Image processing algorithms in microfluidic systems

Témabejelentő oldal helye

Hallgatói nyilatkozat

Alulírott Kufcsák Péter, a Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Karának hallgatója kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, és a szakdolgozatban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a Szakdolgozatot más szakon még nem nyújtottam be."

Budapest, 2017. 12. 07.

_____

Aláírás

1. Contents

2. 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 these extents, and divides this number with the average size of one parasite, that was calculated earlier from the training set videos.

2.1. Absztrakt

3. Introduction

4. Feladat kiírás

- Kutassa föl és tekintse át a témához kapcsoló szakirodalmat (*Trichinella spiralis* tulajdonsága, kimutatására alkalmas, és a mérési beállításokhoz használt eszközök megismerése, korábbi algoritmusok vizsgálata a témában, tovább fejleszthetőségi lehetőségeik vizsgálata)
- Dolgozzon ki egy detektáló algoritmust patogén paraziták számlálása céljára. (célja az eddigiektől eltérő megoldás megtalálása)

- Ismerje fel és számlálja meg a parazitákat a mérési felvételeken. (quantitative hány darabot és milyen hatékonysággal sikerül azonosítani)
- Különítse el a felvételen látható egyéb zavaró objektumokat a parazitáktól. (rendszerbe kerülő kosz illetve egyéb zavaró hatások ne zavarják meg a kiértekelés folyamatát)
- Vizsgálja meg az algoritmus további fejleszthetőségének lehetőségeit. (célja, hogy a jövőben hatékonyabbá tehető legyen a detekció)
- Valósítson meg egy felhasználóbarát kezelési felületet az algoritmus számára. (parazita mérések kiértékelésének javítására könnyen kezelhető program implementálása)

5. Literature, earlier algorithms

5.1. 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]"

$$Re = \frac{\rho u L}{\mu},$$

$$where\ \rho\ is\ the\ density\ of\ the\ field\ \left(\frac{kg}{m^3}\right),$$

$$u\ is\ the\ velocity\ of\ the\ fluid\ with\ respect\ to\ the\ object\ \left(\frac{m}{s}\right),$$

$$L\ is\ a\ characteristic\ linear\ dimension\ (m),$$

$$\mu\ is\ the\ dynamic\ viscosity\ of\ the\ fluid\ \left(Pa * s\ or\ \frac{N * s}{m^2}\ or\ \frac{kg}{m * s}\right),$$

$$so\ this\ a\ dimensionless\ number.$$

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**

5.2. Sample introduction

Trichinella spiralis → Net + library **REF**

Preparation → Ádám

5.3. Describing the measurements

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

5.4. Microfluidic + image processing

Evaluation earlier + researches

5.4.1. 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 average 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.

6. A tervezés részletes leírása

6.1. Improving the saturation algorithm

As I mentioned in the previous section this algorithm had some weaknesses, therefore I started to working on 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, 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.

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 the case of average value, 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.

In order to clear up which part of the measuring equipment is responsible mostly for the former mentioned problem, I needed to make some test measurement with the microscope camera and the microscope itself too. Based on my tests the following parameters were the most interesting:

I. Microscope
   - Aperture level
   - Light intensity level

II. Software
   - Auto-contrast on/off mode
   - Auto-white balance on/off mode

In the interest of 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.

6.2. The new algorithm

6.2.1. Divided approach

Since the previous algorithm used only saturation information, but 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, our main idea was a kind of size measuring algorithm, which firstly defines the shape of objects, then with a hole filler method it fills the objects along their edges, and finally measures the size of objects. Based on our theorem, to count how many parasite is on the image, we need to define the size of one parasite from 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 division, that it uses.

6.2.2. 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]

2. Gradient intensity and direction calculation

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

$$d^x = \left(\frac{\partial f}{\partial x}\right) \qquad d^y = \left(\frac{\partial f}{\partial y}\right)$$

$$d = \|\nabla f\| = \sqrt{(d^x)^2 + (d^y)^2}$$

$$\theta = \tan^{-1}\left(\frac{d^x}{d^y}\right),$$

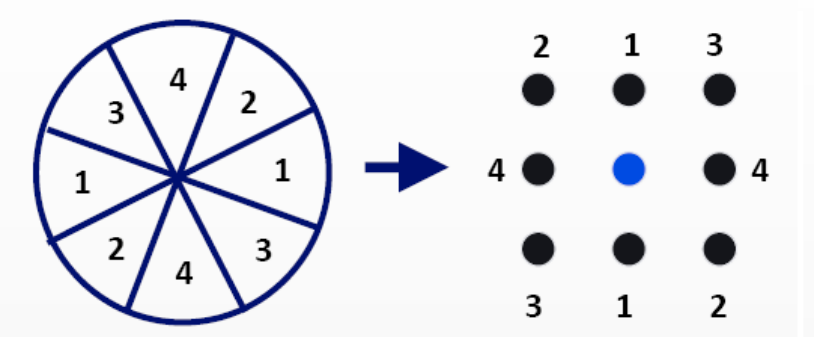
where d^x és d^y *the horizontal and vertical derivatives of the image,*

d *the gradient,*

θ *the direction vector of the gradient*

3. Non-maximum suppression

All edges are categorized into four 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]

4. 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:

$$d(i,j) \geq t_1, then\ (i,j)\ pixel\ is\ a\ real\ edge$$

$$d(i,j) < t_2, then\ (i,j)\ pixel\ is\ not\ a\ real\ edge$$

$$t_2 \leq d(i,j) < t_1, then\ (i,j)\ pixel\ is\ a\ real\ edge,$$
$$only\ if\ one\ of\ its\ negihboors\ has\ the\ same\ direction\ as\ it$$

6.2.3. 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 let 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.

6.2.4. Reconnecting the edges

In this method, I tried to solve the latter 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.

6.2.5. Hole filler

Our divided approach idea bases mostly on this step. 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.

6.2.6. Counting

The major point of the counting method is the predetermined average parasite value. If it is well defined, the function could get good numbers. For that reason, I made a training set on 57 well-selected images, and ran my algorithm on it. I hoped that using good threshold parameters for Canny edge detector, I might get that desired "well-defined" average area size. Beside this, I also had another aim with these runnings. Namely, the size of the parasites is variety, and next to average value I would have like to give a range, which mean value is the average itself. This interval might be useful at the future parts of algorithm development too.

6.2.7. Training set

My training set was built on by the following guidelines:

- How many parasites is on the image?
  - Based on this I made 3 different group
    i) A lot of group, where the number of parasites is more than 15 [kép]
    ii) Moderately lot group, where the number of parasites is more than 8 but less than 15 [kép]
    iii) A few number of group, where the number of parasites is less than 8 [kép]
- Which kind of artefacts is on the image?
  - Gomba(?) group, where some … is on the image [kép]
  - Other kind of artefacts for example dirty things(?) [kép]

My target with this six different type of group was to prepare my algorithm for as many kind of problems and artefacts as I could.

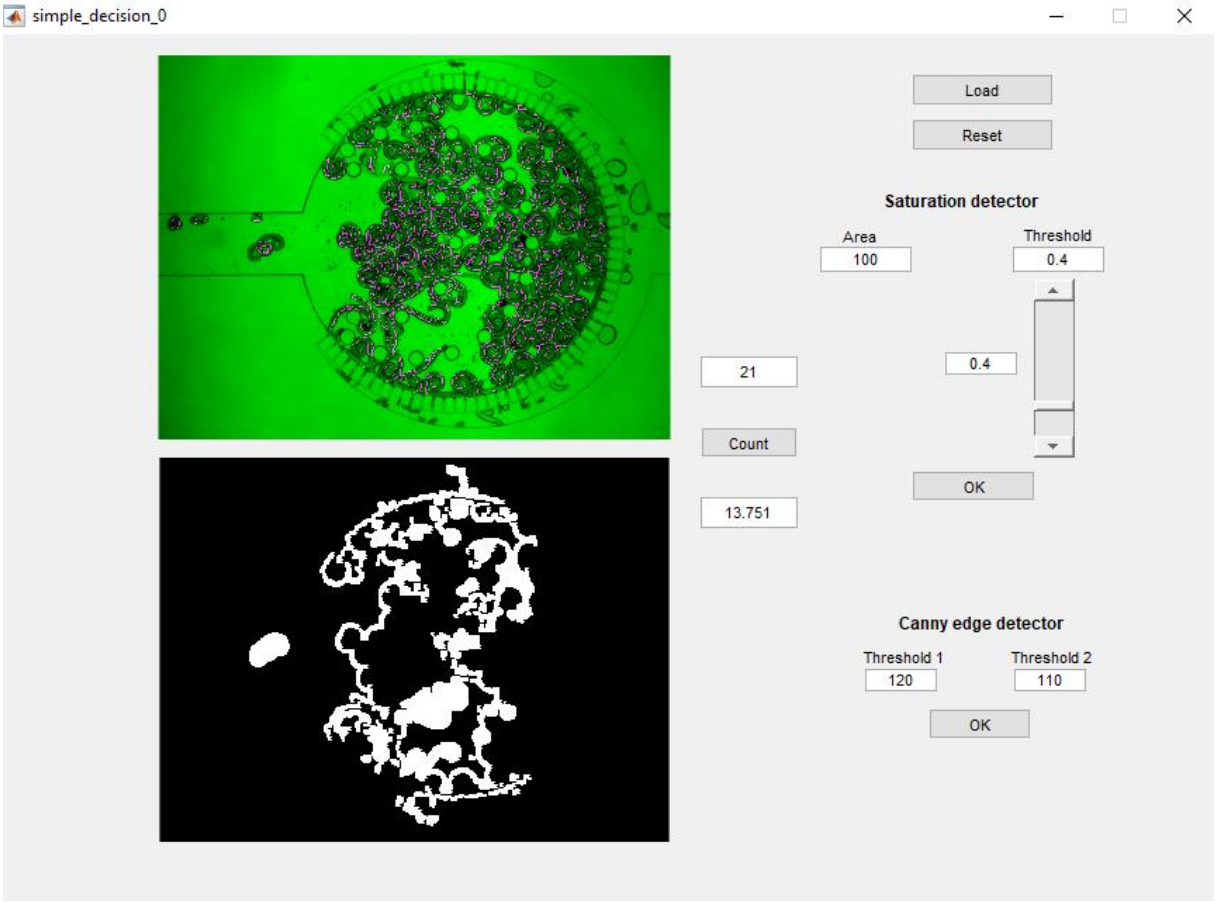
6.2.8. Introduction of user friendly Graphical User Interface

Graphical User Interface is commonly used to make a program available for everyone, without any deeper computer knowledge. My GUI was made for this purpose, but also to compare my algorithm with the improved saturation algorithm.

It contains upper a Load and a Reset buttons, which can load a new image or restore the original one after modifications. Under these two buttons, are located the Area editor for eliminating of small area size objects, and the Threshold editor to multiply its value with the average value of the saturation channel pixel values (when it is 1, we get back the average value). The Slide has the same function as Threshold editor.

Below this section are located the two Threshold editor for Hysteresis thresholding step of Canny edge detector.

Left side I placed two figure panels, where the user could see the results of the algorithms in real time responding for the parameter alterations. A bit left from these figures is the Counting section visible, where we could count by two different counting functions the two different results of the algorithms.



9. figure The GUI helps testing the algorithms.
Left side we could see the results, and right side we can modify the parameters,
or loading another image.

7. Results, evaluation

7.1. Improving the saturation algorithm

I made a lot of measurement with all of the possible variations of the four main parameter. From these, it is concluded that under the forth intensity level of the microscope we could not get normal, usable images. Moreover, we have to leave intensity on the sixth level. We had to drop our intention to make a uniformly illuminated image composition out, because neither the aperture, nor auto-contrast, nor auto-white balance parameter changes did not produce the desirable uniformly illuminated image.
The case of determining a more appropriate area size parameter for the elimination of small objects looks like also an unrealistic dream, because the measurement did not get so well results for the problem. Due to, I could not identify exactly these two parameters, what I would have like to, I let these on the GUI for the user giving a chance them to modificate these as they needed.
Nonetheless, I could not solve these in this paper, I give recommendation for these values.[1. table]

7.2. The new algorithm

7.2.1. Divided approach

Our idea about the divided approach algorithm is proved, because it is working. This makes what we would have like to, detecting and counting the parasites on the newly measured images. The test results have been averaged for every group of the training set, so this is the reason why is only one value for each column visible. [2nd table]. Moreover, based on the result we could see that it could not performed really good on those images, where a lot of parasites were on the screen –only with 53.87% of all, but for those, where the number of parasites was between 8 and 15, it finds 95.37% of the them. [2nd table]
I am going to publish my counting results in the Counting section.
[10th figure about the input and well output image]

7.2.2. Canny edge detector

The algorithm did what I have expected in advance, but not for the first time, so I had to make some changes on it. First of all, the final result after Non-maximum suppression and Hysteresis steps were not acceptable, ergo it did not give as good result as I though.[bad pictures with normal kifejtéssel]. The problem was that when I used for both of them the normal edge direction categories and boundaries, it eroded the parasites edges, and I could not use the hole filler method. But the turn was coming, when I did not anticipate for it, namely when I used for Non-maximum suppression step the given edge boundaries and groups, however for Hysteresis step I used a normal scaled but differently limited edge categorization groups. So, Non-maximum step categorized edges into 0^O, 45^O, 90^O, 135^O groups, but

Hysteresis categorized the edges by 22.5^{O}, 77.5^{O}, 112.5^{O}, 157.5^{O} group boundaries. In total this little bit changes resulted a far better edge detection than it was before. I do not know what is the exact explanation of this, but because of it worked, I used it.
On of the most sensitive part of the algorithm is giving the appropriate threshold values. To solve this problem I tried to define two fix parameter values, but because of these were always differentiated for each strongly different input pictures, I let the fitting for the user too like in the case of threshold level at saturation algorithm.
As in the last, for these parameters I also gave some recommendation. [1. table]
[11th figure about Canny from step to step]

7.2.3. Adaptive thresholding

This method should have eliminated the basic structure of the filter device, before Non-maximum step from the Gradient image. The only problem with it was the fact that it always needs a parameter, which modifies the threshold value. I could determine a fix – a=20 – value for this parameter, but frankly it did not do, what I expected from it several times. So, this part after tests is not unusable, but just in really few cases could give some sensible solution for the problem.
[12th figure about before after pictures]

7.2.4. Reconnecting the edges

Since the results have improved after the upgraded Hysteresis step, but the lines still have not formed a continuous line, what we needed for Hole filler. The method what I implanted in this step solved this problem, but just partially. To be honest, I have not expected to much about this step, only just to help a bit for the resized Canny edge method, where the lines is connected mainly. So, this part made for what it was planned, and thus I built it in my code.
Increasing the number of neighbourhood along the pixel might provide better results, and in the future, I would like to measure its effects too.
[13th figure about before after pictures]

7.2.5. Hole filler

Based on the tests, this part also gave back, what I anticipated from it, filled the parasites creating opportunities to measure these sizes. The removal of the white frame with the double for cycle was not the most sophisticated solution for this problem, but is used. Nevertheless, in the near future I will look after what can cause this failure on the image, I will repair it.
[13th figure about before after pictures]

7.2.6. Counting

After running the Divided approach algorithm for my training set, which contains 57 input images categorized into six different group the average value of the size of the parasite is approximately 906 pixel. I got this number as calculated the mean value of the average parasite size(s) of every input images.[1st row of the 3. table] It sounds a bit confused, but let me explain the calculation method only for one input.

So, after I ran the full Canny edge algorithm extended it with the resizing, the adaptive thresholding, the edge reconnection, and the hole filler methods for the incoming image I used a built-in Matlab function, bwareafilt(), and with it I kept only the *n* largest objects. I set up how big would be *n*, based on how many parasites and how many non-parasites objects could the algorithm detected. If in some cases the algorithm identified more non-parasite object than parasite, I decreased this number, however it could catched mainly parasites, I increased the parameter. [3. table]

In order to, getting more precisely number, I determined the threshold values of Hysteresis thresholding manually for every training set image. Then, I used the bwconncomp() function to get how big is my objects on the screen. Sum up these pixels, I got the whole pixel area of the remaining objects. Finally, I divided this number with the real number of parasites that was counted manually from the original input image by me.

Besides, to calculate the average value, I would have like to give an upper bound and a lower bound, among which the expected value of the size of parasites would have been. In order to compute these boundaries I tried with the variances of sizes, but this did not give back so good results. Instead of this, I calculated the absolute differences of the average value of every input from the average value of average values. Sum up these differences and I made the mean value of these, so I got the mean differences value of the average values of input images from the average of average values. This was an important number, because with it now I have already counted the lower and upper bounds from subtract or add this difference for the average value of average values. [4th and 5th row of the 3. table] My plan was to build these boundaries into my code, instead of *n* parameter as a filter range for bwareafilt(), with it the function could have removed every object from the screen, which would have not been in the interval. After the tests my attempt to use these looked like dead, because of after the detection algorithm, the objects were attached to each other, and in theory their size per unit was between the range, but in reality they were defining together them area value, and thus they were out of range. Therefore, I set the upper boundary to infinity. On the followings, I would like to solve this failure.

7.2.7. Training set

With the six basic group, I could easily train my algorithm, but in the future, I would like to extend it for more and more images.

7.2.8. Introduction of user friendly Graphical User Interface

Whereas, the saturation algorithm has become more parameter dependent thanks to my modifications, and my algorithm have required two threshold value as parameter too, this graphical implementation was proved to be indispensable, because during the tests altering the parameters and running the algorithms were much easily as if I tried to do it in Matlab workspace.

7.3. Comparison of the two algorithm

8. Summary

A jövőben a két algoritmus eredményeit szeretném egyszerre hasznosítani.

9. Acknowledgments

10. References

[1] Nam-Trung Nguyen, Steven T. Wereley. *Fundamentals and Applications of Microfluidics*. Norwood, USA: Artech House, 2006, pp. 5-15.

[2] Sindy K.Y. Tang and George M. Whitesides. "Basic Microfluidic and Soft Lithographic Techniques" in *Optofluidics: Fundamentals, Devices, and Applications*, edition, volume. Yeshaiahu Fainman, Ed. Publishing location: Publishing company, 2009, pp. 7-31.

[3] Pilnam Kim, Keon Woo Kwon, Min Cheol Park, Sung Hoon Lee, Sun Min Kim and Kahp Yang Suh. "Soft Lithography for Microfluidics: a Review". *Biochip Journal*, Vol. 2., pp. 1-11, March 2008.

[4] Reynold number: https://en.wikipedia.org/wiki/Reynolds_number#History

[5] 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].

[6] Zsíros Péter Attila, „Parazitaszámlálás mikrofluidikai képfeldolgozás segítségével", Szakdolgozat, ITK, PPKE, Budapest, Magyarország, 2015

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

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

11. Appendices

	Small size area parameter	Saturation threshold	Hysteresis threshold 1.	Hysteresis threshold 2.
Dirty				
Funghi				
Sok				

Közepes				
Kevés				
Random				

1. table Recommendations for optimal parameter values

	Number of counted parasites	Number of real parasite	Efficiency (%)
Dirty	5.87208	4.6	78.34
Funghi	3.550522	3.4	95.76
Sok	15.02865	27.9	53.87
Közepes	8.77374	9.2	95.37
Kevés	4.099253	3.1	75.62
Random	4.172177143	5.571428571	74.89

2. table Relative efficiency of Divided approach algorithm based on 57 images.

The groups efficiency values is the mean value of every image efficiency in the group.

Average of average parasite sizes	960.62
Minimum of average parasite sizes	178.39
Maximum of average parasite sizes	2845.00
Average of abs differences of average parasite sizes	411.52
Parasite size interval (upper and lower bounds)	549.10 - 1372.15

3. table ?