**Szereted csinálni azt amit csinálsz, és élvezed a kutató munkát!!!! ☺**

**Célkitűzések:**

* Mikrofluidikai rendszerek, szűrések bemutatása (általános + parazita + eszközök ismertetése)
* Irodalomkutatás bemutatása
* Algoritmusláncolat létrehozása:
  1. Sutaration algoritmus (mi az alapja, miben különbözik a többi detekciós eljárástól)
  2. Canny edge detector: (élkeresésről általánosan is írni + Cannyt bemutatni)
     + Sima + skálázott összefűzése (összehasonlítások is persze)
  3. Színinformációs eljárás: (általános bevezetés, hogy mi az alapja, miben más, mint a többi)
     + Lokális színkülönbségeken alapuló élkeresés
  4. Alapszerkezet kiszűrése a képről: (hol szokták használni, miért hasznos [általánosan])
     + Adaptive thresholdolások
     + SIFT alkalmazása
* Matlab program bemutatása

Abstract (2 page)

Contents

* Big numbers for chapters, X.Y.Z number format for subchapters (where X,Y,Z nature numbers)

List of Figures

List of Tables

List of the Used Abbreviations

From here, you count the pages! (30 and 50k character)

Abstract (In english and hungarian too)

Introduction

Main theme, work

Conclusion

Until this point!

References

Appendices 🡪 here you can write the code 🡪 enough to give it on CD

Acknowledgements

**Image processing algorithms in microfluidic systems**

1. **Contents**
2. Contents 2
3. Abstract 3
   1. Absztrakt 3
4. Introduction 3
5. Feladat kiírás 4
6. Literature, earlier algorithms 5
   1. Microfluidic in general
   2. Sample introduction
   3. Descrbing the measurement
   4. Microfluidic + image processing
      1. The Saturation algorithm
7. A tervezés részletes leírása 8
   1. Improving the saturation algorithm
   2. The new algorithm
      1. The main idea
      2. Canny edge detector
      3. Adaptive thresholding
      4. Reconnection (Back loading)
      5. Hole filler
      6. Counting
      7. Training set
      8. Introduction of user friendly Graphical User Interface
8. Results, evaluation 13
   1. Improving the saturation algorithm
   2. The new algorithm
      1. The main idea
      2. Canny edge detector
      3. Adaptive thresholding
      4. Reconnection (Back loading)
      5. Hole filler
      6. Counting
      7. Training set
      8. Introduction of user friendly Graphical User Interface
9. Summary 14
10. Acknowledgments
11. References
12. Appendices 14
13. Abstract

The main goal of the thesis is to collect and merge different kind of algorithms giving an effective program for *Trichinella spiralis* detection and counting its based on videos about biomicrofluidic device measurements. Biomicrofluidic device research projects are a part of ‘Lab on a chip’ program, in that the researches and engineers strive to design a small piece of device for some – nowadays one or two – specific detection or segmentation (van-e más felhasználási területe?) problem of biologic samples. One of the leading project in our laboratory is *Trichinella spiralis* detection procedure. *Trichinella spiralis* is a dangerous parasite for animals as well as humans, which most dangerous source for humans is the pork. Avoiding the sales of infected meat products is really important the continuous and reliable controlling at the slaughterhouses, which nine times out of ten does not have a complete and equipped laboratory. For these purposes, our research team tries to give an alternative detection mode, which requires less laboratory instruments, only a microfluidic device, a syringe pump, some plastic tubes, a light microscope with a suitable camera, and a user-friendly detector program, which alerts the controlling person when some pathogen in the sample is. My task was to research the further possible developments of a latter implemented algorithm by Péter Zsíros and combine its results with some other detection methods taking the algorithm more robustness and effectiveness for real-life application. In the interest of reaching these aims, I worked with Canny edge detector, an adaptive thresholding method, a hole-filler, and a new . After determining the meaningful objects, the program measures all of its extents, and divides this number with the average size of the parasites, what was calculated earlier from the training set videos. The method is called as “Divided approach”.

* 1. Absztrakt

1. Introduction
2. Feladat kiírás
3. Literature, earlier algorithms
   1. Microfluidic in general

Microfluidic systems are those kind of systems, what actually do not need a huge volume from the sample just some microliter. Generally, it has a specific structure, which is designed for its function, and thus one device usually fits only for one or two problem. More exactly it is “The science and engineering of systems in which fluid behaviour differs from conventional flow theory primarily due to small length scale of the system.” [1]. These systems functions could be varied ranging from detection to segmentation onto executing some complex biochemistry technique for example Loop-mediated isothermal amplification – LAMP – or Quantitative Polymerase Chain Reaction – qPCR. Engineers, who works on this field has to study a lot of and various kind of subjects as nanophysics, mathematics, biochemistry and biotechnology, because microfluidic bases on these basic disciplines. Whereas nanophysics and biotechnology just in the last few decades has been growing up as high level as microfluidic needs, this is an absolutely new field of the science. The first devices were made in the 1980s, and through the mid-1990 years the developments of these apparatuses including the microvalves, micropumps and microflow sensors actively continued. At this time, the engineers discovered that if they would like to minimize the size, they will need to use external actuators for microvalves and micropumps. This was came from two basic statement: first is that, if we scale down the size, this will indicate the power decreasing of the device by a length scale cubed, so we don’t anticipate as high level activity from microvalves and micropumps as conventional devices have. The second one is the fact, that “The surface-to-volume ratio varies as the inverse of the length scale” [1]. It means that, a large surface has large viscous forces, and often the integrated microactuators are not able to give enough power to micropumps, which should move the fluid in microfluidic systems. First devices manufactured from silicon, but nowadays, remaining this direction for example Polydimethylsiloxane – PDMS, the industry are focusing some more advanced materials as different type of plastics. Nonetheless, the most physical parameters have not changed since the beginnings, which are the followings:

* Small volume [μL, nL, pL, fL]
* Small size [μm, mm]
* Low volumetric flow rate [μL/s, μL/h, ml/s, ml/h]
* …

Principles of function and behaviour (gen.) 🡪 Library **REF**

As microfluidic channels are in micro- and nanoscale range, the physic in this dimension varies from that Newton physic, what we know very well. The usual gravity force and the buoyancy has less influence to the flowing liquid in the systems, in turn the surface forces like surface tension or van der Walls force between particles are more important. Besides this, because we work with fluids, the viscous force is also important, moreover in laminar flowing this is the dominating force. [2] We can describe these flows whit the Reynolds number, which had been declared in 1883 by Osborne Reynolds as the ratio of inertial forces to viscous forces. It is defined as “[3 hivatkozás]”

If Re is much less 2000, viscous forces dominate the flow, so it is laminar, but on the other hand if this number is higher than 2000, the flow turns to be more turbulent. In our laboratory every devices have a low Reynolds number, so in all of these the flow is laminar.

Manufacture technologies (silicon waffer + PDMS device) 🡪 Misi

Fabricate a microfluidic chip is not as difficult thing as for first hearing it seems to. First of all, we need to order a silicon wafer, which contains our previously designed device structure.

Így készültek a mi eszközeink is.

Our device description (function goal + principle) 🡪 Misi

Outlook (what other devices in a lab too, developments) 🡪 Sci-hub **REF**

* 1. Sample introduction

*Trichinella spiralis* 🡪 Net + library **REF**

Preparation 🡪 Ádám

* 1. Describing the measurements

Main steps + equipment 🡪 Ádám + net **REF**

* 1. Microfluidic + image processing

Evaluation earlier + researches

* + 1. The Saturation algorithm

As I mentioned above, the algorithm is not my intellectual product. I just worked with it and tried to find a solution for its weaknesses. The main idea of it is distinguishing the parasites from other objects on the image. For this purpose, it uses mainly the saturation information of the input image, where the parasites has distinct saturation values. First step of it is the noise reduction, especially defeating Salt&Pepper noise, where the input image is filtered with a 5x5 median filter on every channel. After converting the smoothed image from RGB to HSV, the saturation level is taken out and converted to a grayscale image anticipating the following operation, the Difference of Gaussian. DOG uses two distinct Gaussian filter, and thereafter subtracting one of them from the other convolves the previous image with the production of subtraction. Then, a contrast correction step is performed on Saturation channel. Determining one of the most relevant parameter of the algorithm is the task of next step, where it computes the mean value of the saturation channel, which is immediately subtracted from channel values. After this, Otsu method computes a treshold level and takes a binary mask, what is used on the followings. Then, a morphological opening performed preparing for elimination of small objects. This is the second important step of the algorithm, because at this point it is very parameter dependent. Afterwards, it forms a skeleton from Euclidean distances between the objects, eliminates smaller branches, and filters some random noise from the binary mask. The output of the algorithm is a fused image, what was made by composing the original input image with the final binary mask [final output image].

The parasites counting part bases on this final mask, and after measuring the detected objects area with some built-in Matlab function it counts only those objects, which area size are higher than the previously given number.

This algorithm is not flawless, but it was the first on this theme in our laboratory, and after some consultation with my supervisors, they recommended me to begin thinking about how could I improve this algorithm.

1. A tervezés részletes leírása
   1. Improving the saturation algorithm

As I mentioned in the previous section this algorithm had some weaknesses, so I started to examine it. While I was running the algorithm for different pictures, I discovered that for those pictures, which were derived from the newest measurements of our microfluidic device, the program could detected parasites with really bad efficiency or sometimes could not at all. In the interest of finding the problematic parts of the algorithm, I needed to look deeper into the code. As soon as I have debugged it from line to line, I identified two questionable steps. One of these is the average value subtracting from the Saturation channel before determining the threshold level for Otsu method that makes the binary mask. In particularly not the subtraction itself, but the average value is. The problem with it was the fact that this parameter did not allow too much flexibility for the new images, so it was rigid. Therefore, my supervisors and I assumed that changing the value of this parameter may result better values after subtraction [subtraction image], that giving for Otsu method, the result of computing intra- and inter-class variances might be more accurate, and the method could segment better the foreground and the background pixels from each other. [before Otsu, after Otsu]

The other bigger issue with the program was the magnitude of the small size areas. It was also a fix parameter of the algorithm, and thanks to this attribution, the program could not follow as good the change of inputs as we expected. Because the problem was similar like in average value case, therefore the assumption to solve it would have not been different: if we could induce some improvement on the output binary mask by changing this parameter value, we would make the algorithm more resistant for input change.

The best recording equipment is not enough for taking good quality images. We have to know very well the software and hardware too, what we use under recording, that means we need to understand the most important features and setting parameters. My research in the theme began at this point. To understand, how does the above-mentioned uEye Cockpit software and microscope work together, I had to take some test measurements with its. Based on my tests the following parameters were the most interesting:

1. Microscope

* Aperture level
* Light intensity level

1. Software

* Auto-contrast on/off mode
* Auto-white balance on/off mode

Our plan with these measurements was to improve the results of a previously implemented parasite detection algorithm – the so-called Saturation algorithm, which have been made by Péter Zsíros [6]. The algorithm was super and functioned well, but it was not enough for us. We would have like to develop it to reach higher efficiency in parasite detection than it have been until that time. In order to achieve our goals, I had to make a uniformly illuminated image composition by modifying the light intensity level and the aperture level at the same time. Auto contrast and auto white balance switches turned on/off also for the same reason, reaching that hypothetical light intensity equilibrium state.

Besides these improvement steps, we were thinking about new opportunities, algorithms, which would have used other information from images.

* 1. The new algorithm
     1. The main idea

Since the previous algorithm used only saturation information and we would have like to achieve higher detection rate, we needed to find an algorithm that could work on other information segment of input images. Therefore, the main idea was a kind of size measure algorithm, which firstly defines the objects shapes, then with a hole filler method it fills the objects along their edges, and finally measure the objects sizes. Based on our theorem, to count how many parasite is on the image, we need to define the size of one parasite based on measurements, then with this average parasite value we divide the whole measured area. We named this algorithm, and it got the Divided approach name after the division, that it uses.

* + 1. Canny edge detector

Based on my researches and earlier knowledge from image processing algorithms subject, I made a decision next to Canny edge detector.

This algorithm is one of the most known and widely used object detector in the art. It had been developed by John F. Canny in 1986 to satisfy the following requirements and give better solution than previous edge detection methods:

* Catch as many real edges as possible
* Do not detect false edges, which might arise from image noise
* Detected edges be as close to real edges as possible
* The detection be separate from edge direction - isotropic

Four different part builds it up, but via linking these parts it could get pretty good outcome for well-defined images. In addition, this algorithm is not too costly and easy to implement. The first two advantages were the reason why I choose this, and build it into my own algorithm.

The four main step of the algorithm are the followings:

1. Noise reduction

The incoming image is convolved with a Gaussian kernel reducing the noise of the image, which may come from different sources. [Noise reduction kép]

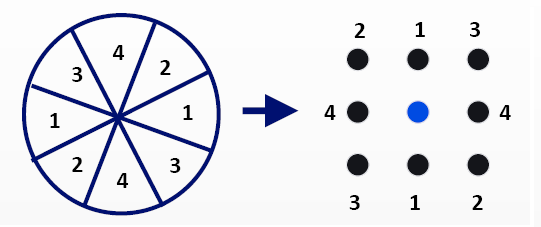
1. Gradient intensity and direction calculation

The algorithm calculates the gradient intensity and its direction to emphasize the edges on the input image.

,

1. Non-maximum suppression

All edges are categorized into for different group – 0o, 45o, 90o, 135o – based on their directions. The selection method will put every edge to the appropriate numbered group, if its direction is between the two boundaries of the group for example: 67o goes to group 2, because its value is more than 45o, but less than 90o. The groups does not finish at 180o, but goes further continuously for second half of the circle until 0o by the corresponding way with the same differentiations between the boundaries. [kör számos kép]. The goal of the categorization is to supress the edges forming the skeleton of the objects. Namely, if the magnitude of an edge is not greater than magnitude of its two neighbours of the gradient direction, its value will set to zero. [9 pontos kép]



[kör számos kép + 9 számos kép] [7]

1. Hysteresis thresholding

In this step the algorithm solves the “one threshold” problem, which claims that if the threshold level is too low, so many invalid edges will appear, however if its level is too high, true edges will disappear. This two threshold system is working by the following way:

* + 1. Adaptive thresholding

Canny looked like good for our problem, but to make it more effective for our inputs I needed to develop it with some easier methods. One of the two method is the adaptive thresholding. This computes the variance of every point of the gradient image, and if the value of one point’s variance is lower than the maximum variance value divided by a freely given number, the method will suppress this point by setting its value to zero. If it is higher, the algorithm will conserve its value as it was before. For the sake of simplicity, I left this freely given number as a fix parameter on the code.

This algorithm extension step is applied to solve the device structure discarding problem, where my goal was to remove the basic structure of the device from the image.

Improving the intermediate information of Canny just was the first part of this upgrading, because of I had to link the object’s lines again, which in most cases were became dashed in Non-maximum suppression or Hysteresis thresholding steps.

* + 1. Reconnection (Back loading)

In this method, I tried to solve the problem the later problem, namely how to make continuous line along the edges of the object for the reason that I could fill these in the Holler filler step. It looked like more difficult than what it became for the end, because in a lot of cases the Non-maximum and especially the Hysteresis steps teared apart the weaker parts of the edges from the stronger ones, hence the planned Hole filler method could have not filled the objects. Solving this problem, I implemented a really easy function, that goes along the whole image matrix, and if a pixel value equals with zero, it looks at their 8 pixel neighbourhoods between one pixel away. If it finds one of these, which value is greater than zero rewrites the original point value with the value of this neighbour.

* + 1. Hole filler

On this step bases mainly our divided approach idea. The input of this method is the output of the extended Canny edge detector, that’s last method is the previously discussed reconnecting method. Thank to built-in imfill() Matlab function the implementation was effortless. The only thing that I had to solve the white frame along every image margin, what was induced after the Gradient calculation, The removal of this inappropriate frame was really crucial, because if I could have not cutted it out, the method could have not filled the objects. In order to erase it, the method goes along the binarized image, and changes the pixel values of it to zero.

* + 1. Counting

The counting method’s main point is the predetermined average parasite value. If it is appropriate, the function could get good numbers.

* + 1. Training set
    2. Introduction of user friendly Graphical User Interface

1. Results, evaluation
   1. Improving the saturation algorithm
   2. The new algorithm
      1. The main idea
      2. Canny edge detector
      3. Adaptive thresholding
      4. Reconnection (Back loading)
      5. Hole filler
      6. Counting
      7. Training set
      8. Introduction of user friendly Graphical User Interface
2. Summary
3. Acknowledgments
4. References

[1] Fundamentals and application of microfluidics, zöld-vörös borítós

[2] Fainman article

[3] Reynold number: <https://en.wikipedia.org/wiki/Reynolds_number#History>

[4] Parazitás képre 🡪 D. Van den Eden, "Illustrated Lecture Notes on Tropical Medicine - Tissue nematodes", *Itg.author-e.eu*, 2013. [Online]. Available: http://itg.author-e.eu/Generated/pubx/173/helminthiasis/tissue\_nematodes.htm. [Accessed: Apr- 2013].

[5] Peti Bsc-s szakdogája

[6] Zsíros Péter Attila, „Optikai úton történő részecskedetektálás mikrofluidikai eszközökben”, Diplomaterv, ITK, PPKE, Budapest, Magyarország, 2016

[7] Dániel Szolgay. Class Lecture, Topic: “2D Convolution” Neumann Lecture Hall, Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Sept. 20, 2016.

1. Appendices

**Literature**

1. Microfluidic (**intro**, **history**, its mechanic + some function, manufacture methods, laboratory applications + our specific device’s functionphysic + useless + outlook)
2. Sample introduction (trichinella spiralis + preparation)
3. Describe the measurement (
4. Microfluidic + image processing (hypothetical + real applications generally + in laboratory – Peti)
   1. Hypothetical + real applications generally 🡪Sci-hub **REF**

**A tervezés részletes leírása:**

1. Improving the saturation algorithm
   1. Detecting weak parts
   2. Microscope setting probes
2. The new algorithm
   1. The main idea
   2. Canny edge detector
   3. Adaptive thresholding
   4. Reconnection (Back loading)
   5. Hole filler
   6. Counting
   7. Training set
   8. Introduction of user friendly Graphical User Interface
3. Microscopic setting probe measurements with uEye camera

Improving the incoming images for Saturation algorithm

1. Saturation algorithm – detection algorithm 1.
   1. Improving the Saturation algorithm
2. Canny edge detector – detection algorithm 2. + modifications
3. Hole filler
4. Training set – how do I select, collect images
   1. Categories
5. GUI – introduction, utilities, benefits

**Evaluation:**

1. Microscopic setting probe measurements with uEye camera
   1. Improving the incoming images (only the results)
2. Saturation algorithm – detection algorithm 1.
   1. Assuming this, I began to modificate the constant, and I got a moderate improvement on the output images.[ip\_find\_result and modifikált result]
   2. Beginning to define better values, I also tried to alter it, and got some kind of improvement on the same input images.
3. Canny edge detector – detection algorithm 2. + modifications
   1. As our images is made by constant lighting with fixed static camera the .
   2. Adaptive thresholding
      1. Cleaning small objects from image
      2. Base removal
      3. Luminosity robustness (rea)
   3. Back loading
4. Counting – evaluate the results, explanation + future outlook (development)
5. Training set
   1. How do I select, collect images 🡪 ennek sikeressége
6. Comparison the 2 algorithm
   1. Based on results
7. GUI – introduction, utilities, benefits (based on 50 person’s feedback)

**Results:**

**References:**

[

**Piszkozat:**

**Abstract:**

* This program is implemented by me, but one part of it is developed by a graduated student, Péter Zsíros.
* The detector program could also help the researches of the Biomicrofluidic laboratory to improve the devices. was built up from different type of shape or edge detection algorithms, which gives a good and plausible solution enhanced each other benefit parts and reduced their disadvantages.
* could be substituted simply by a parasite filtering microfluidic device, a light microscope with a, and an attached user friendly detector program, what is able to detect the parasite.
* This program could help the development of biomicrofluidic device, and in the future will may attach to device as a friendly user program. The plan of device + program combination is to detect/manifest these pathogen parasites more rapid and simply way in the slaughterhouses avoiding to market some infected pork.
* The detection and counting helps device development and could be improved
* Merging algorithms allows us to configure a

**Introduction:**

**Algorithm:**

That means Namely, if we could give a better value for subtraction than it, This We had an assuming that if could Based on some test, we had an assuming that if we could find a better the average value for something better

* 1. Otsu method basically tries to separate
  2. This is an important step, because if we could give a better separated image to the and some modification on it might result
  3. This is the first point, where the algorithm is parameter dependent, but the only one.
  4. These two parameter dependent step were my main goals about this program