harm to patient.

Another motivation for using phenotyping is to develop a marker-less cell characterization system. Biomarkers or biological markers are traceable substances that are introduced in an organism to monitor changes in terms of health or organ function. These biomarkers indicate change in the state of proteins that are closely tied to the progression of a disease or changes after drug treatments. Cellular biomarkers are useful to isolate, sort and quantify cells. Since most biomarkers are fluorescent, differentiating cells on the basis of color becomes easy. However there are certain cells whose antigens or surface proteins do not attract biomarkers. In such scenarios, phenotyping can prove to be a non-invasive (depending on how the cell sample is obtained) as well as an efficient method to sort cells. The reason is
that phenotyping does not depend on the surface antigens but rather associates with a cell’s biophysical properties.

1.2 Dielectrophoresis

Dielectrophoresis (DEP) is a phenomenon in which a force is exerted on a dielectric particle when the particle is subjected to an inhomogeneous electric field. DEP does not require the particle to be charged. In the presence of electric fields, all particles exhibit dielectrophoretic activity. Particles can thus be manipulated. Fig. 1.1 illustrates the process of cell trapping using dielectrophoresis.

A limitation of traditional DEP is that uninsulated metal electrodes are typically used. In contrast, contactless dielectrophoresis (cDEP) introduces fluid electrode channels alongside cell trapping locations [2], thereby avoiding artifacts that could arise from the use of metal electrodes. For example, gas evolution due to electrolysis at metal DEP electrodes. A possible application of dielectrophoretic devices is the continuous separation of rare cells of interest from dilute blood samples [3]. This kind of isolation is very useful in cancer cell profiling, drug development, analysis of infected cells, diagnostics, and related activities.

1.3 Contributions of this work

The objective of this work is to analyze cancer cell phenotypes under cDEP using computer vision algorithms. The processing involves cell detection using image segmentation, connected components analysis, difference of Gaussian filtering, super-pixel algorithm, and cell tracking using nearest neighbor approach, Kalman filter and optical flow techniques. The main contributions of this work are:

1) Automation of cDEP experiments by integrating and controlling the hardware and software using a graphical user interface
3) An algorithm and software that can perform cell counting, cell proximity detection and pearl chaining identification for tumor sub-populations
4) Evaluation of the different methods quantitatively and qualitatively.

1.4 Organization of thesis

This work commences with an overview of past efforts in the field of cell detection and motion analysis in cDEP through the chapter on related work. Following this is an informative chapter on dielectrophoresis that sheds light on the phenomenon for which this work provides imaging analysis tool. In subsequent chapters the experimental setup of the proposed
cDEP system and details about the dataset and annotation are discussed. The chapters on detection and tracking give a detailed description of the approach adopted to tackle the problem statement. Next two chapters on cell counting and proximity detection are the applications that follow up detection and tracking. Finally there are chapters on quantitative and qualitative evaluation of the various algorithms implemented in this work. The chapter on conclusions helps to succinctly list down the inferences drawn from this work.
Figure 1.1: The process of dielectrophoretic trapping. (a)-(c) highlight images taken by a video microscope (field of view is 1.3mm x 0.6mm). (a) Two cell sub-populations, in this case the red and green cells, are flowing with the AC electric field switched off. (b) The electric field is turned on. The green cells (cells of interest) are attracted to the posts and are trapped. Posts are insulated micro-structures used to focus the AC electric field. The liquid buffer which is pumped continuously washes the untrapped red cells out. (c) The field is turned off. The trapped green cells are now released and washed away by the buffer [1] Bioelectromechanical Systems Laboratory - Virginia Tech. [Online]. Available: http://www.sbes.vt.edu/davalos/, Used under fair use, 2016.
Chapter 2

Related work

Motion analysis is a trending research topic today in Computer Vision with applications such as augmented reality, animation, artificial intelligence robots, medical diagnosis and surveillance. Object detection and gathering pertinent attributes about the moving object are the two aspects of motion analysis. Using several processing techniques one can determine many features related to the objects in motion like velocity, acceleration, trajectory, change in shape and size. In this work, the word ”object” or ”entity” will relate to biological cells.

In microfluidics and lab-on-a-chip technology that deals with cell separation and cell characterization, the work of motion analysis holds a very important place. Most of the work done in this area deals with microscopy imaging and time-lapse image sequences. Cells in general are very small, in the range of microns. To study them, adequately powered microscopes are needed. Resolution can be compromised. In other cases, the structural study or study of cell surfaces maybe key in a research leading to use of high powered microscopes giving limited field of view. There have been many advances in microscopic imaging equipment, like using fluorescent probes or feedback-driven high powered microscope optics or high sensitivity electronic detectors.

2.1 Camera-object positioning

Motion analysis can be used in two scenarios. In the first one, the stationary object under consideration undergoes morphological changes with a fixed camera position. For instance, in his dissertation, Crane created a microfluidic platform for high-resolution sorting and imaging of C. elegans [5]. For the computer vision part he implemented a 3-stage framework which consisted of upfront image processing, followed by feature extraction using filters adapted from SIFT features and finally classification based on RBF-kernel linear support vector machine. His work was mostly associated with in-focus high-resolution images.
In the second scenario, either the camera is fixed and the object is moving or the camera is moving and the object is fixed or both are moving. There has been a lot of work in this category. Lisa Mae Sanders in her thesis [6], developed two automated whole cell tracking approaches. The first one used Hough transform for cell attribute detection and nearest Euclidean distance for mapping a cell location between frames. The second cell tracking approach made use of active contours as extension for cell circularity measurement and Kalman filter for cell tracking. A Vicell machine (Beckman coulter) was used to perform a baseline measurement prior to the video being taken in both cases. The accuracy measurements in terms of tracking was 80.6% using the nearest neighbor approach and 93% using Kalman filter respectively. Another study [7] presents how the discrete Fourier transform can be used to quantify patterns related to movements, and how multidimensional scaling techniques can test differences in them.

Most common blood cell tracking approaches make use of spatio-temporal image analysis [8]. But this is bound by limitations like the velocity of moving cells should be fixed or the path should be stable. In [9], the authors present an optical flow method as well as a SIFT flow method for cell tracking.

### 2.2 Analysis approach

Any kind of motion analysis approach can be classified into three distinct areas: feature-based, gradient-based and correlation-based. An example of gradient-based classification is the optical flow approach. Zhang et al. [10] present an optical flow based automatic cell motion tracking algorithm. This algorithm helps estimate the 3D measurements of cell motion like rotation and translation. Optically-induced dielectrophoresis does not require complex geometric electrodes and fabrication. Also the novel part about the experimental set-up that facilitates the vision algorithm is the optoelectronic tweezers which helps measure cell self-rotation where the high-speed camera with 40x lens is placed at a lateral position.

An example of a correlation-based approach would be the work presented in the paper [11]. The approach comprises correlation matching between object region in the current frame and the next frame. This is done by convolving the intensity template over contender regions in the next frame. This overall technique called hierarchical adaptive merge/split structured mesh (HAMSMS) approach is compared with the simple frame difference algorithm and dense motion field algorithm in the paper.

There have been some innovative methods employed for solving the image analysis problems in dielectrophoresis video sequences too. In [12], the solution proposed comprises particle detection based on advanced image-segmentation and feature-aided multi-target tracking paradigm multiple-hypothesis tracking. The improved state space includes both target kinematics and detection-level features such as size, shape and color of particle. Multiple hypothesis tracking is essentially a breadth first search algorithm and thus computationally
expensive as compared to simpler tracking approaches. It is used for scenarios where the cells are closely spaced and cross paths. In this work, a simpler tracking approach on the lines of single-target tracking has been implemented because the cells have a constrained linear flow in the dataset. Information about point of entry and exit has been leveraged.

### 2.3 Software packages

Many commercially available software products help in extracting tons of information automatically. Jaber et al. [13] made use of MATLAB’s image acquisition toolbox in conjunction with MATLAB’s signal processing toolbox to detect neurons within moving cells. Examples of software that do time-lapse video analysis for static cells include ImageJ [14] and cellTrack 1.18 [15]. Many of these are open-source tools. In one study [16], after initial image pre-processing and thresholding, pixel or metric values were calculated using the IMAQ Vision Particle Analysis. Measurements used encompassed attributes like object perimeter, area, hydraulic radius, elongation factor and Heywood-circularity factor.

However there has been a dearth in software that do moving cell analysis, as many constraints like the experimental set-up, resolution of microscope, trajectory and most importantly the kind of information to be extracted remain varied. Ultimately the type of application determines the approach to go about solving the image analysis problem. Computer vision techniques help to bring automation and comprehensiveness when extracting visual information from microscopic cell images. To decide the most apropos technique requires studying the task at hand in detail.
Chapter 3

Dielectrophoresis theory

Electrokinetic phenomena such as electrophoresis and dielectrophoresis describe the movement of particles under the influence of electric fields. Electrophoresis was discovered in 1807 by Ferdinand Frederic Reuss (Moscow State University) [17]. He described it as the movement of charged particles in direct current (DC) or low-frequency alternating current fields. The term dielectrophoresis (DEP) was introduced by Pohl [18]. Dielectrophoresis is the movement of dielectric particles in non-uniform electric fields as seen in Fig. 3.1. This translational force also depends on the magnitude and frequency of the applied electric field. In short, the manipulation of cells became possible.

![Dielectric particle in non-homogeneous or point-plane electric field](M. P. Hughes, "AC Electrokinetics: Applications for Nanotechnology," Used under fair use, 2016.)

The DEP force acting on a spherical particle of radius $r$ suspended in a fluid of absolute dielectric permittivity $\epsilon_m$ is given by

$$F_{DEP} = 2\pi r^3 \text{Re} \frac{\epsilon_p^* - \epsilon_m^*}{\epsilon_p^* + 2\epsilon_m^*} \nabla(\vec{E} \cdot \vec{E})$$

such that

\[ (3.1) \]
\[ K(\omega) = \frac{\epsilon_p^* - \epsilon_m^*}{\epsilon_p^* + 2\epsilon_m^*} \]

where Re\[K(\omega)\] is the Clausius-Mossotti (C-M) factor and determines the effective polarizability of the particle and the factor \(\nabla E^2\) is proportional to the gradient and the strength of the applied electric field with angular frequency \(\omega\). The latter is dependent on the complex permittivity of both particle and medium. Indices \(p\) and \(m\) refer to the particle and the medium, respectively. By varying the frequency of the applied field and surrounding medium, the polarizability parameter Re\[K(\omega)\] can vary between +1.0 and −0.5. This range holds true for spherical particles. A positive value for Re\[K(\omega)\] corresponds to a positive DEP force and leads to an induced dipole moment aligned with the applied field. A negative value for Re\[K(\omega)\] produces a negative DEP resulting in an induced dipole moment aligned against the field. A point to note would be that reversing the polarity of the applied voltage does not reverse the DEP force field since there is a squared term \(\nabla E^2\) in the equation of the DEP force. Intuitively, Fig. 3.2 and (3.1) together indicate how the changing magnitude and direction of the real and imaginary parts of the C-M factor can dominate the DEP force.

DEP experiments conventionally consist of visualizing cell responses to the changing frequencies and voltages of applied electric field, and determining the instant where the DEP force is null. These crossover frequency values are dependent on the type of the cell under consideration as well as conductivity of medium. Subsequently, this phenomenon makes for an excellent platform that not only can compare different cells, but also identify and isolate cells if their crossover frequencies are a priori knowledge. Other characterization techniques based on the Clausius-Mossotti factor of a particle could be particle velocity measurement in an electric field gradient, batch-wise collection and counting of particles (in a suspension of known concentration) that collect at posts, and measurement of levitation height which is proportional to the negative-DEP force. Dielectric particles or cells collecting at electrodes tend to have an effect on the impedance of the electrodes too, which can be exploited for cell characterization.

In this work, the dataset used is a result of DEP experiments aimed at cell separation (video courtesy Dr. Davalos’s lab at Virginia Tech and online video) [1][21]. The proposed system will also work on similar lines of batch-wise cell isolation and separation, giving a higher priority to cell characterization.
Figure 3.2: Plot of real and imaginary parts of Clausius-Mossotti factor. The solid lines indicate the real part and the dotted indicate the imaginary part. The different colors are indicative of different media conductivity (from [20] O. Castillo-Fernandez, N. Uria, F. X. Munsoz, and A. Bratov, “Cell concentration systems for enhanced biosensor sensitivity,” in Nanotechnology and Nanomaterials - "Biosensors - Micro and Nanoscale Applications", 2015, ch. 7, Used under fair use, 2016).
Chapter 4

Setup of proposed system

The videos used in this work have been courtesy Dr. Davalos’s lab [1] and an online dataset [21]. The original idea was to make use of the videos out of the proposed phenoCHIP pump-station (courtesy phenoCHIP LLC [22]). The setup discussed here relates to phenopCHIP’s experimental setup for dielectrophoresis.

The main components of this setup are as follows: the microfluidic chip, Labsmith pump-station [23], Labsmith Synchronized Video Microscope (SVM) [23], Agilent function generator [24], Trek amplifier (for low frequencies) [25] or a customized high voltage-high frequency (HVHF) amplifier [26] and the PyQt graphical user interface. The graphical user interface is a contribution of this work. The rest of the components are proprietary to phenoCHIP LLC. The variables in the experiments include the choice of dielectric particles such as different types of tumor cells, stem cells, polystyrene beads (for initial testing) of varying diameters (in µm). Other things that are manipulated are the buffer or media (of varying permittivity), electrode geometries (using various fabrication techniques) and AC field parameters such as voltage, frequency and phase. All this is done to obtain the highest degree of cell separation between any two types of cells under the dielectrophoresis effect.

4.1 The microfluidic chip

The microfluidic chip is a set of micro-channels moulded from a polymer called Poly-DimethylSiloxane (PDMS). The microfluidic chip used in the proposed system comprises micro-channels replicated as 12 chambers which provide the bio-chemical environment for the experiments. This microfluidic chip also incorporates insulated pillars or microstructures (of size 20 µm). The electrode chambers lie on the two edges of the chip in which the ionic liquid is maintained. This liquid acts as the electrode (positive and negative) after being charged by the AC field using lead contacts. The microchannels are connected to the outside world through input and output holes pierced in the chip. It is through these holes that the
buffer carrying the cells is injected and removed using syringes and capillary tubing. The average dimensions of a chip are 50mm x 75mm. The manifold that houses this chip is of the same dimensions.

4.2 Labsmith pump-station

The pump-station comprises two 40 µl Labsmith SPS01 syringe pumps, three 3-port switching two-position Labsmith AV202 valves and a 4-port switching two-position Labsmith AV202 valve. A 4-port Labsmith manifold controls the switching between the four valves. All the above components are integrated on an Electronic Interface Board (EIB) board from Labsmith, which is connected to the computer. The software that communicates with the above devices through the Electronic Interface Board (EIB) is the uProcess software.

4.3 Labsmith SVM

The SVM used is the SVM340, an inverted fluorescence video microscope popularly used for microfluidics and microbiology finding. It comprises a synchronously pulsed illuminator, sensitive 30 frames per second (fps) camera, and powerful video analysis software called
4.4 Amplifier

The DEP experiments can be run either at low frequency (10kHz to 10 MHz) or at a higher frequency range of 55 KHz to 110 MHz. The reason is that cells encounter two crossover frequencies. In other words, while plotting the Clausius-Mossotti factor against frequency there are two instances when the C-M factor is reduced to zero or the DEP force is reduced to null, as seen previously in Fig. 3.2. The first crossover frequency can be found at 500 kHz to 10 MHz and the second in the 55-110 MHz range. Meanwhile the aim of the proposed system is to run cDEP experiments at the lower cross-over frequency.
Nonetheless the amplification range of conventional off-the-market high frequency-high voltage amplifiers (like Trek) which was used for prototyping purpose (still used), was limited to around 1 MHz. In the proposed system, a special high frequency-high voltage amplifier (courtesy Dr. Swami’s lab at the University of Virginia) was built to leverage higher operating frequency with adjustable gain so that the amplification response is not skewed. More information about this can be read in [26, 28].

The dataset videos were however taken at a low operating frequency (40kHz). The experiments that will be using the proposed system will be currently limited to the range of the first crossover frequencies, leaving the experiments in the second crossover frequency range as a future endeavour.

4.5 Agilent function generator

The function/waveform generator belongs to the 332XX family with a maximum frequency of 10 MHz. As an input to the HVHF amplifier, the output needed is a low distortion sine wave of constant value of 10 volts. The frequency is set to about 4-10 kHz. When used as an input to the Trek amplifier, the output voltage is set to the stepped down value of the desired amplified voltage. The frequency is set to the desired output frequency.

4.6 Graphical User Interface

The graphical user interface is a contribution of this work in the experimental setup. Automation of the cDEP process in the experimental setup provides integration of all the above components. It also proves useful as many iterations of the experiment with varying values
of voltage and frequency can be run without human assistance.

The GUI provides control over not only individual components, but introduces the functionality of running a special script. This script is a set of sequential commands to run the cDEP experiment. It mainly comprises three sections. The first section aims at filling the pump-station syringes with the mixed cells and the buffer followed by injecting them together into the microfluidic chip. The second section includes turning on the electric field to commence trapping of cells of interest followed by injecting the pure buffer to wash out the non-enriched cells. The final set of commands deal with turning the field off and injecting the pure buffer into the chip to wash out the trapped cells. Meanwhile the camera captures this entire process. Cell and buffer flow-rates as well as valve settings is controlled by the script. Controlling the electric field implies controlling the function generator and the amplifier inputs, precisely, operating voltage and frequency. There are two scripts for incorporating the two different types of amplifiers: the prototypical one (Trek) and the customized one (HVHF).
Figure 4.9: The GUI. The Browse button under uProcess Settings is used to run the script file. Other options can be used to control respective hardware controls like SVM, function generator and amplifier.
Chapter 5

Detection

5.1 Introduction

Detection in images implies the process of figuring out the location of objects of interest. These objects can be faces, things, or even blobs. In cell phenotyping, the various statistics can be measured accurately only if the cells are detected with good recall and precision. The cells in this work can take different shapes but usually look like solid blobs.

Three approaches to cell detection have been taken in this work as described below. These three approaches were implemented and their performances were evaluated. The results have been discussed in chapters 10 and 11.

5.2 Connected component analysis

Also known as region extraction, connected component analysis uniquely labels all pixels alike that are a part of a common connected region. The scanning algorithm used in this analysis scans the image from top-left corner to the bottom-right corner, giving same label to adjacent pixels with exactly the same intensity value. This labelling is done on the basis of pixel connectivity and is thus a graph theory application.

For cell detection, each connected component can be considered as a cell if the pixels corresponding to the dielectrophoresis medium and the electromagnetic poles are considered as background. Thus a binary image must be created. So the preprocessing steps include firstly converting the color image to gray-scale, followed by a global thresholding to obtain a binary image. In order to select the best threshold, Otsu’s thresholding method is used. This method selects the threshold by performing an exhaustive search for a value that minimizes the intra-class variance, with the classes being, foreground and background. Intuitively, this
should work because the cell pixels and the medium pixels come from different visual distributions. Once the binary image is obtained, the labelling algorithm is run and each label corresponds to a unique cell. From the labelled image, the bounding boxes of the cells can be computed easily. The scikit-image package was used in the implementation. Scikit-image is an open-source collection of algorithms for image processing in python.

The limitation of this method is the thresholding requirements. Take for instance, a non-uniformly illuminated frame. Binarizing it using a simple global threshold would lead to loss of information. Instead, adaptive thresholding would be needed in such a scenario. Another instance where thresholding would matter is if the object of interest changes from being lighter to darker than the surrounding. The resulting binary image would need to be inverted. In short, thresholding becomes heavily dependent on the video sequence under consideration.

5.3 Difference of Gaussians

The second approach to detection considers cell detection equivalent to traditional blob detection. A blob is a region of pixels of similar intensities embedded in a neighborhood that is visually different. Thus a blob is demarcated by edges and hence, filters that are sensitive to edges can help detect blobs. So the advantage of using blob detection is that areas that are considerably lighter or darker than the surrounding will both be detected successfully. There are various filter based approaches to blob detection. The Laplacian of Gaussian (LoG) filter results in strong positive responses for blobs that match the size of the chosen Gaussian kernel. Using a scale-normalized Laplacian operator and searching for the maxima of the filter response in both scale and space gives a multi-scale blob detector. To achieve efficient implementation of LoG operator, the LoG is approximated by the DoG. The rationale behind using a blob detector stems from the fact that the cells are solid and thus can be considered as solid blobs. The cells vary in sizes. So scale-normalized DoG at several scales is convolved with the image. The maxima of DoG response in scale-space gives the desired binary circle of radius \( r \). Thus we can say that DoG is scale-invariant, making detection of cells of varying radii possible.

However this method faces a drawback. DoG is a circularly symmetric operator in 2D. In the dataset under consideration, cells did not remain circular in shape throughout. On the contrary, many cells were arbitrarily shaped. Cells constantly changed shape due to the forces acting on them. So though this approach was accurate in detecting the center and approximate radius of a cell, unlike connected component analysis, information about cell dimensions was not retrieved.
5.4 Simple Linear Iterative Clustering (SLIC)

The advantage of the second detection approach (section 5.3) over the first approach (section 5.2) is that blob detection is independent of thresholding since the DoG filter is convolved with a 2D grayscale image. Connected component analysis, on the other hand, may lose out on critical information due to binarization using global thresholding. On the other hand, blob detection does not give cell dimension details, unlike connected component analysis. In order to combine the best of both approaches, the third approach was implemented. This uses the SLIC superpixel algorithm along with DoG.

The crux behind any superpixel algorithm is replacing the conventional pixel grid with a grid of larger-sized arbitrary-shaped superpixels [29]. SLIC is a superpixel algorithm that uses a variant of K-means clustering on the (L, a, b) values of the CIELAB color space and (x,y) locations to generate superpixels. It is faster, more memory efficient with higher segmentation performance as compared to other state-of-the-art superpixel algorithms [29]. To detect cells, a two-step process was followed. The DoG blob detector first identified the centers of objects of interest. Then the superpixels that contained these centers were extracted using SLIC and approximated to be cells. Thus the idea was that the SLIC superpixel would better approximate the cell and at the same time not rely on a global threshold. Subsequently, the information regarding the shape of the cells would be more accurate and would not be limited to circular blobs. Also, SLIC can take advantage of the input color space (Lab). However, running SLIC involves the selection of two hyper-parameters which are the number of superpixels to be created as output and compactness factor. This indirectly influences the range of the size of a detection and helps improve segmentation performance if the dataset is known beforehand, which is a valid assumption in this work.
Figure 5.1: An illustration showing difference of Gaussian filter. The filter helps to reduce high frequency components [30] [Online]. Available: http://www.olympusmicro.com/primer/java/digitalimaging/processing/diffgaussians/, Used under fair use, 2016.

Figure 5.2: Sample segmentation output from [29] (“Slic superpixels compared to state-of-the-art superpixel methods,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, 2012, pp. 2274–2282, Used under fair use, 2016) using SLIC algorithm.
Chapter 6

Tracking

6.1 Introduction

Tracking means to follow either a single or multiple objects of interest in a video, and to provide their position continuously. Such track information can be used to extract various interesting properties pertaining to the dynamics of the system under study. In this work, tracking moving cells is of utmost importance as these cells change their morphological features like shape and size as they translate. All these properties can be useful in studying the dielectrophoretic force and its influence on various types of cancerous and non-cancerous cells. This information can be exploited to characterize different cells in the long run based on automated cell-profiling, which has been a motivation behind this work. Target representation and localization are major components of detection in visual tracking. The two different ways to localize moving objects can be in terms of the natural features or some visual markers. This work has concentrated on the natural features of the cell object rather than any marker. The features, precisely the centroids of the blob or cell-boundary, are nothing but the results obtained from the detection methods discussed in the previous chapter.

The next step in tracking after target localization is filtering and data association. It is important that information in previous frames is successfully linked to the current frame. Many algorithms perform this association hypothesis wherein the most probable location of the object under consideration is estimated, like using past measurements or nearest neighbour distance and so on.

6.2 Nearest neighbor tracker

The first implementation of the tracking algorithm is based on the nearest neighbor paradigm. The detection algorithm is run on every frame and a simple nearest neighbor clustering across
time is used to group centroids of a cell. For a detected cell, the Euclidean distance from all centroids in the next frame is calculated. The minimum value of this distance corresponds to a match. A pre-set maximum threshold on the distance helps in identifying scenarios where there are no suitable data-points available for mapping in the next frame. Thus if there is no centroid available within the threshold, it implies that either the cell has been occluded in that particular frame or it may have left the area of interest/camera-field altogether. In such cases, depending on the prior history, centroids are extrapolated for those frames where data-points are missing. If the number of frames for which centroid values are extrapolated crosses a predetermined value, tracking for the cell in question is stopped.

### 6.3 Kalman filter

Kalman filtering is an iterative prediction-correction process [31]. It makes an estimate of an unobserved variable on the basis of measurements which may be noisy. The first step, prediction, deals with guessing the expected value of the state with corresponding uncertainties. In the correction step, the correction to the estimate of current state is calculated using the noisy measurement of the state. The state is modelled as a vector which comprises 4 variables, namely, \(x\)-coordinate of current position, \(y\)-coordinate of current position, current speed in \(x\) and \(y\) directions. In this work, the initial position for each of the cells is obtained from the cell-detection algorithm which is the measurement itself. Furthermore, zero initial velocities in \(x\) and \(y\) directions are set. Transition covariance and observation covariance are set to some predefined values. The points to be tracked for every frame are assumed to be the centroids of the blobs. Kalman filtering is a special case of Bayesian filtering where the filtering is linear, quadratic and Gaussian. The python version of Kalman filtering in OpenCV was used in the implementation.

### 6.4 Optical flow

Optical flow is the pattern of motion of objects of interest in a sequence of frames [32]. It is a tracking method where positions of points in a frame are followed and estimated in the following frame. Accordingly, it becomes very important to determine the good features or points to track. The implementation of optical flow in this work assumes the points to be tracked as those provided by the detection algorithm. In other words, it is the centroid of the cell that is obtained from the detection method that is fed to the optical flow method, along with the previous and the current frame. The assumption on which this method works is that neighboring group of pixels have the same intensity and same motion as shown below.

\[
I(x, y, t) = I(x + dx, y + dy, t + dt)
\]
By taking Taylor series approximation of the right-hand side, removing common terms and dividing by $dt$, the optical flow equation is given as

$$f_x u + f_y v + f_t = 0$$

where $f_x$ is image gradient along $x$-axis, $f_y$ is image gradient along $y$-axis and $f_t$ is time gradient. To determine the velocity vector unknowns, $u$ and $v$, the Lucas-Kanade method is employed which obtains a compromise solution by least squares principle thus giving less weight to outliers. The movement of cells being small suits the main assumption of optical flow method that motion of objects of interest should be small.
Chapter 7

Cell Counting

7.1 Introduction

As mentioned in the earlier chapters, this work is an analytical tool for cell cytometry, that supplements the proposed phenotype-based cell isolation system. The main goal of the system is to perform cell enrichment, or in other words separate and collect cells of interest without using any cell surface markers. In addition, this work supports the larger goal of developing an efficient cancer cell profiling system.

Cell cytometry is the measurement of the characteristics of cells like cell size, cell count, cell morphology (shape and structure) and so on. A very important aspect of cell enrichment is keeping count of the cells that have been separated. The first ever known method for cell-counting was using the hemocytometer [hemo]. Traditionally these were used to aid manual counting and continue to be used even today. Later came the advent of image cytometers that made use of optical microscopy to analyze fluorescence-stained (fluorochromes) cells. However there were technical difficulties in primitive imaging systems, which in turn paved way for the flow cytometer. With this, cells were manipulated using flow techniques and characterized optically or using an electrical impedance method called the Coulter principle. Currently, image cytometers have advanced because of the advancement in digital cameras.

One of the applications targeted in this work, cell counting, has been implemented to achieve functionality similar to that of an image cytometer. The advantage of this application over the latter is that the in-house imaging system supports the cell-counting software to count cells as they continue to be manipulated under the DEP force during the experiments. This is not possible with any of the aforementioned cytometers. Open-source image processing and the in-house imaging system lead to reduced costs. Customized analysis of cells under cDEP effect, taking into consideration chip and electrode geometry, leads to accurate data acquisition. Unlike the cytometers, the advantage of being able to observe the cells over time when they are actually under DEP effect, is unparalleled. Another benefit is the absence of
phototoxic fluorochromes that impart fluorescence to the cells and are commonly used along with flow and image cytometers.

## 7.2 Approach

The final goal in this work is to not only count the total number of cells but also to count individual sub-populations of cells. During video analysis, blobs or cells detected in every frame using the detection algorithms are independently tracked and cell history is maintained. Whenever a new cell enters the field of view, a new object is created in the software and its tracking commences. A common norm of cells flowing from left to right in the videos is assumed. Whenever a cell leaves the frame, the corresponding object in the software is assumed to be no longer active, though its history remains logged. The $x$-coordinate of the cell or blob is instrumental in deciding the entry (first $k$ columns of the image grid) and exit (last $k$ columns of the image grid) of a cell.

Two approaches have been implemented for counting of the cells. The first approach works best if the camera is placed at the point of exit on the chip, where the chamber narrows down. In this first approach, the time-interval for which the electric field is switched on is of importance. We assume that the electric field is on for a time-interval sufficient to cause the cells that are not attracted to the posts to leave the chip entirely. Cells are either attracted by the pillars (enriched cells) or not attracted by them (non-enriched cells). The number of objects counted after the field is turned off correspond to the enriched cells. The difference between the above two gives the non-enriched cell population.

The second approach exploits the velocity information of each cell to estimate the sub-populations. Each cell is tracked as an independent entity over several frames. Initially, the electric field is turned on. New cell objects are created in the software as and when cells enter the frame. Some cells would be trapped, others would go by. Cell objects that show non-zero velocity throughout their history are clearly the non-enriched cells. Cell objects with zero velocity for some time-span, sum up to give the count of enriched cells. Thus, exact field status at each frame is not necessary but such information can help if cells can get stuck momentarily due to unexpected phenomena. This approach can be harnessed to a greater level if the camera-view could span the entire chip, else averaging the count value can be done.
Chapter 8

Proximity Detection

8.1 Introduction

In the study that follows DEP, the analysis of distances between cells as they flow through the chambers is of great interest. Distances between cells give information about the interactions between cells. Cells in electric fields can interact with each other to form what are called pearl chains. Pearl chains exist mainly because of the dipole-dipole forces that act between polarized cells in close proximity, leading to formation of doublets, triplets or higher order chain-like structures. In a pearl chain, the opposite charges on cells are aligned facing each other, leading to a strong attractive force between the cells. The pearl chain’s induced dipole is oriented along the electric field lines.

Pearl chains can influence the crossover frequencies; frequencies where a transition between positive and negative DEP force may occur [34]. As a result, though pearl chains may be a desired phenomena in certain applications like creating nano-wires (Fig. 8.1) or other similar structures, in this work they are perceived as an undesired effect. Primarily because of the effect they have on the cDEP experimental results. In this work, a functionality to determine whether a given cDEP video is pearl-chained or not has been incorporated.

8.2 Approach

From the analysis point of view, each cell has been considered as an object in software, and a history of its varying parameters like shape, velocity has been maintained using detection and tracking methods. Making use of this knowledge, a proximity detection algorithm has been implemented. In this, a predetermined radius around each cell is checked in every frame, and cell IDs of all other cells in this marked neighborhood are stored. For every cell in each frame, a list of all close-proximity cells for the last \( k \) frames is maintained. A threshold for the occurrences of these closely spaced cells is determined at the beginning of the analysis. Next the proximity list described earlier is checked against the predetermined occurrence threshold. If two or more cells cross this threshold, it becomes indicative of pearl-chain formation. The reason for maintaining this window of \( k \) frames and the threshold is to make sure that ephemeral instances of chaining or dipole-dipole attraction are not falsely classified as pearl chaining instances.
Chapter 9

Dataset and Annotation

9.1 Introduction

The dataset used for the experimentation consisted of 4 videos (found online and courtesy Dr. Davalos’s lab) [1][21]. Each of the videos is unique, indicating that the dataset was diverse in terms of colorspace, pillar-sizes and cell sizes. The following are the screen-shots of the videos used along with the total number of key frames. FFMPEG (a cross-platform multimedia framework) was used to convert the video into image sequences comprising key frames [35].

The diversity is specified by the varying pillar sizes, different chip-geometries, the different fluorescent wavelengths emitted indicating different color intensities and different cell sizes. Annotating the videos was challenging given that a single video comprises multiple frames and each of the frames consisted of more than 40 cells on an average. Also, the cells being small objects are not easy to annotate. Various tools were experimented with while selecting the best annotation interface.

9.2 VATIC

Video Annotation Tool from Irvine, California, is an open-source online interactive tool for video annotation [4]. It facilitates the user to draw boxes around objects of interest in a frame, giving a new label to a newly created object. Then it tracks the objects through the video with the label remaining consistent, giving the user chances to rectify the bounding boxes in case predicted value is incorrect. VATIC was used in the offline mode (without crowd-sourcing the work to Amazon’s Mechanical Turk). Figures 9.2 shows the tool in the process of annotating a video.
However VATIC proved a bit limited in use. Primarily because it worked well for annotating larger objects in a frame that moved in some consistent direction in smaller steps. The objects of interest in the dataset under consideration (i.e., the cells) showed rapid movements in arbitrary directions depending on the forces such as DEP, hydrodynamic drag and gravity acting on them. Mostly the bounding boxes predicted by VATIC were slightly-off in every frame and had to be set right manually after every few frames. This was time-consuming and still did not give the best ground truth results.

9.3 MATLAB interface

As an alternative to VATIC, a two simple interfaces for detection and tracking annotation were implemented Fig. 9.3 and Fig. 9.6. The first interface makes use of the detection algorithms discussed in the earlier chapter, for annotation. Firstly, the qualitatively best detection algorithm for a video was chosen, detection was run on the video in Python and all the bounding boxes logged in .mat files. These .mat files were then read and the boxes were plotted on the original frames using the imrect functionality in MATLAB, displaying them on a MATLAB user interface (UI). The user is then given options to resize the boxes, move them, delete them or add new ground truth boxes which are overlaid on the original frame as seen in Fig. 9.4 and Fig. 9.5. Once the bounding boxes appear sufficiently close to the ground truth, the user can pass on to the next frame in the video sequence and repeat the above steps. The bounding boxes, thus corrected, become the detection ground truth for the particular video sequence under consideration. Evaluation of the various detection algorithms is done against this ground truth. To evaluate the performance of the tracking algorithms, ground truth for tracking is required. In tracking, labelling the object of interest becomes fundamental. The labels are synonymous with the cell IDs. So bounding boxes present in every frame of the video sequence should have the correct corresponding cell IDs. To accomplish this, one of the proposed tracking algorithms was used. The ground truth bounding boxes obtained using the aforementioned detection MATLAB UI were fed to the nearest neighbor tracker. The output of this tracker was then rectified manually using the tracking MATLAB UI. This interface allowed easy re-labelling of any incorrect labels obtained from the chosen tracking algorithm. Figure 9.6 shows the interface with the current frame displayed on the left and the previous, corrected frame on the right for easy reference. The current predicted labels are displayed in text boxes that allow correction. Thus, in this annotation task, only the labels are changed and bounding boxes remain constant.
(a) Video sequence 1: 1783 frames

(b) Video sequence 2: 529 frames

(c) Video sequence 3: 1783 frames

(d) Video sequence 4: 100 frames. This is a test video to study the detection and tracking performance on set-ups similar to cDEP. It consists of microfluidic oil droplets in water manipulated by liquid electrodes [21] [Online]. Available: https://www.youtube.com/watch?v=5dPNzAh4tg0, Used under fair use, 2016.

Figure 9.1: (a)-(d) one frame from each of the dataset videos.
Figure 9.2: (a), (b) show two consecutive frames being annotated in VATIC. After annotating the first frame it was expected that the next frame would be well-annotated. However manual annotations had to be repeated for every frame as the cells were not followed close enough from the previous frame. In many scenarios the moving/resizing was minute but given the number of objects of interest, it was a tedious task.
Figure 9.3: The MATLAB UI to facilitate creation of ground truth boxes.

(a) Single detection for nearly fused cells. (b) Resizing and adding new boxes. (c) Final ground truth boxes.

Figure 9.4: (a)-(c) highlight the scenarios where the detection algorithm may not perform well, requiring resizing/adding of new bounding boxes.
Figure 9.5: (a) shows the output of the detection algorithm overlaid on the frame using the detection MATLAB interface. (b) shows the ground truth boxes overlaid on the frame. These were obtained using the MATLAB UI functionality. As is seen, the ground truth boxes are more closely-fitted to the cell boundaries than the detection.

Figure 9.6: Re-labelling of labelled ground truth boxes for precise track-correspondence using tracking MATLAB UI.
Chapter 10

Quantitative Evaluation

10.1 Detection metrics

For detection, firsts the ground truth (GT) was extracted and a complete dataset was prepared. As discussed in the previous chapter, a simple MATLAB interface was devised to annotate the dataset. After obtaining the ground truth, average precision, recall and F-measure values were measured for each detection algorithm to score the retrieval quality.

Mathematically,

\[
\text{Precision} (P) = \frac{TP}{TP + FP} = \frac{|G_b \cap P_b|}{|P_b|} \tag{10.1}
\]

\[
\text{Recall} (R) = \frac{TP}{TP + FN} = \frac{|G_b \cap P_b|}{|G_b|} \tag{10.2}
\]

\[
F - \text{measure} (F_1) = 2 \frac{P \cdot R}{P + R} \tag{10.3}
\]

where,

\[TP = \text{True positives}\]
\[FP = \text{False positives}\]
\[FN = \text{False negatives}\]
$G_b$ = Total ground-truth boxes
$P_b$ = Total predicted boxes

A detection was considered a true positive if its intersection over union (IOU) with respect to the ground truth bounding box was above 0.5. Precision was calculated as the ratio of number of true positives over the predicted bounding boxes. Recall was computed as the ratio of true positives to the actual number of bounding boxes in the ground truth.

Table 10.1: Average precision and recall values for different videos in the dataset (Fig. 9.1) for different detection algorithms.

<table>
<thead>
<tr>
<th>Detection algorithms</th>
<th>Video 1 (100 frames)</th>
<th>Video 2 (100 frames)</th>
<th>Video 3 (100 frames)</th>
<th>Video 4 (100 frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F-measure</td>
<td>Precision</td>
</tr>
<tr>
<td>Connected component</td>
<td>0.0019</td>
<td>0.0009</td>
<td>0.0012</td>
<td>0.0019</td>
</tr>
<tr>
<td></td>
<td>0.5943</td>
<td>0.5341</td>
<td>0.5626</td>
<td>0.5943</td>
</tr>
<tr>
<td></td>
<td>0.8330</td>
<td>0.7916</td>
<td>0.8118</td>
<td>0.8330</td>
</tr>
<tr>
<td></td>
<td>0.8591</td>
<td>0.9514</td>
<td>0.9029</td>
<td>0.8591</td>
</tr>
<tr>
<td>Difference of Gaussian</td>
<td>0.9596</td>
<td>0.2471</td>
<td>0.7752</td>
<td>0.7768</td>
</tr>
<tr>
<td></td>
<td>0.4983</td>
<td>0.2316</td>
<td>0.7486</td>
<td>0.8821</td>
</tr>
<tr>
<td></td>
<td>0.9539</td>
<td>0.2391</td>
<td>0.7617</td>
<td>0.8261</td>
</tr>
<tr>
<td>DoG + SLIC</td>
<td>0.518</td>
<td>0.7683</td>
<td>0.8245</td>
<td>0.8568</td>
</tr>
<tr>
<td></td>
<td>0.513</td>
<td>0.7199</td>
<td>0.7951</td>
<td>0.9714</td>
</tr>
<tr>
<td></td>
<td>0.5155</td>
<td>0.7433</td>
<td>0.8095</td>
<td>0.9105</td>
</tr>
</tbody>
</table>

In Table 10.1, connected components analysis fails for Video 1, primarily because of the thresholding needed for that video due to non-uniform illumination. This could be a drawback for connected components, since the analysis becomes threshold dependent.

The high performance of difference of Gaussian for Video 1 may be a bit biased as it was the base technique used for the ground truth annotation for that video. However the cells in Video 1 were mostly spherical and were surrounded by a thick boundary. While annotating the ground truth these boundaries were taken into consideration as cell region. The reason why DoG + SLIC does not perform that well is because the thick boundaries were considered as super-pixels themselves by SLIC and were neglected since the blob centers did not lie in the boundary super-pixels. DoG on the other hand gave loose square bounding boxes that encompassed the thick cell boundaries too. That explains the high performance of DoG.

In Video 2, most of the cells looked ellipsoidal with the major axes of the ellipses aligned along the $x$-direction. This could be due to the higher flow-rate of the buffer during cDEP experimentation and the resultant drag force acting on the cells. The reason why DoG did not seem to perform well could be due to the cells no longer remaining circular in shape. This may have led to lower IOU values as compared to the other videos in the dataset.

On the whole, the performance of DoG + SLIC appears to be the best since it is consistent with the varying dataset.
10.2 Tracking metrics

The evaluation methodology adopted for tracking was similar to that mentioned in [36]. The input to the tracking algorithms were the ground truth detection boxes. Precision and recall were measured in a similar way as in detection metrics but with the added concept of successful frames. While comparing the ground truth track and the prediction track, each prediction track was scored on the basis of the IOU of bounding boxes in the predicted track with respect to ground truth bounding boxes. To measure the performance on a sequence of frames, the number of frames whose overlap was larger than the given threshold was counted. These frames are called successful frames. If the ratio of successful frames to total number of ground truth frames was greater than 0.5 then the track was considered a positive track, else it was classified as a negative track. Accordingly, precision was calculated as the ratio of number of positives over the total predicted tracks. Recall was computed as the ratio of positives to the actual number of tracks in the ground truth.

<table>
<thead>
<tr>
<th>Tracking algorithms</th>
<th>Video 3 (100 frames)</th>
<th>Video 4 (100 frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest neighbor tracking</td>
<td>Precision 0.9375</td>
<td>0.9167</td>
</tr>
<tr>
<td></td>
<td>Recall 0.9615</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>F-measure 0.9493</td>
<td>0.9565</td>
</tr>
<tr>
<td>Kalman tracking</td>
<td>Precision 0.9259</td>
<td>0.9167</td>
</tr>
<tr>
<td></td>
<td>Recall 0.9615</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>F-measure 0.9434</td>
<td>0.9565</td>
</tr>
<tr>
<td>Optical flow tracking</td>
<td>Precision 0.4792</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Recall 0.5897</td>
<td>0.5455</td>
</tr>
<tr>
<td></td>
<td>F-measure 0.5287</td>
<td>0.5218</td>
</tr>
</tbody>
</table>

Table 10.2 indicates that Kalman filtering and nearest neighbor tracker perform better than optical flow. One reason could be that optical flow is suitable for tracking objects with smaller displacements. Initially it was expected that cell movements would be small. However on testing the algorithms and even during VATIC annotation it was found that cells under DEP have arbitrary displacements. Another point to be noted is that the optical flow algorithm tracks good feature points in order to estimate the next points. In this work, the good features given to the algorithm included the centroids of cells obtained after detection. The lack of significant feature points for a cell could also be a reason for the poor performance of optical flow. This could be leading to error accumulation or drift, since current predictions were dependent on previous predictions.

Meanwhile, since each of the tracking algorithm was fed with ground truth detection boxes,
the values of precision, recall and F-measure were expected to be high for any decent tracking algorithm as is seen in the case of Kalman and nearest neighbor Table 10.3 shows the result for all the tracking algorithms with detection boxes obtained using DoG + SLIC (approach 3). Kalman filtering and nearest neighbor tracker continue to perform better than optical flow.

Table 10.3: Average precision and recall values for tracking algorithms using DoG + SLIC detection boxes.

<table>
<thead>
<tr>
<th>Tracking algorithms</th>
<th>Video 3 (100 frames)</th>
<th>Video 4 (100 frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest neighbor tracking</td>
<td>Precision 0.7209</td>
<td>0.8462</td>
</tr>
<tr>
<td></td>
<td>Recall 0.7949</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>F-measure 0.7561</td>
<td>0.9167</td>
</tr>
<tr>
<td>Kalman tracking</td>
<td>Precision 0.7469</td>
<td>0.8462</td>
</tr>
<tr>
<td></td>
<td>Recall 0.7564</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>F-measure 0.7516</td>
<td>0.9167</td>
</tr>
<tr>
<td>Optical flow tracking</td>
<td>Precision 0.3587</td>
<td>0.3810</td>
</tr>
<tr>
<td></td>
<td>Recall 0.4231</td>
<td>0.7273</td>
</tr>
<tr>
<td></td>
<td>F-measure 0.3883</td>
<td>0.5</td>
</tr>
</tbody>
</table>

10.3 Cell count metric

To measure the cell-count performance, the mean root-mean square (RMS) cell-count error was computed using the predicted cell-count and the ground truth cell-count. Average RMS cell count error over all frames for a video was computed. Mathematically,

\[
RMS\ cell\ -\ count\ error = \sqrt{\frac{\sum_{k=1}^{N} [PB_k - GB_k]^2}{N}}
\]

(10.4)

where,

\[PB_k = \text{Number of predicted boxes in the } k^{th} \text{ frame}\]
\[GB_k = \text{Number of ground-truth boxes in the } k^{th} \text{ frame}\]
\[N = \text{Total number of frames}\]

In Table 10.4, the cell count error for Video 1 using connected components is large. This is indicative of the failure in detection as explained in section 10.1. Connected component showed a higher error count as compared to the other two algorithms. This could be due to the fact that cells that are too closely spaced and share a boundary may have been detected as one entity by connected component analysis. Subsequently causing cell count error.
### Table 10.4: RMS cell-count error values for different videos in the dataset for different detection algorithms.

<table>
<thead>
<tr>
<th>Detection Algorithms</th>
<th>Video 1 (100 frames)</th>
<th>Video 2 (100 frames)</th>
<th>Video 3 (100 frames)</th>
<th>Video 4 (100 frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected component</td>
<td>113.79</td>
<td>3.94</td>
<td>2.49</td>
<td>1.30</td>
</tr>
<tr>
<td>Difference of Gaussian</td>
<td>2.82</td>
<td>2.81</td>
<td>1.90</td>
<td>1.13</td>
</tr>
<tr>
<td>DoG + SLIC</td>
<td>2.82</td>
<td>2.81</td>
<td>1.90</td>
<td>1.13</td>
</tr>
</tbody>
</table>
Chapter 11

Qualitative Evaluation

11.1 Detection

Connected component analysis

Fig. 11.1 highlights a couple of results obtained by running the first detection approach on the dataset (connected component analysis as explained in section 5.2). Fig. 11.1a and Fig. 11.1b show successful detection scenarios. However for Video 1 (Fig. 11.1c), adaptive thresholding was applied instead of Otsu thresholding since there were spatial variations in illumination. Subsequently, Fig. 11.1d shows the binary image of a frame from Video 1 which was given to the labelling algorithm. This explains the poor performance of the algorithm.

(a) A frame from Video 2. Successful detection. (b) A frame from Video 3. Successful detection.

Figure 11.1: Results for connected components detection.
Figure 11.1: Results for connected components detection (contd.).

**Difference of Gaussian**

Fig. 11.2 highlights results obtained by running the second detection approach on the dataset (difference of Gaussians as explained in section 5.3). It can be noted from Fig. 11.2a to Fig. 11.2c that the detection boxes are perfectly square. In Fig. 11.2a, most of the cells generally appear spherical. However in Fig. 11.2b, information regarding cell height, width is lost to a certain degree. Similarly Fig. 11.2c shows that the oblong cells are not bounded tightly.

Figure 11.2: Results for difference of Gaussian detection.
DoG and SLIC

Unlike the other two detection methods, the boxes detected using DoG + SLIC (discussed in section 5.4) are more compact and retain the information pertaining to cell dimensions. The two SLIC parameters varied for each video were compactness and number of segments. This was dependent on the average size of cell in a given video. Fig. 11.6 highlights results obtained by running this detection algorithm.
11.2 Tracking

Fig. 11.5 shows final tracked cell paths for Video 3 using the respective tracking methods. It is seen in Fig. 11.5c that many cell centroids are not tracked accurately. These instances are highlighted too.

Fig. 11.4 shows a limitation faced by all the tracking algorithms: merging of cells. When a cell is occluded, the tracking algorithm will keep predicting the position of the cell for a predetermined number of frames so that the tracking can recover when the cell appears again. But this has the unwanted side-effect of penalizing the tracking algorithms when cells merge and move as a single entity. Fig. 11.4a indicates a frame at time $t$. Till this moment the cells move distinctly and therefore are tracked successfully. (b) indicates a frame at time $t+k$ when the two specified cells have merged. It is seen that two cells are being tracked instead of one.
(a) A frame from Video 4. Successful tracking.

(b) A frame from Video 4. Failed tracking.

**Figure 11.4:** Merging of cells.
(a) Tracks for Video 3 using nearest neighbor tracking. Successful tracking.

(b) Tracks for Video 3 using Kalman filtering. Successful tracking.

Figure 11.5: Tracking results for Video 1.
11.3 Proximity detection

Fig. 11.6a and Fig. 11.6b show the result of running the proximity detection algorithm on Video 2 and 1 respectively. The red circles are indicative of an alerting mechanism if the cells get closer than the predetermined threshold. This threshold is dependent on the radius of the cells as explained in chapter 8 on pearl chaining. Video 1 is a classic example of pearl chaining.
(a) A frame in Video 2.

(b) A frame in Video 1. In this frame the voltage is turned on. The cells are therefore trapped to the pillars.

Figure 11.6: Some proximity detection results.
11.4 Cell characterization

Fig. 11.7 shows visualizations of cell dimensions and velocities. Fig. 11.7a shows the plot of width and height of a random cell plotted for every frame for which the cell existed in the field of view. Fig. 11.7b shows the plot of $x$-velocity and $y$-velocity of another cell plotted for every frame for which the cell existed in the field of view. These results were obtained using the tracking algorithm run on a video sequence. Subsequently this illustrates how the knowledge of cell specific characteristics can be useful in studying more about effect of DEP on cell shape and velocity.

(a) Width and height changes for a cell.

(b) Velocity changes for a cell.

Figure 11.7: Visualizations of stored data.
Chapter 12

Conclusions

Three detection algorithms and three tracking algorithms were successfully implemented and tested with a decent performance in terms of precision, recall and F-measure. Given the dataset at hand, difference over Gaussians along with SLIC proved to be the best detection method with the lowest RMS cell-count error and highest average F-measure of 0.745. The underlying notion for this detection approach was that cells resemble blobs which aren’t perfectly spherical. The Kalman tracking proved to be the best tracking algorithm with an average F-measure of 0.95. Its predictive modelling approach and the fact that it can handle large cell displacements in random directions proved to be instrumental for this dataset. This combination of detection and tracking method proves to be best suited for cell phenotyping in cDEP experiments using the proposed setup. The average F-measure with DoG + SLIC detection and both nearest neighbor and kalman tracker was 0.83.

Future Work

The proposed phenoCHIP setup is similar to the one being used in Dr. Davalos’s lab. The devised algorithms can be thus run on the videos obtained from the proposed setup. They can be fine-tuned in order to obtain higher precision, specific to the setup. For DoG + SLIC hyperparameters, precision values can be plotted for a range of these hyperparameters (number of segments and compactness) for a given video and the best hyperparameter combination can then be chosen. To improve the performance of Kalman filter tracking, the extended Kalman filter could also be implemented in future. Videos of cells moving at the entry or exit of the microfluidic chamber can give accurate count of cells. So setting appropriate camera imaging position on the chip will prove to be fruitful. The log of information such as height, width, velocity in x-y directions of the cell entities can be used for further study of cell phenotype. This can be made possible using a DEP electric field overlay obtained using COMSOL software for the cDEP device.
Bibliography


[27] https://www.youtube.com/watch?v=qVmjcfyBocg, The illustration indicating the cDEP device is the work of Dr. R. Davalos’s lab (cit. on p. 12).


