CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF CYBERNETICS

VESSEL DETECTION AND RED BLOOD CELLS VELOCITY ESTIMATION ALGORITHM FOR MICROCIRCULATION ANALYSIS

MASTER´S THESIS

Petr Kočiš

Thesis supervisor:
prof. Dr. Ing. Jan Kybic

Consultant:
MUDr. Ing. David Macků

May 2019

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kočiš Petr** Personal ID number: **405357**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Biomedical Engineering and Informatics**

Branch of study: **Biomedical Engineering**

II. Master's thesis details

Master's thesis title in English:

Vessel Detection and Red Blood Cells Velocity Algorithm for Microcirculation Analysis

Master's thesis title in Czech:

Algoritmus na detekci cév a určování rychlostního profilu erytrocytů pro analýzu mikrocirkulace

Guidelines:

Create a software for automated analysis of small vessels videos from the MicroScan device for microcirculation measurement. The software should find the vessels in the picture, measure their size and estimate the statistical distribution of red blood cells velocities. Learn about standard descriptors of microcirculation (Proportion of perfused vessels (PPV), deBacker score, Total vessel density (TVD), Perfused vessel density (PVD), Microvascular flow index (MFI), flow heterogenity index (FHI)), create a method to compute them, and compare the outcomes with the information from the software, which comes with MicroScan. Furthermore, create new descriptors based on the statistical distribution of velocities in small, medium and big vessels. Create a classifier which will distinguish (on the base of the descriptors mentioned above) the physiological microcirculation (measured by the entering and leaving the hospital department) and expected pathological microcirculation (measured during cardiopulmonary bypass). Test the classifier, assess its accuracy and reproducibility, as well as the suitability of the descriptors.

Bibliography / sources:

[1] Can Ince et al.: Second consensus on the assessment of sublingual microcirculation in critically ill patients: results from a task force of the European Society of Intensive Care Medicine. Intensive Care Med (2018) 44:281-299
[2] B.K.P. Horn and B.G. Schunck, "Determining optical flow." Artificial Intelligence, vol 17, pp 185–203, 1981
[3] A. Bruhn et al: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. IJCV (2005), 61(3), 211-231
[4] Duda, Hart: Pattern Classification. Wiley
[5] E. Türetken; F. Benmansour; B. Andres; P. R. Glowacki; H. Pfister et al. : Reconstructing Curvilinear Networks using Path Classifiers and Integer Programming; IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016. DOI : 10.1109/Tpami.2016.2519025
[6] E. Türetken; G. González Serrano; C. Blum; P. Fua : Automated Reconstruction of Dendritic and Axonal Trees by Global Optimization with Geometric Priors; Neuroinformatics. 2011. DOI : 10.1007/s12021-011-9122-1
[7] Gonzalez G, Aguet F, Fleuret F, Unser M, Fua P (2009) Steerable Features for Statistical 3D Dendrite Detection. In: Conference on Medical Image Computing and Computer Assisted Intervention, vol 12, pp 625–32
[8] http://www.microvisionmedical.com/
[9] Scikit learn Python library, scikit-learn.org

Name and workplace of master's thesis supervisor:

prof. Dr. Ing. Jan Kybic, Biomedical imaging algorithms, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **26.11.2018** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until: **30.09.2020**

prof. Dr. Ing. Jan Kybic Supervisor's signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department's signature	prof. Ing. Pavel Ripka, CSc. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

ABSTRACT

The monitoring of microcirculation becomes an attractive tool for evaluation of state of health of critically ill patients. Microcirculary alternations have been observed by patients with sepsis or organ failures. Those parameters, which describe number and distribution of vessels (TVD, deBacker score) can be assessed automatically, but the properties of blood flow (MFI, FHI) have to be determined manually. The goal of the thesis is to develop a software for automatic analysis of blood flow velocity from SDFI (sidestream dark field imaging) microcirculation videos.

The software contains automatic correction of video instability, automatic vessel segmentation, measurement of vessel dimensions, two traditional (TVD, deBacker score) and two novel (optical flow descriptor (OFD), pixel intensity fluctuations (PIF)) descriptors.

The function of software was verified by comparison of deBacker score values with those, which were determined by commercial software. The average difference was 25 % for deBacker score for all vessels and 22 % for deBacker score for small vessels. OFD estimates optical flow, the estimation is correct up to the velocity of 17 pixels/frame in artificial videos. However, it is not suitable for real data assessment, because it estimates identical values in vessels with visually different blood flow velocity. We analysed the videos with both physiological and pathological (patients on CPB) microcirculation and used the data to create a classifier of pathological microcirculation. We found a significant difference in three from 16 (each of four descriptors was computed for the vessels in four different ranges of diameter) parameters (deBacker score, p=0.035), (PIF for all vessels (p=0.034), PIF for large vessels (p=0.001).

The main outcome of the thesis is the software for automatic analysis of microcirculation, which provides two conventional and two novel descriptors. We found out, that software enables to classify pathological microcirculation. However, just a small dataset of 151 videos was used to train and test the classifier. We discovered, that our implementation of optical flow is not suitable for blood flow velocity estimation, probably due to unsufficient contrast and texture and presence of noise.

We developed the software for automatic microcirculation analysis. It describes number and distribution of vessels and blood flow.

Key words: microcirculation, optical flow, automatic vessel detection

ABSTRAKT

Monitorování mikrocirkulace je stále více považováno za významný nástroj při hodnocení zdravotního stavu kriticky nemocných pacientů. Změny v mikrocirkulaci lze sledovat například u pacientů se sepsí či orgánovými selháními. Zatímco parametry, které popisují množství a rozmístění cév (TVD, deBacker score) jsou určovány automaticky, deskriptory toku erytrocytů (MFI, FHI) jsou zatím určovány manuálně. Cílem práce je vyvinout software na automatickou analýzu mikrocirkulačních videí získaných SDFI (sidestream dark field imaging) technologií, přičemž důraz bude kladen na hodnocení toku krve, především rychlosti.

Základem softwaru jsou stabilizace videa, automatická segmentace cév, měření rozměrů cév a tradiční deskriptory mikrocirkulace, jež lze určit automaticky. Funkce zmíněných implementací byla ověřena porovnáním hodnot tradičních deskriptorů s hodnotami, které byly získány analýzou komerčním softwarem. Hodnoty se lišily v průměru o 25 % (deBacker score) a 22 % (deBacker score pro malé cévy). Následně byly implementovány dva nové deskriptory toku, jeden založený na odhadu optického toku (opical flow descriptor (OFD)), druhý na sledování časových fluktuací pixelových intenzit (pixel intensity fluctuations (PIF)). Z experimentu na umělých datech vyšlo najevo, že OFD určuje správně rychlost do 17 pixelů/snímek. Pro popis reálných videí se však nehodí, protože určuje totožnou rychlost u videí s vizuálně odlišnými rychlostmi toku. Software byl použit na analýzu souboru videí, který obsahoval fyziologickou (pacienti v normálním stavu) a patologickou mikrocirkulaci (pacienti na mimotělním oběhu). Data z analýzy byla použita na vytvoření klasifikátoru obou typů videí. Významný rozdíl se potvrdil u tří z 16 hodnocených parametrů (každý ze čtyř deskriptorů byl určen pro čtyři skupiny cév dělené podle velikosti), a to jak u tradičního (deBacker score, p=0.035), tak u nového deskriptoru (PIF pro všechny cévy (p=0.034), PIF pro velké cévy (p=0.001)).

Hlavním výsledkem práce je software pro kvantitativní automatickou analýzu mikrocirkulace, který poskytuje dva tradiční (TVD a deBacker score) a dva nové (PIF, OFD) deskriptory. Experiment prokázal, že na základě PIF lze klasifikovat fyziologickou a patologickou mikrocirkulaci. Úskalím experimentu je malý objem dat (151 videí). Zjistili jsme, že naše implementace optického toku není vhodná pro určování rychlosti toku červených krvinek ze SDFI videí, nejspíše kvůli nedostatečnému kontrastu, nedostatečné textuře uvnitř cév a přílišnému šumu.

Výsledkem práce je software na automatickou kvantitativní analýzu mikrocirkulačních videí, který kromě stávajících parametrů jsou rozmístění či plocha cév, popisuje také tok krve.

Klíčová slova: mikrocirkulace, optický tok, automatická detekce cév

Author statement for undergraduate thesis:

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date.................... ....

signature

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor prof. Dr. Ing. Jan Kybic for his helpful approach and valuable advice concerning the thesis. I would also like to acknowledge MUDr. Ing. David Macků for his helpful advice and for providing the data on which the thesis is built.

LIST OF ABBREVIATIONS

TVD	total vessel density
PPV	proportion of perfused vessels
PVD	perfused vessel density
MFI	microvascular flow index
HVM	hand-held vital microscope
CPB	cardio-pulmonary bypasss
ECMO	extracorporeal membrane oxygenation
IABP	intra-aortic balloon pulsation
OPS	orthogonally polarized spectral imaging
IVM	intravital microscopy
SDF	sidestream darkfield imaging
RBC	red blood cells
IDF	incident darkfield imaging
FHI	Flow heterogeneity index
DB	deBacker score
STD	space-time diagram
PIV	particle image velocimetry
OFD	optical flow-based descriptor
PIF	pixel intensity fluctuations
FR	frame rate
fps	frames per second
SVM	support-vector machine
roi	region of interest
GUI	graphical user interface

LIST OF FIGURES

TABLE OF CONTENTS

1 Introduction

Microcirculation has been lately widely discussed, because it has been demonstrated that the behaviour of blood flow is related to many diseases, such as septic shock, cardiac disease or diabetes. [1] The automatic evaluation of number or distribution of vessels in microcirculation videos is common, but the reliable estimation of blood flow velocity is a challenging problem. Our work is dedicated to the software for automatic microcirculation analysis, where both traditional and novel descriptors are determined.

1.1 Microcirculation

By microcirculation we understand blood perfusion in vessels with smaller diameter than 100 µm. It is the primary site of oxygen and nutrient exchange. In addition to delivering nutrients and removing waste products essential for moment to moment function, microcirculation plays an essential role in fluid exchange between blood and tissue, delivery of hormones from endocrine glands to target organs, bulk delivery between organs for storage or synthesis and providing a line of defence against pathogens. [1,2]

Physiological microvascular net is dense and perfusion is continual and homogenous. The velocity of blood cells depends on diameter of vessels and ranges between 0.5-1.5 mm/s [19].

A real time video of microcirculation can be captured, which means, that the area, where an important part of metabolism takes place can be observed. Based on the visualization, a lot has been written about microvascular alternations.

It has been demonstrated that the performance of blood flow is related to many diseases, such as cardiac dysfunction, hypertension, cerebral cavernous malformations or diabetes [19]. There are investigators, who have reported, that the microcirculation is markedly altered in sepsis and that these alternations are more severe in nonsurvivors than in survivors [15,16,17]. Persistent microvascular alterations are associated with development of multiple organ failure and death [18]. The alterations typically include decreased capillary density, there can be substantial heterogeneity in microvascular perfusion between areas separated just by a few millimetres as well. Microcirculation alternations occur by the patients with extracorporeal membrane oxygenation (ECMO) and intra-aortic balloon pulsation (IAMB) or during anaesthesia [8].

Another research field focuses on cardiac surgery and cardiopulmonary bypass (CPB). In [12] the changes of PPV during cardiac surgery are assessed. Moreover, the comparison of cardiac surgeries with and without CPB is present. The outcome of the study is, that the alternations of PPV are observed in cardiac surgery patients whether or not CPB is used. In [13] the quality of microcirculation with and without CPB was compared. They claim the quality decreased during early surgery with CPB. Off-pump resulted in a significantly better microcirculation compared to on-pump for three of six parameters during surgery. However, by the end of surgery there was no difference between the groups. In [14] twelve adult cardiac surgery patients received microvascular assessment before and after cardiopulmonary bypass. After cardiopulmonary bypass, sublingual microcirculation showed a significantly decreased microvascular flow index and increased heterogeneity index.

Next, we can find in literature, that analysis of microcirculation videos enables to detect four types of shock - hypovolemic, distributive, cardiogenic, obstructive [8]. Hypovolemic shock is caused by the loss of blood volume because of gastrointestinal bleeding, extravasation of plasma, major surgery, trauma or severe burns. Obstructive shock is caused by an obstruction of circulation by pulmonary embolism or pericardial tamponade. Distributive shock results in an excessive vasodilation and impaired distribution of blood flow, patients have good capillary refill. Cardiogenic shock is characterized by myocardial dysfunction resulting in adequate cardiac output, capillary refill is poor. [7]

Practically, any kind of disease which is related to the blood perfusion alternation might be observed in microcirculation videos. We might expect, that in the future microcirculation analysis will be the standard method of assessing patient's state of health. Moreover, based on quality of microcirculation, side-effects of such procedures like ECMO or CPB might be revealed. Even though it is a young field, there are many important findings, which show, that microvascular visualization is a promising direction.

1.2 Visualization of microcirculation

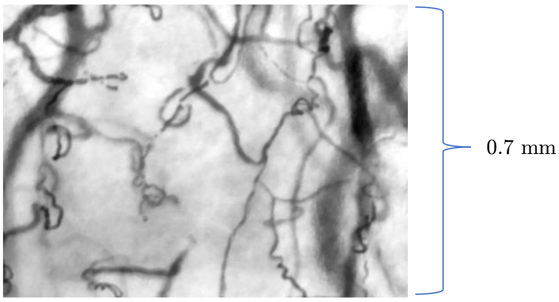


Figure 1: One frame of a video of sublingual microcirculation. Video was captured by Microscan device with Sidestream Dark Field Imaging Technology from MicroVision Medical, NL

Until late 1990s intravital microscopy (IVM) was considered a gold standard for microcirculation visualization. It enables imaging cells through an imaging window that is implanted into the tissue. However, IVM could not have been performed in patients, because there was a need of implantation of an imaging window. Thus, the research concerning microcirculation depended on animals. Another possibility of assessing microcirculation was laser Doppler technique, which enables blood flow velocity measurement. However, it provides just information about average of the velocities in all vessels in a defined tissue volume, which is not enough for complex microvascular analysis. [63]

The boom in the observation of microcirculation started after introduction of non-invasive hand-held vital microscopes (HVM) based on the principle of orthogonally polarized spectral imaging (OPS) in 1999 [8]. Principle of the technique is the illumination of tissue by light source (λ=548 nm) of linearly polarized light. Most of the reflected light retain its polarization and cannot pass through the orthogonal polarizer (analyser) to absorbing material in the foreground, where image is created. Those beams, which penetrate the tissue more deeply and undergo multiple scattering events become depolarized. These beams pass through orthogonal polarizer and back-illuminate absorbing material. The first generation of HVM has been replaced and is not available commercially anymore [8]. However, it was a milestone of microcirculation visualization by that time.

Figure 2: First generation of hand-held vital microscope with the principle of orthogonally polarized spectral imaging, taken from [64].

Second generation of HVM is commercially available and is called sidestream darkfield imaging (SDF). It is based on the principle, that haemoglobin absorbs green light. Green light ($\lambda = 548$ nm) is emitted by peripheral light-emitting diodes toward tissue arranged in a circle at the end of the light guide. The light is absorbed by RBCs, whereas the rest of tissue reflects the light back to the camera. This causes RBCs are displayed in a black color and their surrounding is brighter [3,63].

Recently, a third generation of HVM, called incident darkfield imaging (IDF) was introduced. It is an alternative mode to SDFI with improved optical resolution. It has a computer controlled high-density image sensor synchronized to an illumination unit [65].

1.3 Manual microcirculation analysis

The main disadvantage of manual assessment is subjectivity. There are many rules and instructions, which try to unify the assessment of microcirculation, however, every human treats them differently.

1.3.1 Proportion of perfused vessels (PPV)

PPV describes the overall quality of perfusion in a video. The number of non-perfused vessels is defined. Vessel is considered non-perfused, when RBCs are present, but they do not move. Moreover, it is such vessel, where flow occurs, but at least 50 % of the time there are no RBCs in the vessel. [1]

$$PPV = 100 \cdot \frac{N-NP}{N} \, (\%) \ . \tag{1}$$

where N is the total number of vessels and NP number of non-perfused vessels.

Exact manual assessment of PPV is time-consuming [8]. It is difficult to specify the number of vessels for the operator. Often it is hard to decide, whether the vessels bifurcate or just cross in different depths of the tissue. Besides that, sometimes a single crooked vessel looks like the group of different vessels. Nevertheless, it is possible to roughly estimate the percentage of altered-flow vessels quite quickly.

1.3.2 Microvascular flow index (MFI)

This descriptor evaluates the overall type of flow in the video. The more vessels with the insufficient perfusion the video contains, the smaller MFI is. Video is split into four quadrants, the operator labels each quadrant with a certain number according to the type of flow he assumes is dominant. The flow is characterized as absent (0), intermittent (1), sluggish (2), or normal (3). MFI is computed as the average of the four quadrants values. [1] MFI values ≤ 2.6 are agreed to identify microcirculatory alteration. [8]

The measurement is quick, because the principle is simple. However, MFI is is not relevant for medical assessment in several ways. The score for video labelled as 1,1,2,2 is identical with the score for the video assessed as 0,0,3,3, even though the difference of the flow in the videos is obvious by eye. On the other hand, both such microcirculations are probably altered, which is what MFI recognizes.

1.3.3 Flow heterogeneity index (FHI)

FHI provides an additional information based on the previously named descriptors. It describes the heterogeneity of the flow between different videos of microcirculation. That means it compares three to five videos from one patient between each other. Having MFI or PPV from these videos, it is computed as the difference of their extreme values divided by their mean value. [1]

1.3.4 Perfused vessel density (PVD)

$$PVD = TVD \cdot PPV \tag{2}$$

It describes the amount of perfused vessels in video. TVD (total vessel density) will be defined in 1.4.1

1.4 Automatic microcirculation analysis

1.4.1 Total vessel density (TVD)

TVD is defined as

$$TVD = \frac{Sv}{St} \tag{3}$$

where Sv refers to the total area of vessels and St total captured area of video. [8]

Principle of TVD is the fact the vessels in the microcirculation videos are visible only if they are perfused. When the perfusion in some vessels is decreased so much they do not appear in the video at all, TVD value decreases too.

The utility of TVD is based on the assumption, that the vessel net in normal microcirculation videos is very dense. However, from our experience it is not always true. Moreover, the reliability of TVD is influenced by the size of captured vessels. Occasionally, a wide vessel occupies a significant part of the frame which leads to the increase of TVD.

Interestingly, we found a study [62], where TVD is employed with unit mm, which means only the length of vessel segments is taken into consideration.

1.4.2 DeBacker score

DeBacker score is an effective simplification of TVD during manual analysis. It describes the number of visible vessels without the need of exact summation of the perfused segments. Just as TVD it is based on the assumption, that the vessel net in normal microcirculation videos is very dense.

The principle of the descriptor is as follows: The video is symmetrically split by three vertical and three horizontal lines.

DeBacker score is computed as the number of vessels crossing the lines divided by the total length of the lines in millimetres. [1]

It is possible to extend the descriptor by using only vessel with certain range of diameters for computing the score. Usually, deBacker score for small vessels with the threshold of 20 μm and for the rest of the vessels is computed.

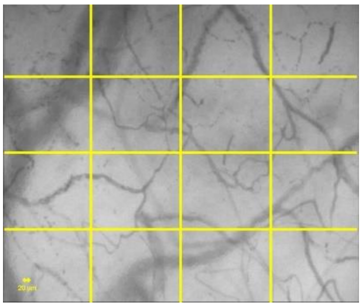


Figure 3: The principle of deBacker score. Taken from [1].

1.5 Constraints of automatic analysis

There are several aspects in microcirculation videos, which hamper the automatic analysis as subsurface scattering, defocus, sensor noise and field of view drift. [11] Scattering of light on the path from the capillaries to the camera reduces contrast of the images. Defocus is present because capillaries are embedded at varied depths while the depth of field of the camera is fixed, so some capillaries appear blurred. The instability of the video makes the video unusable for any analysis quite often. Not only the fact, that the recording device has to be kept in hand of the operator during the recording causes the drift of field of view. Moreover, there are motions induced by heart beat and respiration of the subject. [11] Moreover, pressure artefacts [2] destroy significant part of videos (10 % of videos in our dataset). The artefacts are caused by the camera, when it develops too much pressure on the captured tissue. Blood flow in vessels decreases and the information the video provides becomes unreliable.

1.6 Existing software tools

Several software tools for microcirculation analysis have been developed, such as AVA (Microvision Medical B.V.) [9], tSICA (Microvision Medical B.V.) [4] or CapiScope (KK Technology). [5]

1.6.1 AVA

It is the most common software. Besides others, it determines deBacker score and PPV automatically (we consider PPV parameter of manual analysis, because, the decision, whether a certain vessel is perfused or not has to be made). AVA provides more descriptors, but we do not find them interesting, because they do not belong to the commonly used descriptors. To recognize, if the vessels are perfused or not and to estimate velocity of flow, space-time diagrams (STD) are used in AVA. STDs will be described in section 1.7.3. Architecture of AVA software is displayed in figure 4.

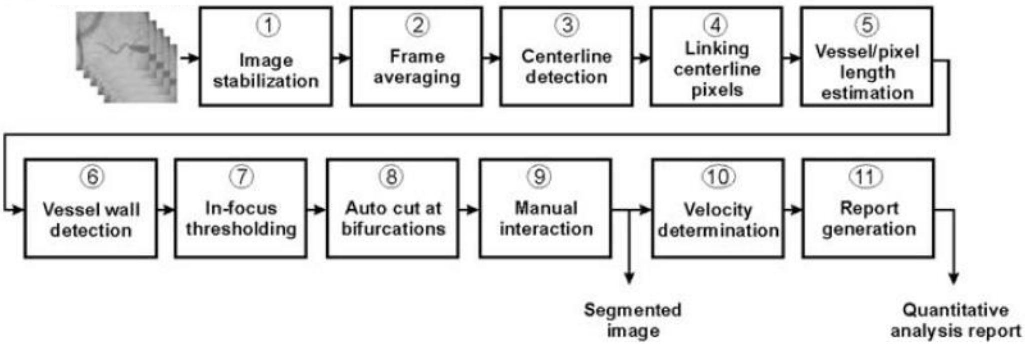


Figure 4: Architecture of AVA software, the commercial microcirculation analysis tool, taken from [9]

1.6.2 tSICA

Software tSICA is a faster version of AVA. The velocity measurement is based on the temporal pixel intensity fluctuations. It is just a quick method, so as a result, the software only decides, whether the vessel is perfused or not. Time-averaged frame from the video is created. Next, temporal pixel intensity fluctuation picture is generated. It is quantified as the standard deviation of the intensities in time. [4]

1.6.3 CapiScope

According to [10], CapiScope computes the correlation between the centerlines of two sequential frames for various shifts of the centerlines. The shift with the biggest correlation is marked as the distance RBCs travelled between the frames.

There are other approaches for complex microcirculation videos analysis described in [11] and [19]. In [11] 3D convolution with pre-defined spatial-temporal filters is implemented to assess the speed of RBCs. In [19] particle image velocimetry and optical flow are used.

1.7 State-of-the-art of particular software components

1.7.1 Correction of video instability

Stabilization by estimating and smoothing a motion model is widely used. Such an approach can be divided into two categories: 2D methods and 3D methods, depending on the model they use. Theoretically, 3D methods are better, but they are fragile and slow. In 3D methods the full 3D structure of the scene is recovered and motion is modelled directly using the recovered position of the camera. The algorithm, which is used to recover 3D structure from a video is sensitive to noise. [54] 2D methods are faster than 3D methods. They are based on the assumption that the motion between two frames can be represented by a single 2D transformation matrix. It models translation, rotation and scaling. However, they only work well on single plain scenes or scenes, which are far away from the camera. In [55] four parameter transformation and in [56] six parameter transformation is used.

Besides motion model methods, there are trajectory based methods. Video is stabilized by smoothing the extracted trajectories without setting up a motion model. The trajectories are extracted by feature tracking [54]. Extracting feature trajectories is tricky, because not all movements present in the video are caused by the camera. Sometimes the feature is not static, so it undergoes displacements that are caused by both the camera motion and the movements of the object in the scene. These moving objects need to be separated. [57]

In [20] a block matching algorithm is used while stabilizing microcirculation videos. The similarity of the blocks in consecutive frames is measured by cross-correlation coefficients. Block matching algorithms use a predefined size of blocks or even entire images to estimate motion vectors. However, if a window does not contain distinctive details, there is a high probability of mismatch. Typically, this happens when just background pixels are included. To avoid such error the processed blocks are checked to ensure they include vessels using Laplacian of Gaussian filtering. Gradient of the Gaussian improves visibility of blood vessels. The maximum values from several areas of the frame are chosen to be the distinctive features. Typically, branching points meet the condition. The blocks are implemented as follows: 25×25 pixel window around each control point is selected as subregion. The cross-correlation is calculated between these sub-regions in the current frame and a 40×40 surrounding. The dimensions of the windows are state empirically.

In [11] the stabilization is based on the patches, that are matched between frames using template matching. Just like in [20], the patches with enough texture are chosen according to the variance of pixel intensities.

1.7.2 Image segmentation

A wide range of image segmentation methods is described in literature. Their suitability always depends on the problem to be solved, there is no universal segmentation technique that works equally well in all situations. Generally, image segmentation methods can be categorized into two classes, i.e. semi-automatic and automatic.

Semi-automatic techniques require some sort of human input and usually provide better performance. In automatic techniques no operator intervention is needed [25]. Mostly, while segmenting an image, such properties like color, texture, gradient or spatial attributes are assessed. Namely, region growing, region splitting, region merging, parametric/geometric active contours, histogram thresholding, feature based clustering, graph-cuts or watersheds. [28]

Concerning vessel detection, automatic methods comprise of vessel tracking, matched filter responses, grouping of edge pixels, model based locally adaptive thresholding, topology adaptive snakes and morphology-based techniques. In contrast, supervised methods require manually labeled images for training, features and classifier. From the training sets feature vectors are constructed and they are labeled as vessel or non-vessel. It is assumed, that feature vectors from a certain class cluster together in the feature space. In that case a classifier that determines a decision boundary between the different classes can be designed. After the training, a non-labeled feature vector can be classified by determining on which side of the decision boundary it is situated. such classifiers like k-nn, SVM or random forest can be employed [34]. Concerning semi-automated image segmentation, wide range of software tools is available. Usually, besides other functions, they determine sufficient features and train a classifier automatically based on the training data. Such tools are for example Dragonfly (Object Research Systems (ORS) Inc., Montreal, Canada), TurtleSeg (Oxipita Inc., Vancouver, Canada), Fiji [42] or Ilastik [24].

To facilitate vessel segmentation, Frangi filter [27] is widely used. It has been employed in many research projects concerning vessel detection [29,30,31,32,33]. It enhances contrast of tubular structures in image. The filtering process searches for geometrical structures which can be regarded as tubular. The probability that the pixel belongs to the tubular structure is called tubularity. It is provided by quantification of following parameters: deviation from a blob-like structures, the difference between plate-like and line-like structures, and background noise, respectively [33]. Information about contrast and direction is extracted from Hessian matrix of each pixel. Hessian matrix contains second order derivatives of the smoothed image. Matrix H for pixel x,y in smoothed image I is constructed [29]:

$$H(x,y) = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \tag{4}$$

Gonzales [35] extracts features from Hessian matrix to train a tubular structure classifier. He claims, that the second-order derivatives used to compute the Hessian matrix do not provide a local description that is powerful enough to account for irregular tubular structures and perform in noisy images. Computing of higher-order derivatives is recommended.

In both AVA software [53] and in Liu's microcirculation analysis approach [11] automatic vessel detection is based on a automatic detector of curvilinear structures described by Steger [36]. Hessian matrix is employed to detect vessel centerlines. For each pixel, analysis of the matrix results in a vector that points in the vessel direction (***t***) and a vector in the perpendicular direction (***n***). The pixel is considered to be the candidate for the centerline pixel if the second order spatial derivative in direction of ***n***, is markedly higher than in ***t*** direction, Candidate pixels are identified as centerline pixels if the intensity profile in ***n*** direction, is locally at its extremum. In [11] (v thesis) just the centerlines are detected, finding the complete vessel segments is not the purpose of the approach. In [53], besides centerlines vessel walls are detected. To do so, cross-sectional intensity profile of each centerline pixel is analysed. The wall is marked by the points where the cross-sectional intensity profile in each centerline pixel shows its maximum steepness in the direction of vector ***n.*** The cross-sectional intensity profile is obtained by sampling the image at sub-pixel level in the normal direction.

Besides [36] there are more methods for centerline detection. They can be divided in two categories. In the first category vessel segmentation is needed. Thinning-based methods [39], which perform skeletonization of segmentation, and active contour-based methods [40] are included. In both cases segmentation has to be created before finding the centerline. These techniques are efficient only when good segmentation is provided [41]. Thinning-based methods often produce disconnected components and artefacts on noisy data, which then require post-processing and analysis to merge into a meaningful tree. In the second category, just tubular images (like in [36]) are provided. Tubularity is determined and evaluated. It can be either computed or classified by machine learning [41]. There are tracking methods, where set of seed points is initiated and high tubularity paths are recursively traced. It means tubularity is determined locally [38]. Or tubularity can be computed globally in whole image and a global objective function can be optimized to create the most convenient centerline tree [37]. Or, like in [41], the samples of centerline points are determined and connected by maximum probability paths.

1.7.3 Velocity measurement

Many motion estimation techniques have been employed to determine RBCs velocity from sequence of frames, including space-time images [9], cross correlation methods [66,19], optical flow [19], temporal pixel intensity fluctuation [4], 3D convolution with predefined spatio-temporal filters [11].

Cross correlation methods search particular structure from consecutive frames by cross-correlation. We can find implementations of both 1D and 2D methods. 1D method was employed in CapiScope software [66]. Centerline of a certain vessel segment was compared in two images. In fact, centerline is a vector of pixel intensities. Vectors can be shifted mutually and the shift with the highest correlation is supposed to be the inter-frame motion. Practically, this method is limited because the centerline structure is not constant in time. Moreover, there are curved structures which influent flow unpredictably. 2D cross-correlation method is called PIV (particle image velocimetry). Correlation is employed to find a window from first image in the subsequent frame [19]. In fact, it is the same method, which is used in AVA software to stabilize videos - windows with distinctive texture are selected and cross-correlation is used to find them in subsequent images.

Optical flow is another possibility of velocity determination. This approach estimates motion vector for each pixel of region of interest. There are several techniques to do so. In each case a supposition has to be determined. In [19] comparison of both optical flow and PIV method is provided. The experiment on artificial microcirculation video was performed. The author claims, that optical flow performs better, but combination of both techniques works the best. Optical flow determination was based on Horn-Schunck algorithm, where smoothness in the flow over the whole image is presumed.

3D convolution with predefined spatio-temporal filters is based on the assumption, that certain flow patterns repeat in microcirculation videos. In that case artificial filters, which correspond with certain flow patterns can be generated. Parameters of such filters are direction and velocity of shift. Velocity is determined according to the energy response of the filters. [11]

Temporal pixel intensity fluctuation does not describe velocity directly, but it relates with it. It is based on observation of brightness changes of particular pixels in time. Naturally, in vessels with no flow brightness of pixels does not change in time, whereas when movement occurs, so do pixel intensity fluctuations.

In STD (figure 5), each column is created by the centerline of the vessel from a certain frame in time order of the video. The elements, which occur in the centerline, change their position within the vessel because of the blood flow. This leads to the creation of the continuous lines in STD. The orientation of the line shows the direction and the velocity of the flow. When the lines are detected and their orientation is determined, quantitative analysis of the flow can be provided.

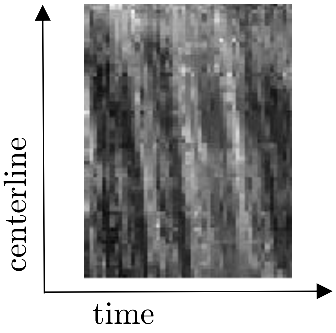


Figure 5: Space-time diagram (STD) is a common technique of RBCs velocity determination. Each column in STD is created by centerline of vessel from a certain frame in time order of the video. The orientation of lines corresponds with the velocity of centerline elements.

2 Main research goals

The goals of the thesis are:

1) to design a descriptor of blood flow velocity in microcirculation videos,

2) to develop a software for microcirculation analysis, which will determine the novel descriptor and traditional microcirculation descriptors,

3) to process microcirculation videos by the software,

4) to compare the results of analysis with results from commercial software

5) to create a classifier of pathological and physiological microcirculation.

We have no ground truth of RBCs velocity, so we decided to create artificial videos, where flow velocity is known.

We have videos with altered circulation and with normal circulation at our disposal. By altered microcirculation we mean the videos, which were captured, while the patient was on CPB during cardiac surgery. Based on available literature, we expect, that on-pump microcirculation is altered.

We will compare our results with AVA. It provides deBacker score and PPV. We will implement deBacker score in our software, because it does not require any qualitative assessment (unlike PPV).

3 Description of data

While building the software, we could have used 1087 videos, which were recorded by Microscan device with Sidestream Dark Field Imaging Technology from MicroVision Medical, NL. The dataset was created in order to verify the assumption, that a patient with non-pulsatile flow on the cardiopulmonary bypass would represent the stage with an altered microcirculation. The videos were captured on 6 subjects at the cardiac surgery department of Na Homolce Hospital, Prague, Czech Republic. We received the videos from MD, MSc David Macku, the leader of the experiment.

In total, there were 563 videos of on-pump subjects, 524 videos of off-pump subjects. Each video was roughly 1 second long.

There were 4 different types of off-pump videos, marked M1, M2, M4, M5:

M1: Captured at the standard ward before surgery.
M2: Captured after the introduction into total anesthesia.
M4: Captured at the end of CPB, during the suture of sternotomy.
M5: Captured at the standard ward before the patient was discharged.

There were 4 different types of on-pump videos, marked M0, M45, M90, M120:

M0: Captured during the surgery procedure after the onset of cardiopulmonary bypass.
M45: Captured 45 minutes after initiation of CPB.
M90: Captured 90 minutes after initiation of CPB.
M120: Captured 90 minutes after initiation of CPB.

3.1.1 Dataset for analysis

The initial dataset of 1087 videos contained a lot of low-quality records. Common defects were: poor focus, no visible vessels, bubbles, scattered red blood cells all over the field of view (probably caused by bleeding in the captured area), big camera motion (captured area changed entirely within the video). So, we sorted out the videos manually, we selected 260 good-quality videos - 120 off-pump videos and 140 on-pump videos manually.

We processed the selected videos by our stabilization algorithm and checked the stabilized videos manually. We rejected 93 videos (36 %), which were not stabilized sufficiently. From the group of stable videos we eliminated those, where pressure artefact seemed to occur, which was in 16 cases (10 %). Finally, dataset of stable videos, which were sufficient for analysis, consisted of 151 videos-75 on-pump, 76 off-pump.

The overview of the distribution of video types, which were present in the final dataset, is depicted in table 1.

Table 1: Distribution of videos in the dataset for analysis. In off-pump row, M1-5 labels when the video was captured. M1: at standard ward before surgery, M2: after introduction into total anesthesia, M4: at the end of CPB, M5: at the standard ward after surgery. In on-pump row M0-120 stand for: M0: at the beginning of surgery after onset of CPB, M45: 45 minutes after onset of CPB, M90: 90 minutes after onset of CPB, M120: 120 minutes after onset of CPB.

Off-pump:	M1	M2	M4	M5	Total:
	12	33	19	11	**75**
On-pump:	M0	M45	M90	M120	Total:
	40	14	16	6	**76**

4 Methods

4.1 Software

Our approach is inspired by AVA [9], but it differs in several aspects. Simultaneously, Liu's approach [11] is a guideline from time to time.

In contrast with AVA, vessel segmentation in our case is based on machine learning instead of vessel wall detection and thresholding. Furthermore, we have implemented optical flow method for flow velocity estimation and temporal pixel intensity fluctuation-based descriptor of microcirculation.

The structure of our software is displayed in figure 6.

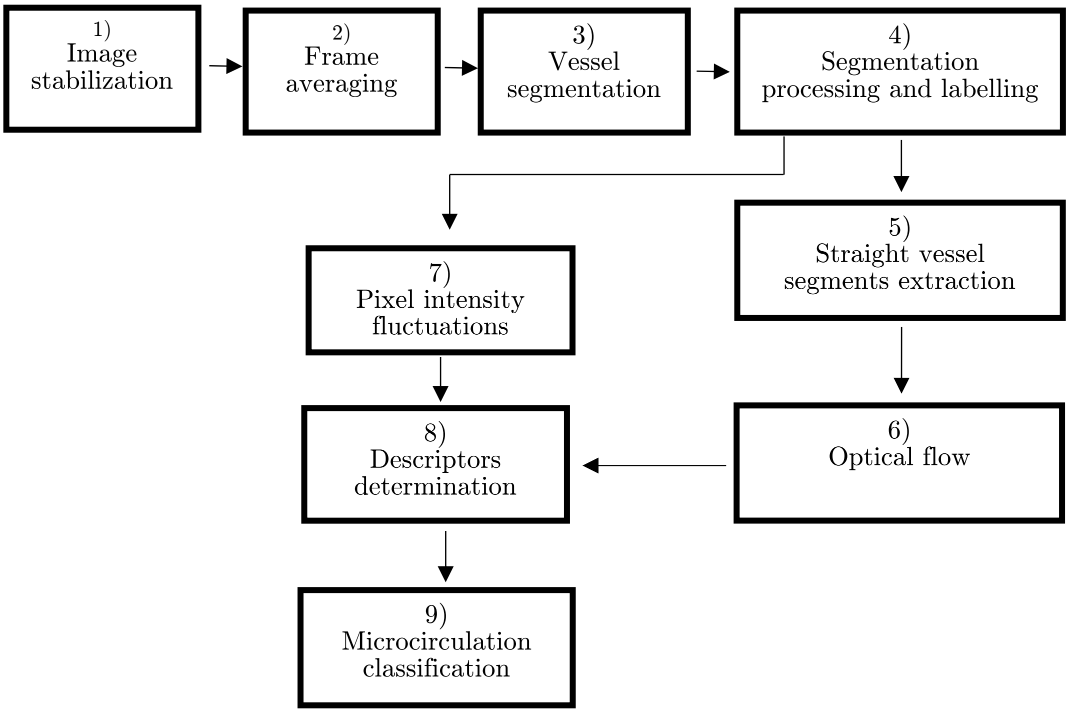
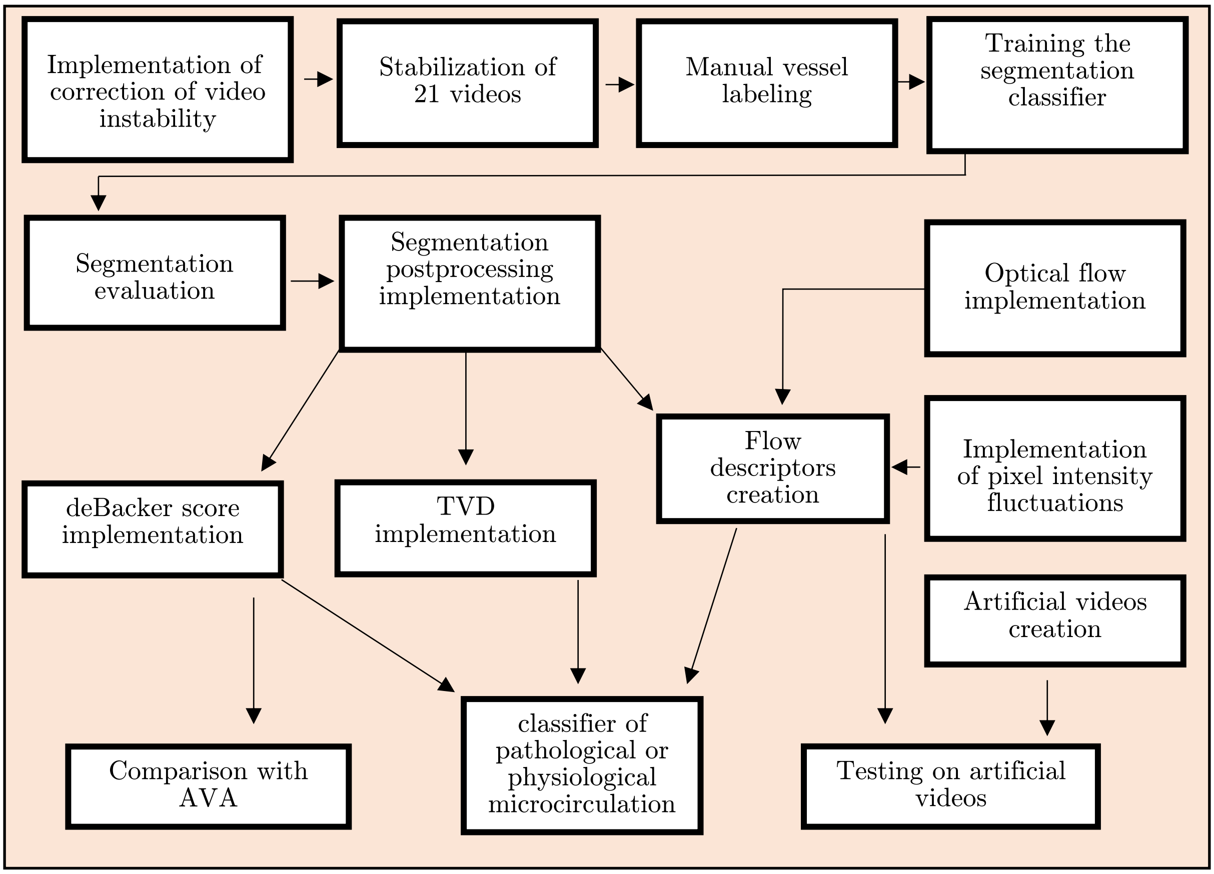

1) Image stabilization
2) Frame averaging
3) Vessel segmentation
4) Segmentation processing and labelling
5) Straight vessel segments extraction
6) Optical flow
7) Pixel intensity fluctuations
8) Descriptors determination
9) Microcirculation classification

Figure 6: Software architecture

4.2 Project workflow

In figure 7 we display the working diagram which includes the steps we processed to achieve all the goals of the thesis.



Implementation of correction of video instability
Stabilization of 21 videos
Manual vessel labeling
Training the segmentation classifier
Segmentation evaluation
Segmentation postprocessing implementation
Optical flow implementation
Flow descriptors creation
Implementation of pixel intensity fluctuations
deBacker score implementation
TVD implementation
Artificial videos creation
Comparison with AVA
classifier of pathological or physiological microcirculation
Testing on artificial videos

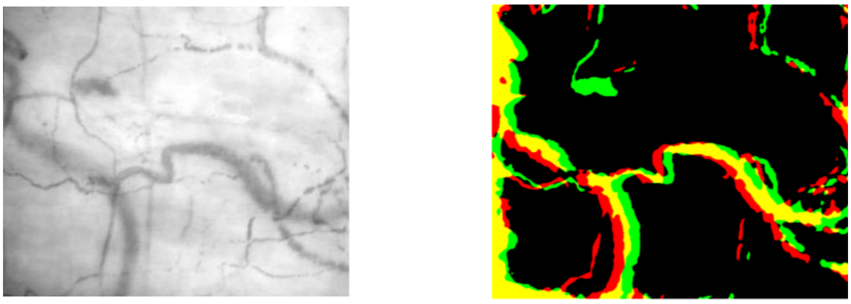
Figure 7: Working diagram. The steps processed to achieve all the goals of the thesis are displayed.

5 Correction of video instability

Our approach is based on 2D motion model estimation. We took the implementation of motion estimation from [50].

5.1 Image pre-processing

Since a natural movement of the RBCs is present in the video, it is challenging to recognize just the movement caused by camera while stabilizing. Moreover, quite often a motion between tissue layers in different depths appear. To prevent this, we process following steps: First, we enhance the contrast of the image by adapting the histogram of pixel intensities. Next, we blur the frames to conceal the movement of blood cells. Blurring is processed by a 2-D Gaussian smoothing kernel with the window size of 3 pixels. To get rid of the motion in different depths, we apply thresholding, because deeper structures appear brighter. Threshold is computed from each image using Otsu's method [26]. It assumes, that two classes of pixels are present in an image. It calculates the optimum threshold separating the two classes so that their combined intra-class variance is minimal. Like this we remove the bright structures and keep just the important dark structures.



(a)
Original frame, its segmentation is displayed in green color in (b).

(b)
Segmentations of two consecutive frames, first one in green, second one in red color. Yellow color marks the area where both segmentations overlap.

Figure 8: Segmentations of dark areas in unstabilized frames. The distinct structures from (a) are segmented as green color in (b). The segmentations of two consecutive frames (b) is used for motion model estimation.

5.2 Motion estimation

The estimation of the motion model is based on spatial and temporal derivatives of binary frames. Once the model is estimated, the default frames of the video can be stabilized.

Motion between two images $I(x_{t-1},y_{t-1})$ and $I(x_t,y_t)$ is modelled with a 6 parameter affine transform [50]:

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \cdot \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} + \begin{pmatrix} m_5 \\ m_6 \end{pmatrix} \quad (5)$$

When we assume that picture $I(x_{t-1},y_{t-1})$ and $I(x_t,y_t)$ are the same, but $I(x_t,y_t)$ is deformed, we can write the equation (1) as:

$$I(x_t, y_t) = I(m_1 x_{t-1} + m_2 y_{t-1} + m_5, m_3 x_{t-1} + m_4 y_{t-1} + m_6), \quad (6)$$

where parameters m_1, m_2, m_3, m_4 control scaling and rotation and m_5, m_6 control translation. Despite the approach in [50], we found out it was more convenient to align all the frames to a certain frame, which is in the middle of the sequence, so all the frames are compared with just one picture. We determine $I(x_{tm},y_{tm})$ to be the middle frame of the video sequence.

In order to estimate the parameters of the model, following quadratic error function has to be minimized [50]:

$$E(m) = \sum_{x,y \in \Omega} [I(x_t, y_t) - I(m_1 x_{tm} + m_2 y_{tm} + m_5, m_3 x_{tm} + m_4 y_{tm} + m_1)]^2 \quad (7)$$

Where Ω is the region of interest of the frame.
The equation (7) is non-linear in its unknowns. To compute the inter-frame motion, differential motion estimation is used. We approximate equation (7) by the first-order truncated Taylor series expansion:

$$E(m) \approx \sum_{x,y\in\Omega} [I - (I + (m_1x + m_2y + m_5 - x)I_x + (m_3x + m_4y + m_6 - y)I_y - I_t)]^2$$

$$= \sum_{x,y\in\Omega} [I_t - (m_1x + m_2y + m_5 - x)I_x - (m_3x + m_4y + m_6 - y)I_y]^2 \tag{8}$$

where I_x, I_y and I_t are partial derivatives of the image I with respect to position x, y and time t.

For notational convenience, the parameters $m_{1\text{-}6}$ are dropped and k and $\boldsymbol{c}$ are given as:

$$k = I_t + xI_x + yI_y \tag{9}$$

$$\boldsymbol{c} = (xI_x \quad yI_x \quad xI_y \quad yI_y \quad I_x \quad I_y) \tag{10}$$

Finally, the approximation of the error function is:

$$E(\boldsymbol{m}) \approx \sum_{\Omega} [k - \boldsymbol{c}^T\boldsymbol{m}]^2 \tag{11}$$

The quadratic error function is now linear in its unknowns and can be minimized analytically by differentiating with respect to m. Setting the result equal to zero and solving for m:

$$m = \left[\sum_{\Omega} \boldsymbol{c}\boldsymbol{c}^T\right]^{-1} \left[\sum_{\Omega} \boldsymbol{c}k\right] \tag{12}$$

Since we want to control just rotation and translation, but not scale, we extract the rotation angle θ from the affine parameters [21]:

$$\theta = arctg\left(\frac{m_3}{m_4}\right) \tag{13}$$

We can now deform the image $I(x_t y_t)$ and create $I(x_t' y_t')$, which is aligned with the reference frame $I(x_{tm} y_{tm})$. First, we rotate and translate the coordinates [22]:

$$\begin{pmatrix} x_t' \\ y_t' \end{pmatrix} = \begin{pmatrix} \cos\theta & -sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} x_t \\ y_t \end{pmatrix} + \begin{pmatrix} m_5 \\ m_6 \end{pmatrix} \tag{14}$$

By x_t' and y_t' the query points for an interpolation are defined. Image $I(x_t' y_t')$ is obtained by the cubic interpolation of $I(x_t y_t)$.

5.2.1 Coarse to fine method

We use the implementation described in [50] of coarse to fine method to estimate large motion, typically more than 10 pixels. The motion is estimated in different resolutions like displayed in figure 9. The motion estimated at coarser level is used to warp the frame at the next finer level, until the finest level of the pyramid is reached. Like this, large motion estimation from the coarse level is iteratively refined. When warping the image in a finer level, the estimation of translation vector (m_5, m_6) from the coarser level has to be multiplied by 2. The way the affine estimation parameters are accumulated is described in equations (16 - 18).

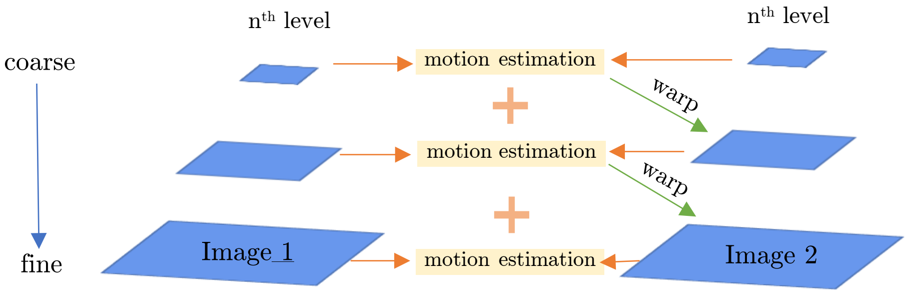


Figure 9: Multiresolution method of motion estimation. Images are down sampled *n* times. Motion is estimated on small images, so bigger velocity can be recognized. In finer level the estimation is enhanced. Small images are up sampled, estimated translation vector is multiplied by 2 and one of the images is warped. Motion is estimated again. Like this, iteratively, the finest level estimation is achieved.

In our case image segmentations are used to estimate the motion, so the actual video frames are not warped during the process. We need to find the finest motion estimation and apply it on the video frame. We accumulate the model parameters during the multiresolution process like described in [50]. The transformation of the original image coordinates by the parameters estimated in level *n* is given by:

$$\begin{pmatrix} x_t{}' \\ y_t{}' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} x_t \\ y_t \end{pmatrix} + \begin{pmatrix} 2^{n-1}m_5 \\ 2^{n-1}m_6 \end{pmatrix} \tag{15}$$

Like this the coordinates are transformed repeatedly during multiresolution. When we declare

$$A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad \textit{and} \quad \boldsymbol{t} = \begin{pmatrix} m_5 \\ m_6 \end{pmatrix}, \tag{16}$$

repeated coordinates transformation by A and $\boldsymbol{t}$ obtained in different levels n and n-1 can be written as:

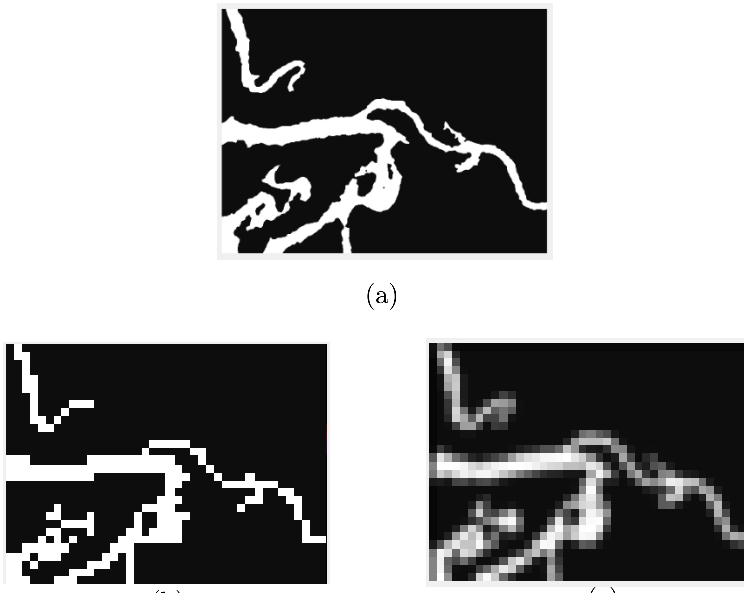
$$\begin{pmatrix} x_t' \\ y_t' \end{pmatrix} = A_{n-1}A_n \begin{pmatrix} x_t \\ y_t \end{pmatrix} + A_{n-1}\boldsymbol{t}_n + \boldsymbol{t}_{n-1} \tag{17}$$

In the manner of equation (17) the model parameters from different levels are accumulated. The combination of coarser and finer level estimation is given by:

$$A = A_{n-1}A_n \qquad \boldsymbol{t} = A_{n-1}\boldsymbol{t}_n + \boldsymbol{t}_{n-1} \tag{18}$$

While setting up the maximum level of the pyramid, resolution of the original image should be considered. In equation (12) it is assumed that the first term, a 6 × 6 matrix, is invertible. To be so, integrating over sufficient amount of pixels is needed. In our case the resolution can be reduced six times at most. Original resolution of the videos is 960×1280 pixels. Minimal resolution, which is necessary to solve equation (11) is $(960/2^6)\times(1280/2^6)$, which is 15×20 pixels.

We filter the images before down sampling from level *n* to level *n*-1 by binomial filter (1/4, 1/2, 1/4) twice along each spatial dimension, as proposed in [23]. In figure 4 down sampled image with and without filtering is displayed. In the filtered image more details are preserved. By filtering incorrect edges are created, so we reduce roi by 3 pixels from each side. Moreover, when frames are transformed, edges without any content remain on their sides. To prevent the edges from disturbing the model estimation in finer levels, roi is determined not to cover the edges of the frames, which were warped according to the estimation from coarser levels.



(a)

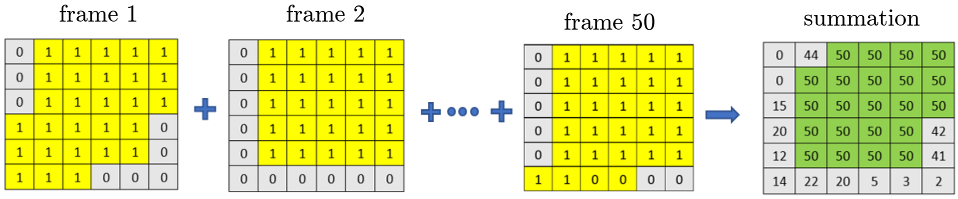
(b)

(c)

Figure 10: The effect of filtering during the reduction of resolution, (a) original resolution, (b) 5× reduced resolution with no filtering, (c) 5× reduced resolution, filtered twice in each level.

5.3 Region of interest in stabilized videos

After the frames are deformed, edges with no content are created. The edges have to be considered when the video content is analysed. So the mask of interest is created. By the mask of interest we mean such a mask, where the area of edges by all the video frames is labeled by zero, the rest is labeled by one. The mask of interest is created as follows: Individual mask of interest is created after each frame of the sequence is warped. The individual masks are summed up when all the frames are deformed. In the summation mask, the value, which equals the number of frames in the video, labels the pixel, where no edges occurred within the video. Any other value corresponds to edges.



0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	0
1	1	1	1	1	0
1	1	1	0	0	0

0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	0	0	0	0	0

0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
1	1	0	0	0	0

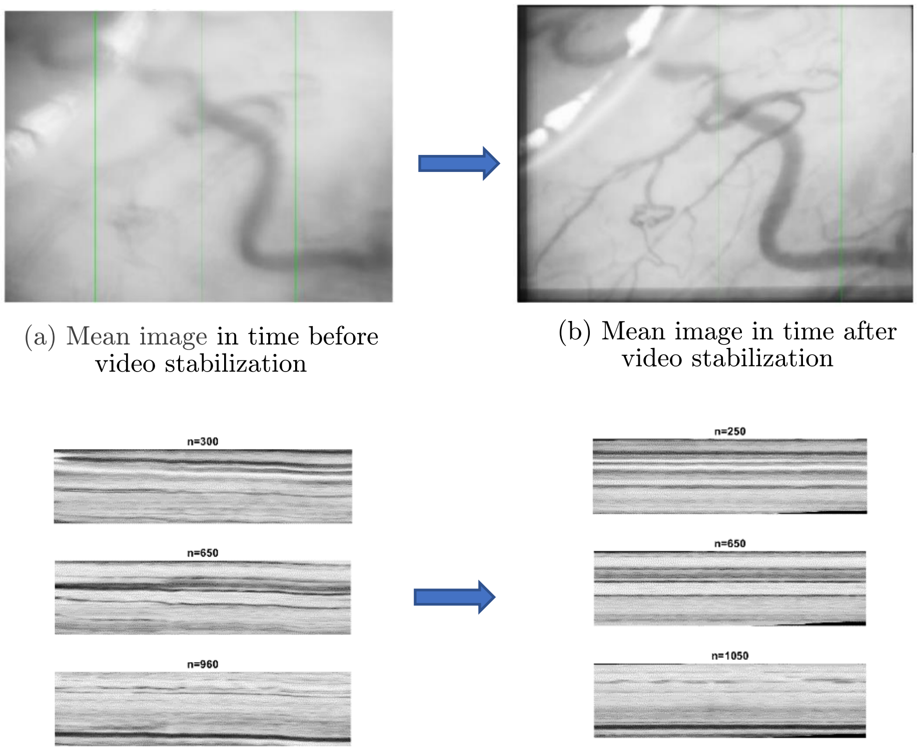
0	44	50	50	50	50
0	50	50	50	50	50
15	50	50	50	50	50
20	50	50	50	50	42
12	50	50	50	50	41
14	22	20	5	3	2

Figure 11: Scheme of how the mask of interest is created. The area of edges is labeled by 0 in each warped frame, the rest is labeled by 1 (yellow). All the consecutive masks are summed up. In the summation, the area, which equals the number of the frames (green), is chosen to be the region of interest.

5.4 Efficacy of implementation

We stabilized 260 videos and checked the effect of stabilization manually. We selected those videos, where, visually, stabilization was not efficient enough. From 260 videos 167 were well stabilized, which is 64.2 %. 93 videos were not stable enough, which is 35.8 %.

The effect of stabilization is depicted in figure 2. Mean image in time is displayed before and after stabilization. When the stabilization is successful, the mean image turns from blurred into sharp (fig. 12a,c). Moreover, the space-time image created by stacking nthcolumn of pixels from each frame of the video next to each other is attached. The selected n columns are marked by the green line in the mean image. After stabilization the crooked lines become straight (fig. 12b,d).



(a) Mean image in time before video stabilization

(b) Mean image in time after video stabilization

(c) Space-time image before video stabilization

(d) Space-time image after video stabilization

Figure 12: Demonstration of successful stabilization. Temporal average image turns from blurred into sharp after stabilization (a,b), space-time image turns from crooked to straight (c,d).

5.5 Summary

We have implemented an algorithm for automatic correction of video instability and checked its function. The stabilization is successful in 64 % of cases. Generally, there are two reasons of unsatisfactory stabilization:

1) Motion of camera is too big, typically more than 50 pixels between subsequent frames. This is the reason in 21 from 93 cases (22.6 %) of unsatisfactory stabilization. In such videos, frames are usually blurry due to the motion, so they do not provide good quality information anyway. Moreover, big motion causes that small roi is provided, as depicted in figure 13. From these reasons we do not suppose it is important to deal with this problem on the software basis at present. It is rather important to maintain stability during capturing process.

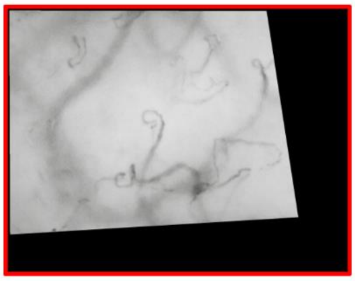


Figure 13: Region of interest in a stabilized video, where big motion of camera occurs. Big edges are created on the sides due to the transformation of frames and much information is lost.

2) There is motion between different layers of the captured area, which occurs in 72 from 93 cases (77.4 %). From our experience, this is a frequent problem. The ideal stabilizing solution should recognize those structures, which are deeper and in addition it should recognize, whether those structures move in relation with the structures in shallow layer. The moving structures should be removed and the others should be kept.

6 Automatic vessel detection

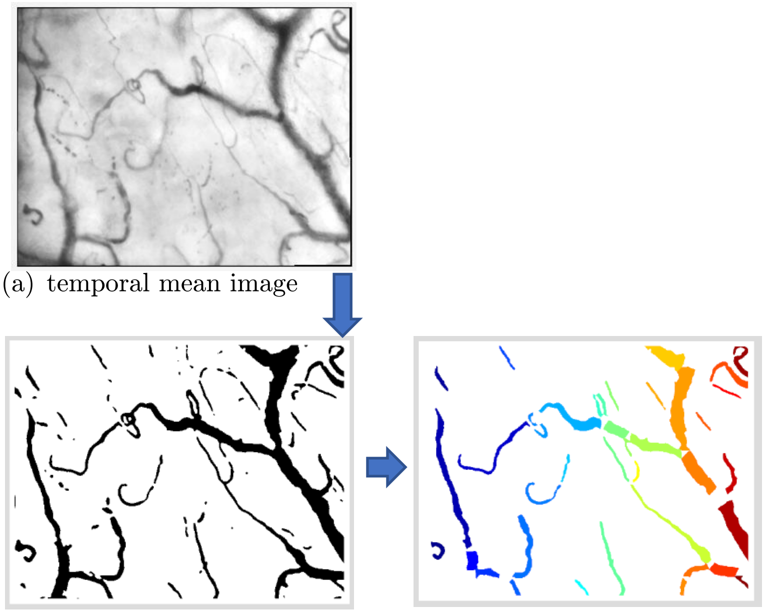
6.1 Introduction

Vessel detection is necessary in order to measure red blood cells velocity inside vessels. The aim is to implement an algorithm, which creates an individual vessel segment mask (fig. 14c) for each microcirculation video. In individual vessel segment mask all the pixels of a certain segment are lebeled by the same non-zero number. The number is specific for each segment. Number zero labels the area with no vessels. In fig. 14c different label values of individual segments are visualized by different colors. When such a mask is created, blood cell velocity in any labeled segment across the video can be calculated.

There re two main stages of vessel detection in our software. Firstly, binary vessel segmentation is created (fig. 14b). Secondly, individual vessel segment mask is created from binary segmentation.

Based on the search, we decided to employ a semi-automatic technique to obtain binary segmentation, because it usually performs better than automatic methods. We created training data and employed an automated segmentation tool to train the classifier.

A thinning-based method is used to label vessel segments.



(a) temporal mean image

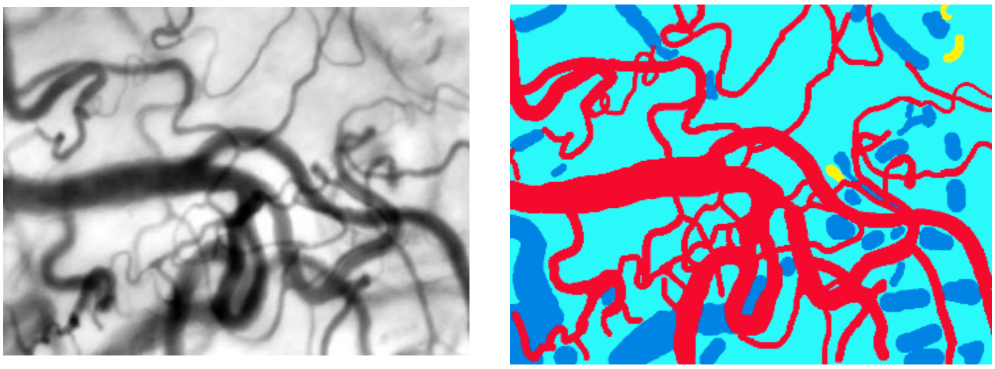
(b) binary vessel segmentation (c) individual vessel segment mask

Figure 14: Diagram of vessel detection. Based on the temporal mean image(a), binary vessel segmentation (b) is created. Binary segmentation is further processed to create individual vessel segment mask (c).

6.2 Binary segmentation

Training dataset was created manually. We worked with temporal mean images of microcirculation videos, like proposed in approach of AVA [53]. Mean images are convenient, because they have better contrast than individual frames. Moreover, in some vessels blood flow is intermittent, so they are not visible on all the frames. By averaging these segments become apparent. We segmented 21 mean images manually. Based on the video content we decide to employ 4 classes of segmentation:

1) Clearly visible vessels. This class labels the areas, which we want to analyse for sure
2) Blurred vessels in deep layers, where blood flow is not recognizable. We determine these structures by a specific label, because in numerous videos they occupy large areas, but they do not provide any information. We suppose they are recognizable, because they are brighter then clearly visible vessels, but darker than background.
3) Background. We attempt to label the areas, which do not contain any structures but noise.
4) Isolated distinct dark structures with different than tubular shape. We use this class, because sometimes there are structures, where we are not sure, whether they are vessels or not. They are as dark as well visible vessels, but we do not want them to be included in clearly visible vessels class, because they do not have tubular shape. It is the least common category.



(a) Input image - average image of video in time

(b) Manual segmentation

🟥 1) clearly visible vessels
🟦 2) blurred vessels in deep layers
🟦 3) background
🟨 4) isolated distinct dark structures

Figure 15: The example of manual segmentation

To create manual segmentations, we used Gimp image editor [44]. From Gimp we exported the completed segmentations in png format. No interpolation and transformation is applied in png format, so no other than four label values are exported to the image.

To segment the vessels Ilastik software [24] is employed. It is a free tool, which enables machine-learning based pixel classification. Once the classifier is trained, it can be used in „headless“ mode, so Ilastik GUI is not needed anymore. So it is possible to implement the classifier into an automatic detecting algorithm. Ilastik workflow offers a choice of generic pixel features, such as smoothed pixel intensity, edge filters and texture descriptors. Once the features are selected, a random forest classifier is trained on training data, which, in our case, were manual vessel segmentations.

6.2.1 Features selection

Pixel features are used to discriminate between the different classes of pixels. Three classes of features are provided in Ilastik:

Color/Intensity: Should be selected if the color or brightness can be used to discern objects.

Edge: should be selected if brightness or color gradients can be used to discern objects.

Texture: Should be selected if the objects in the image have a special textural appearance

All the features can be selected on different scales. The scales correspond to the sigma of the Gaussian which is used to smooth the image before application of the filter. Filters with larger sigmas can thus pull in information from larger neighbourhoods, but average out the fine details. [24]

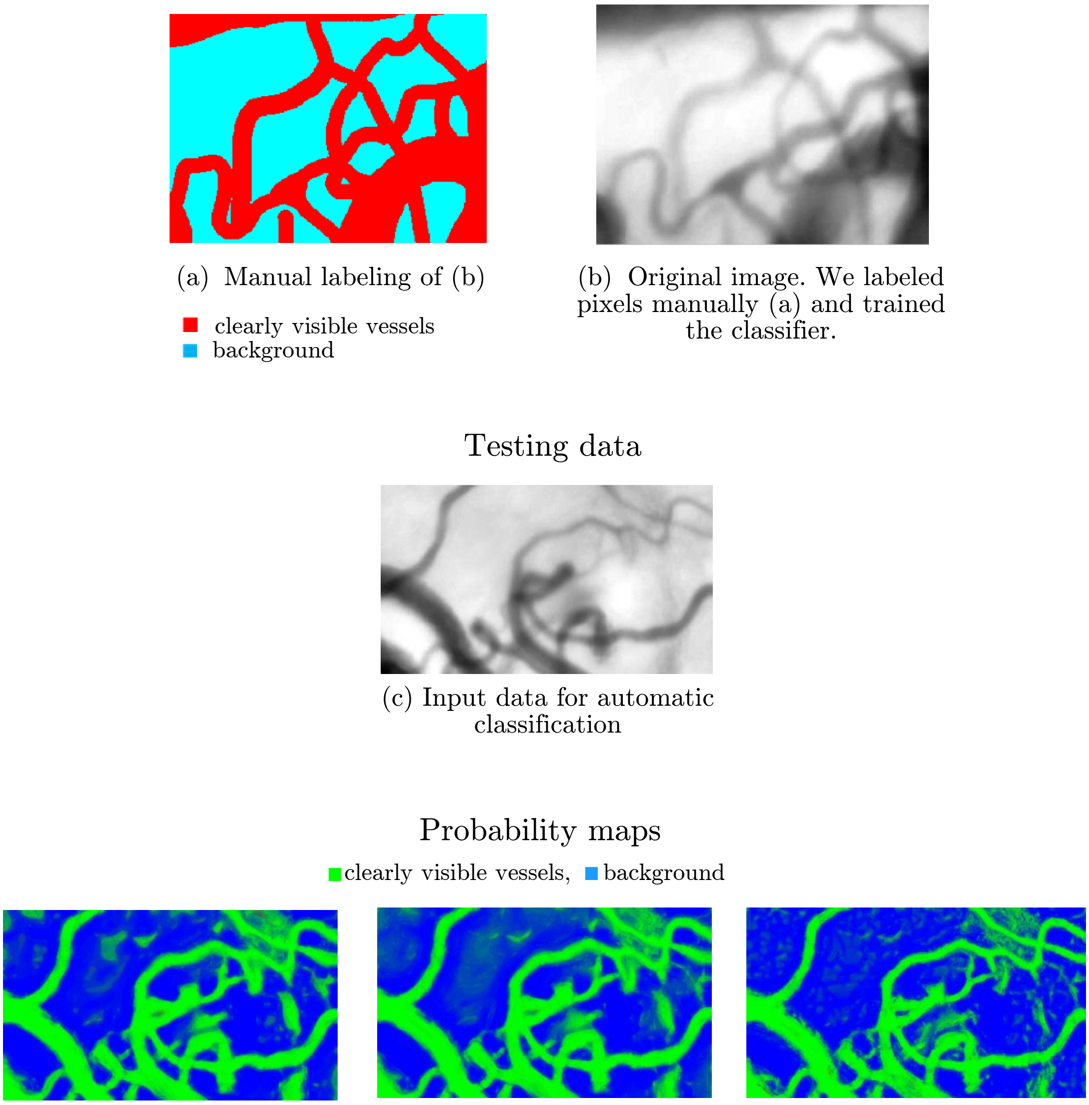
It is recommended to choose as many features as possible in wide range of scales in the beginning of the training process, because one barely knows, which features are significant in a particular experiment. However, with more features computational time is bigger. So, we made several experimental segmentations with different combinations of features to see how particular features influent the results. In the first stage, the experiment was performed on small images, so the classifier was trained quickly (fig. 16). Based on the experiment we decided to choose color/intensity features and texture features. We rejected edge features, because they do not contribute to classification significantly. There are not many significant edges in the vessels and in the background, so there is no need to recognize them. By this we reduced time to train a classifier on a small image from 3 to 2 minutes. We were assessing the effect of scale σ as well. We believe that wide range of sigma is important. However, to estimate the ideal range of scale, size of the image has to be concerned. So, in the second stage, we were searching for the ideal range of σ in full-resolution images.

Summary of pixel features and scales, that we decided to employ in classification of microcirculation images, is presented in table 2. It took approximately 24 hours to train a classifier with the training set containing 14 images.

Table 2: Summary of pixel features, which were chosen for pixel classification process in Ilastik. One color/intensity feature and two texture features were employed. All the features were used in wide range of scale σ.

	Features		
	Color/intensity	Texture	
	gaussian smoothing	structure tensor eigenvalue	hessian of gaussian eigenvalues
σ	0.7, 1, 1.6, 3.5, 5, 10, 30, 50		

Experiment concerning pixel features selection



Training data

(a) Manual labeling of (b)

🟥 clearly visible vessels
🟦 background

(b) Original image. We labeled pixels manually (a) and trained the classifier.

Testing data

(c) Input data for automatic classification

Probability maps

🟩 clearly visible vessels, 🟦 background

(d) Included feature groups: color/intensity, edges, texture σ: 0.7, 1, 1.6, 3.5, 5, 10

(e) Included feature groups: color/intensity, texture σ: 0.7, 1, 1.6, 3.5, 5, 10

(f) Included feature groups: color/intensity, texture σ: 0.7, 1, 1.6, 3.5

Figure 16: Experiment concerning pixel features selection. We were training the classifier on training data (a,b). We were choosing various combinations of pixel features groups (color/intensity, edges, texture) and scale σ. We were assessing probability maps of the classifications visually (d,e,f). In probability map, color intensity specifies the probability of a a correct classification of a particular pixel into a class. Darker tones indicate bigger uncertainty of classification. We display the first stage of experiment, when small pictures were employed to save time during training and classifying process. We found out, that texture based features do not participate in classification significantly - comparison between (d) and (e). However, scale definition plays an important role - (e) vs. (f). In (f), where scale is smaller, more details are classified correctly, but probability in bigger objects decreases.

6.2.2 Individual objects extraction

We unite classes 2), 3) and 4) in probability map into one class, so two classes probability map is created (vessels and background). We blur the two-classes probability map and then we transform it into individual objects by thresholding.

We were looking for two parameters, while extracting individual vessels: radius R of circular averaging filter and threshold probability value t. Blurring by averaging filter enhances the accuracy of segmentation in some cases, because it removes small holes and ragged edges in final binary segmentation.

We stated the parameters as follows: We had 7 manual segmentations of 7 pictures, which were not used for training the classifier (testing segmentations). We let the classifier create a probability map for each of these 7 images. Subsequently, we were blurring each map by filters with various radius. We thresholded each blurred probability map by various thresholds. In each case we computed the number of incorrectly classified pixels in comparison with training segmentation. We chose the combination of filter size and threshold, where the number of mistakes was minimal. Scheme of the procedure is displayed in figure 17.

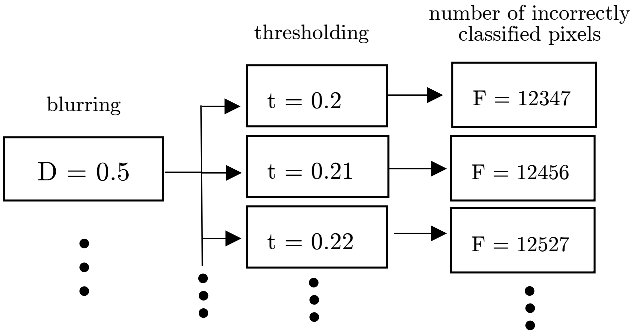


Figure 17: Determination of ideal filter size and threshold while extracting individual objects from probability map. Two classes probability map is blurred by a filter of certain size D. Then thresholding in different levels t is employed. For each combination of D and t the number of incorrectly classified pixels F is computed. The procedure is repeated many times for various D. Combination D and t, where F is minimal, is considered the best.

The combinations of filter size and threshold, which lead to minimal amount of incorrectly classified pixels for all testing pictures are displayed in table 3.

We use the average of values from all the testing pictures as the parameters of individual objects extraction procedure (column ‘average’ in table 3).

Table 3: Combination of threshold value and averaging filler diameter, which lead to minimal amount of incorrectly classified pixels. P1 - 7 are training pictures. We created their segmentations manually and compared manual segmentation with automatic segmentation.

	P1	P2	P3	P4	P5	P6	P7	average
radius	6.5	7	7	6	5.5	7	6	**6.5**
threshold	0.486	0.518	0.549	0.506	0.478	0.514	0.498	**0.507**

6.2.3 Segmentation accuracy

In table 4 we present the accuracy of segmentation. 7 testing images were classified. The probability maps were smoothed by circular averaging filter with radius 6.5 pixels and thresholded by level 0.507. We computed accuracy ACC of segmentation for each testing image P:

$$ACC = \frac{T}{N} \cdot 100 \quad (\%) \tag{19}$$

where T is number of correctly classified pixels and N number of pixels in image.

Next, we determined accuracy of vessel pixels classification ACC_V:

$$Acc_V = \frac{T_V}{N_V} \cdot 100 \quad (\%) \tag{20}$$

where T_V is the number of correctly classified vessel pixels and N_V the number of vessel pixels in manual segmentation.

Furthermore, we counted the ratio R_V of correctly classified vessel pixels T_V and number of vessel pixels in automatic segmentation N_{VS}

$$Acc_V = \frac{T_V}{N_{VS}} \cdot 100 \quad (\%) \tag{21}$$

Table 4: Accuracy of segmentation. P1-7 are testing pictures. We compared manual segmentations of these pictures with automatic segmentation. The average accuracy of two-classes segmentation is 93 % (parameter ACC). Concerning vessels, 78 % of vessel pixels from original image are classified as vessels in binary segmentation on average (parameter ACC$_V$). 80 % of vessel pixels in binary segmentation are labeled correctly on average (parameter R$_V$).

	P1	P2	P3	P4	P5	P6	P7	average
ACC	0.96	0.93	0.93	0.90	0.92	0.91	0.96	**0.93**
ACC$_V$	0.80	0.92	0.75	0.71	0.82	0.68	0.83	**0.78**
R$_V$	0.87	0.75	0.84	0.67	0.81	0.93	0.72	**0.80**

Average accuracy of two-classes segmentation is 93 % (parameter ACC). Both vessels and background classes are included in this parameter and there several times more background pixels in binary segmentation than vessels. To describe the accuracy of vessel pixels classification we compute ACC$_V$ and R$_V$. ACC$_V$ describes, how many of vessel pixels from original image is classified as vessels in binary segmentation. In our algorithm, 78 % of vessel pixels from original image are classified as vessels in binary segmentation. R_V describes, how many of vessel pixels in binary segmentation are labeled correctly. In average, 80 % of vessel pixels in binary segmentation are labeled correctly.

6.3 Individual segment identification

The diagram of individual segment identification process is displayed in fig.18 We find skeletons in binary image. Subsequently, we find branching points in the skeleton. We remove the branching points to separate individual skeleton segments. Separated continuous objects are labeled by unique identifiers in skeleton image. Based on the distance, vessel pixels are matched with close skeletons and labeled by their identifiers.

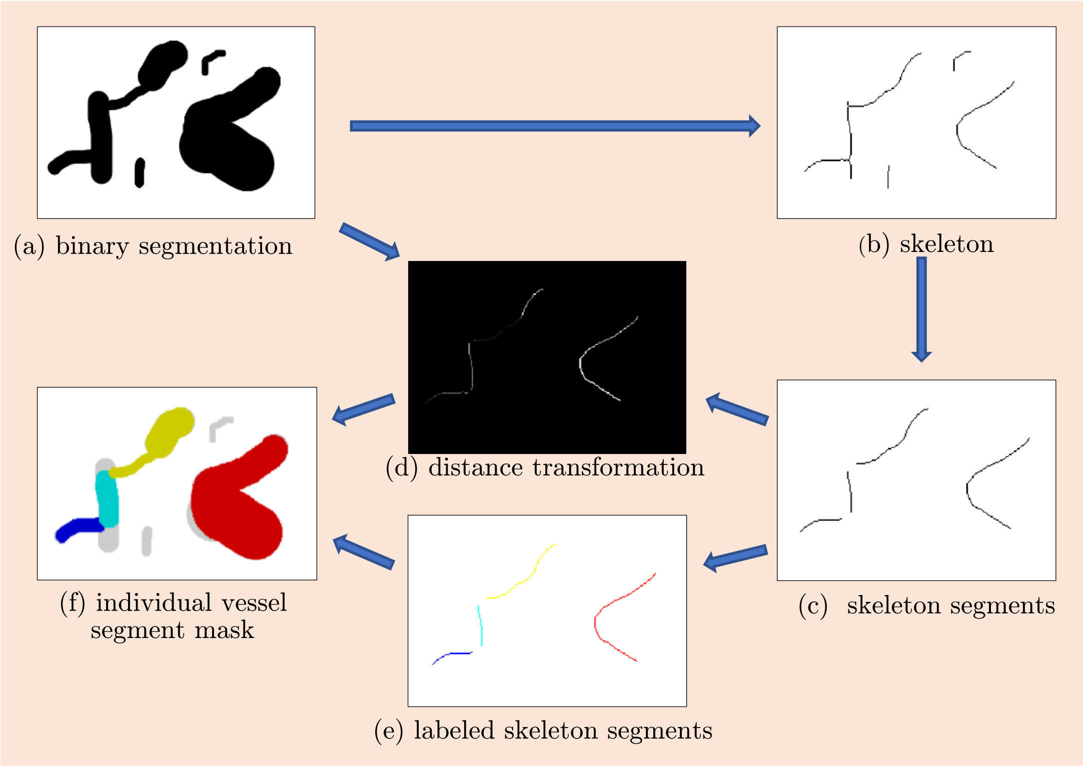

(a) binary segmentation
(b) skeleton
(d) distance transformation
(f) individual vessel segment mask
(c) skeleton segments
(e) labeled skeleton segments

Figure 18: Process diagram of segments labeling and dimensions measurement. Binary segmentation is an input. Mask, where each segment is labeled by unique classifier is the output.

Skeletonization is based on thinning of the binary image. Medial axis transformation described in [49] is used in the algorithm. While thinning, the neighbourhood of 4 pixels is used. Such thinning leads to the pruned skeleton without short branches.

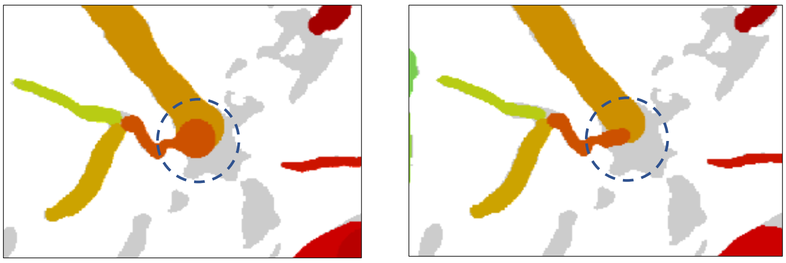
We separate skeleton segments by removing branching points. Each non-zero pixel, which has at least 3 another non-zero pixels in its 3×3 neighbourhood is labeled as a branching point. By removing the branching points we find out, how many individual vessels the frame contains. The individual vessel is meant to be the segment of the vessel between bifurcations. Naturally, bifurcation is the place, where diameter of vessel and blood flow velocity change.

To separate skeleton, we dilate the branching points into the disks with the radius of 9 pixels. We stated the radius empirically. Afterwards the branching points can be removed and separate skeleton segments remain.

Once the segments are separated, we find all the continuous areas, which is each 8-connected component. We sum the pixels in each continuous area to find out their lengths. In this stage it is easy to remove those segments, which are not long enough. Binary segmentation always contains some short segments, which do not label vessels. We remove the skeleton segments, which consist of 50 pixels and less. Afterwards each skeleton segment is labelled by a certain brightness value, which is a unique identifier.

6.3.1 Individual vessel segment mask creation

We create individual vessel segment mask by uniting circles along each centerline. Each centerline pixel is the center of one circle. The diameter of the circle equals distance of the centerline pixel from closest background pixel. All the circles along the centerline are united and the covered area is labeled by centerline's identifier. This approach causes, that individual vessel segment mask is cleaner than binary segmentation. Since we have removed too short centerlines, pixel along these centerlines are not included in individual vessel segment mask. Such pixels are displayed in grey color in figure 19. These are pixels we do not want to include into analysis, because they mostly belong to background or to too blurred vessels with insufficient contrast. This approach leads to unsatisfactory labeling close to bifurcations of thin and wide vessels. In such area the distance transform value of thin vessel centerline pixels is bigger than in the rest of the vessel. So, big circle is created (fig. 19a). We solve this problem by replacing the actual vessel radius by the average vessel radius in each centerline pixel, where the actual value is bigger than the average (fig. 19b).

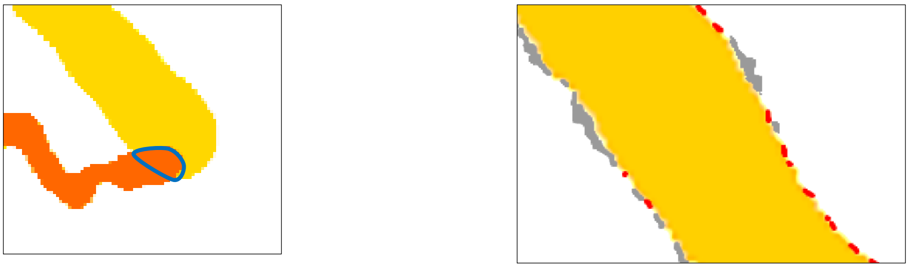


(a) before correction (b) after correction

Figure 19: Correction of diameter determination close to the bifurcations

There are two drawbacks in our labeling approach:

- The end of thin vessel is sometimes involved in large vessel. (fig.20a)
- Labeled area does not entirely follow the vessel in binary vessel segmentation image. (fig.20b)



(a)

Small vessel (orange) is not labeled correctly in the area (blue outline), which is close to the bifurcation with large vessel (yellow). Blue outline should be filled by yellow color, because the area belongs to the large vessels.

(b)

labeled area does not follow the vessel in binary vessel segmentation image entirely

🟨 correctly labeled area (true positive pixels)
⬛ false negative pixels - they are not labeled, even though they should be
🟥 false positive pixels - they are labeled, even though they should not be

Figure 20: Imperfections of labeling algorithm

6.4 Vessel segments radius measurement

We measure radius r by distance transform. It calculates euclidean distance from background in each non-background pixel. Distance from background is determined for each non-zero pixel, so it is determined for centerlines as well. Distance r_n of centerline pixel from background equals the radius of the segment measured in pixels in a certain section. As we know how many centerline pixels particular segment contains (N), we can compute the average radius of each vessel segment:

$$\bar{r} = \frac{1}{N}\sum_{n=1}^{N} r_n \tag{22}$$

7 Straight vessel segments extraction

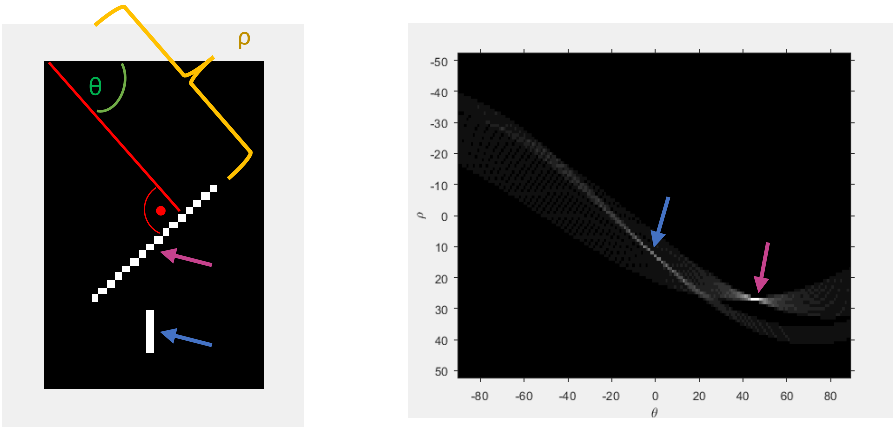
The aim of the operation is to obtain an image, which contains only the pixels, which belong to long enough straight part of the vessel as in figure 22g. We extract the segments to estimate optical flow there.

The process is based on Hough transform [47] of the vessel segment centerline.
The algorithm is based on the parametric representation of a line [47]:

$$\rho = x \cdot cos(\theta) + y \cdot sin(\theta) \tag{23}$$

ρ is the distance from the origin to the line along a vector perpendicular to the line. θ is the angle between the x-axis (figure 23a) and this vector. The peaks in Hough matrix represent potential lines in the input image. [48]

Deeper explanation of Hough transformation is provided in figure 23.

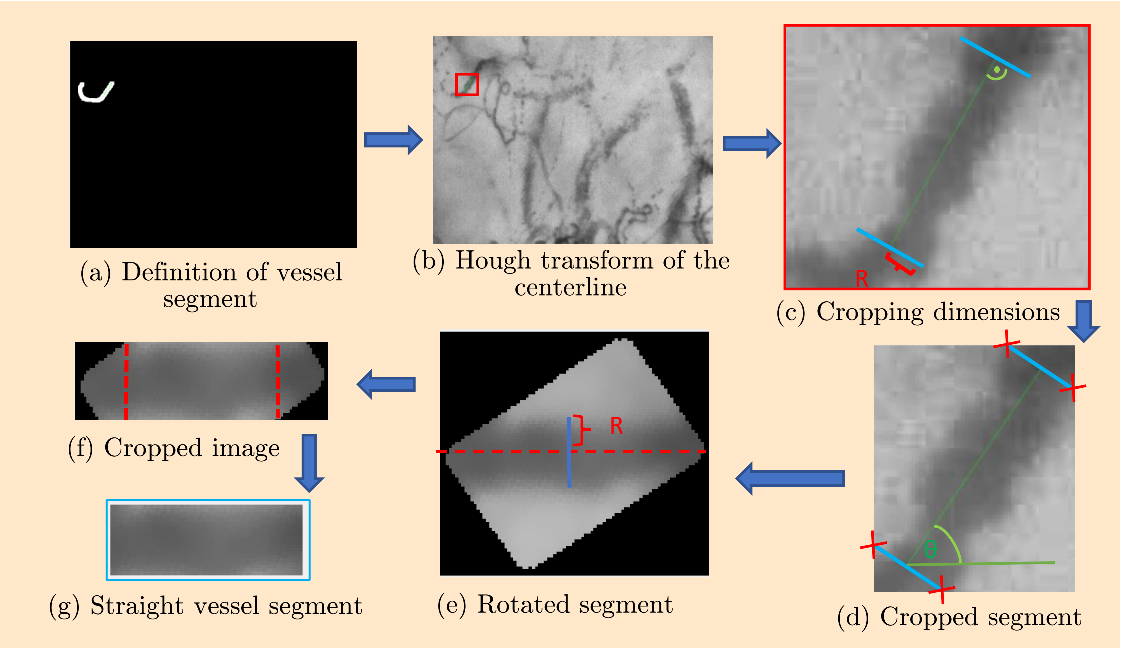


(a) Original image

(b) Hough transform matrix of original image

Figure 21: Explanation of Hough transform for line detection in binary image. Original binary image (a) contains two white lines. Each possible line can be determined by the distance from origin ρ and angle θ. Both parameters are displayed for longer line in original image. Based on original image Hough transform matrix (b), which is 2D histogram of lines occurnces, is created as follows: Through each white pixel of original imge imaginary lines with various directionns are drawn. Each of these lines is described by ρ and θ and added into appropriate bin in Hough transform matrix. When some white pixels are in a line in original image, appropriate bins in Hough matrix grow and get brighter, because imaginary line, which have direction of real line in original image, are added into the same bin. The brightest spots in Hough transform matrix indicate ρ and θ of real lines in original image (arrows).

First, we choose a segment (fig. 22a). We detect straight parts of its centerline by Hough transform (green line in figure 22c). Next we find the borders for cropping. The border points (red crosses) lie on the blue lines displayed in fig. 22c, which intersect the green line endpoints and hold the right angle with the green line. The euclidean distance of the border points from the endpoints equals the radius of the vessel. Since we know the angle θ, we can rotate the image (fig. 22e). Then we crop the rotated image (fig. 22f). The horizontal centerline of the cropped area is the horizontal centerline of the rotated image and the height of the cropped area equals to the diameter of the vessel. We remove all the rows and columns from the cropped image, which contain zeros, to obtain the final vessel extraction (fig. 22g).



(a) Definition of vessel segment

(b) Hough transform of the centerline

(c) Cropping dimensions

(f) Cropped image

(g) Straight vessel segment

(e) Rotated segment

(d) Cropped segment

Figure 22: Diagram of straight vessel segment extraction. (a) vessel segment is selected, (b) Hough transform of vessel segment centerline to detect straight section, (c) roi determination, (d) cropping and rotating of roi, (e,f) another roi definition and cropping, (e) resulting horizontally oriented straight vessel segment.

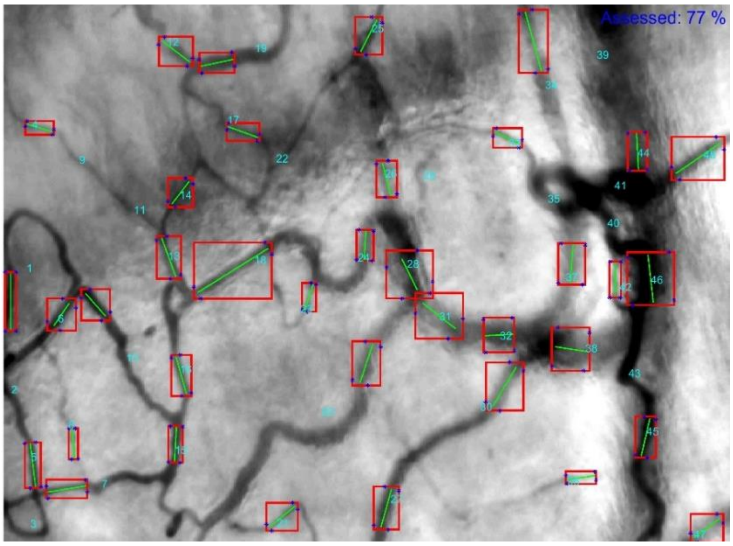
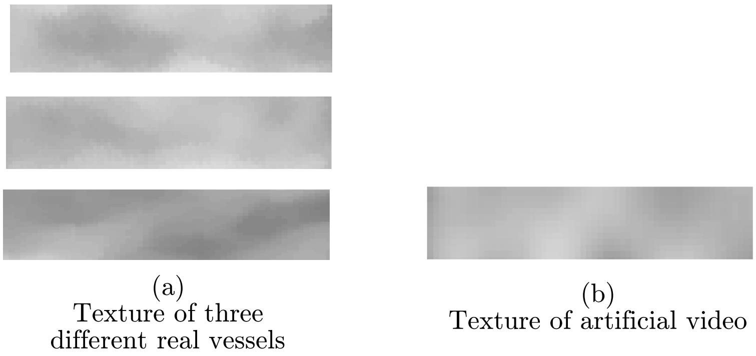


Figure 23: Detection of straight vessel segments within whole image. Segments will be extracted from the regions of interest, which are marked by red rectangles.

8 Artificial videos

By artificial videos we simulate RBCs flow. We try to imitate the inner structure of straight vessel segments (fig. 24). The diagram of artificial video acquiring is displayed in fig. 25. Initial 10×225 array with uniformly distributed random intensity values in range 0 - 255 was generated. The array was smoothed by gaussian filter. Each video is a sequence of 10 10×175 images, which were obtained from the initial array. First image of each video contains 1st - 175th column of the initial array. Depending on the inter-frame motion n, the second image of the video contains from $1+n$th to $175+n$th column of the initial array. Shifting the region of interest over the initiation array about n pixels, 10 frames of the video are created.


(a)
Texture of three
different real vessels

(b)
Texture of artificial video

Figure 24: Comparison of texture, (a) real vessel segments, (b) artificial video.

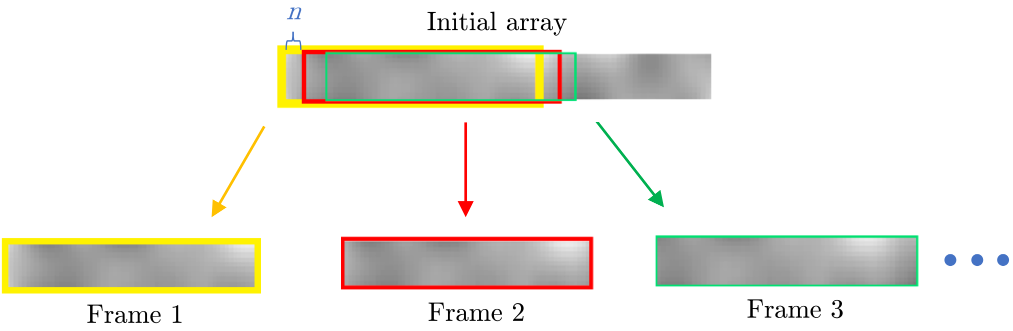


Figure 25: Creation of artificial videos. Region of interest is shifted by n pixels over the initial array, which simulates the pattern of real vessels.

9 Velocity measurement

9.1 Maximal blood flow velocity

We determine maximal RBCs velocity, because it is important to know the range of motion we need to measure. According to [19] RBCs velocity v can reach up to 1.5 mm·s^{-1}. Frame rate FR of our videos is 43 fps, motion a between two frames caused by maximal velocity v is

$$a = \frac{v}{FR} = 35\ \mu m \tag{24}$$

Each pixel of MicroScan videos occupies the area $S = 0.56$ µm^2 [58], so length l of pixel side is

$$l = \sqrt{S} = 0.75\ \mu m \tag{25}$$

Interframe motion m (in pixels) performed by RBCs at velocity v is then

$$m = \frac{a}{l} = 46.7\ pixels \tag{26}$$

Ideally, we need to be able to detect the interframe motion of around 50 pixels to cover the range of RBCs velocity.

9.2 Optical flow

We decided to implement optical flow, as recommended in [19]. Unlike [19] we implemented Lucas-Kanade method.

Optical flow is a field of motion vector per each pixel of an image. The vectors show movement of the pixel between two sequential pictures in time. That is, what optical flow equation for a pixel *x,y* claims [51]:

$$I_x(x,y)\boldsymbol{v}_x + I_y(x,y)\boldsymbol{v}_y = -I_t(x,y) \tag{27}$$

where $I_x(x,y)$, $I_y(x,y)$, $I_t(x,y)$ are the partial derivatives of the image I with respect to position x, y and time t and ($\boldsymbol{v}_x$, $\boldsymbol{v}_y$) is the motion vector.

Lucas-Kanade method is based on the assumption, that the displacement of the image contents between following frames is small and locally approximately constant. So there is the same motion vector ($\boldsymbol{v}_x$, $\boldsymbol{v}_y$) for all pixels within a window. The compromise solution is based on the least squares method.

The optical flow equations for all n pixels of the window can be written in a matrix form

$$A\boldsymbol{v} = \boldsymbol{b} \ , \tag{28}$$

where

$$A = \begin{bmatrix} I_x(x_1,y_1) & I_y(x_1,y_1) \\ \vdots & \vdots \\ I_x(x_n,y_n) & I_y(x_n,y_n) \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_y \end{bmatrix} \quad b = \begin{bmatrix} -I_t(x_1,y_1) \\ \vdots \\ -I_t(x_n,y_n),(x_n,y_n) \end{bmatrix} \tag{29}$$

Equation (28) resembles equation (7) in section 5.2 (section Motion estimation in chapter Correction of video instability). Unlike (28) eq. (7) has a form of error function. The equations differ in the number of parameters to be estimated. In eq. (7) 3 parameters for each spatial direction are assessed, in equation (28) just 1 parameter for each spatial direction is estimated. Moreover, in eq. (7) all the pixels of an image are included. In contrast, Lucas Kanade method estimates the motion vector just for a local area.

Vector $\boldsymbol{v}$ is computed by least squares method as:

$$\begin{bmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} w_i I_x(x_i, y_i)^2 & \sum_{i=1}^{n} w_i I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_{i=1}^{n} w_i I_y(x_i, y_i) I_x(x_i, y_i) & \sum_{i=1}^{n} w_i I_y(x_i, y_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} w_i I_x(x_i, y_i) I_t(x_i, y_i) \\ -\sum_{i=1}^{n} w_i I_y(x_i, y_i) I_t(x_i, y_i) \end{bmatrix} \tag{30}$$

Equation (30) resembles eq. (12) in section 5.2, but, as we have mentioned, both approaches differ in number of parameters.

We implement equation (30) as follows:

$$\begin{bmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_y \end{bmatrix} = \begin{bmatrix} I_{xx}(x, y) & I_{xy}(x, y) \\ I_{xy}(x, y) & I_{yy}(x, y) \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} w_i I_x(x_i, y_i) I_t(x_i, y_i) \\ -\sum_{i=1}^{n} w_i I_y(x_i, y_i) I_t(x_i, y_i) \end{bmatrix} \tag{31}$$

where

$$I_{xx} = {I_x}^2 * W \tag{32}$$

$$I_{yy} = {I_y}^2 * W$$

$$I_{xy} = I_x I_y * W$$

where I_x is partial derivative of image I with respect to x and I_y is partial derivative of image I with respect to y. W is matrix of n^2 elements, where the central element equals 1 and every other element *p(i,j)* equals a Gaussian function of its distance from central element of the matrix:

$$p(i,j) = \frac{1}{\sqrt{2\pi}} e^{-\frac{d^2}{2}} \tag{33}$$

where *d* is distance of element *p(i,j)* from the central element of the matrix in pixels.

We compute partial derivatives as convolution of image *I* with kernel d = 1/12[-1,8,0,-8,1], as proposed in [61]. For I_x kernel is oriented horizontally, for I_y vertically.

I_{xx} , I_{yy} , I_{xy} are such matrices, where each element equals the sum of weighted partial derivatives in a window of side *n.* So, each element of appropriate matrix equals the appropriate summing expression in matrix in equation (30).

The derivative in respect to time is computed as the difference of consecutive frames *I(t)* and *I(t+1)* :

$$I_t = I(t+1) - I(t) \tag{34}$$

We implemented multiresolution technique to enlarge the recognition range of optical flow (just like in section 5.2.1, where the approach of finding affine transformation model of an image in different scales is described)

Multiresolution method is based on reducing the image resolution. However, the width of some vessel segments is quite often just around 10 pixels and the window size is typically around 10×10 pixels. We need enough pixels in the region of interest to fit the window, so we can not change the resolution of vessels too much. We solve this problem by reducing just one dimension of vessel segment. When the initial size of vessel segment is 10×100 we reduce it to 10×50. Like this fitting of window is enabled. We suppose, that RBCs move just in one direction, so reduction of resolution in another direction is not necessary. To ensure this, we extract straight vessel segments and rotate them, so they are always oriented vertically.

In the coarsest level L resolution of frame is $m\times n$. Initial array of motion vector v_{xL}, v_{yL} with dimensions $m\times n$ is estimated. In a finer level, L-1, resolution of frame is $2m\times 2n$. To adapt the resolution of motion vector array for level L-1, it is interpolated, so its resolution equals the resolution of frame $2m\times 2n$. To transfer the value of motion vector into level L-1, v_{xL} is multiplied by 2, v_{yL} remains the same, because we do not change the resolution in vertical direction (eq. 35).

$$v_{xL-1} = 2v_{xL} \tag{35}$$

$$v_{yL-1} = v_{yL}$$

In level L-1, coordinates of local window of frame $I(t+1)$ are transformed

$$x'_i = x + v_{xL-1} \tag{36}$$

$$y'_i = y + v_{yL-1}$$

and the window is interpolated. Again, motion vector $\boldsymbol{v_x}$, $\boldsymbol{v_y}$ is computed and the result is added to current estimation. Then the value of vector is transferred into finer level:

$$v_{xL-2} = 2(v_{xL-1} + v_x) \tag{37}$$

$$v_{yL-2} = v_{xL-1} + v_x$$

Equations (34-36) describe the estimation of motion vector during multiresolution process. Finally, vector $\boldsymbol{v}$ we search for, is the estimation from the finest level 1.

$$\boldsymbol{v} = \begin{bmatrix} v_{x1} \\ v_{y1} \end{bmatrix} \tag{38}$$

Finally, we compute the size s of motion vector:

$$s = \sqrt{v_{x1}^2 + v_{y1}^2} \tag{39}$$

Size of motion vector describes the interframe motion (in pixels) of a certain pixel between two frames.

The lengths of the vessel segments vary from 60 to 160 pixels in our videos. So, we apply 5-levels multiresolution on each vessel and we can be sure, that in some level the window fits into the frame.

We filter the images before down sampling from level n to level n-1 by binomial filter (1/4, 1/2, 1/4) along horizontal dimension, as proposed in [23]. We never up sample the reduced images, but we always down sample the original image to reach demanded resolution.

We use the square window with n^2 elements, where n equals 7. We do so, because such a window fits into the most frames we analyse. The minimum height of the frames to be analysed is 7 pixels in our case. Bigger window leads to a more precise estimation of velocity vector. From our experience, window with the side of 7 pixels is a good compromise between the demand of the image size and estimation quality.

9.2.1 Experiment with artificial vessel segments

We executed an experiment, where we examined, whether blood flow velocity is determined correctly. We tested the algorithm on artificial videos with known inter-frame motion. 50 videos were included in the experiment. There was a specific inter-frame motion in each video in the range from 1 to 50 pixels/frame. We tested the performance of the algorithm in this range of velocities, because such range should be found in real microcirculation videos.

We used histograms of *estimations vectors*. *Estimations vector,* is a group of all the estimated sizes of optical flow vectors from all the frames within the video (except the last frame).

We display a histogram of estimations vector in figure 26. The vector was extracted from artificial video with inter-frame motion $n = 5$ pixels. We can see, that the most of the estimated motions rank among 4.5 and 5.5 pixels. It is a correct value, because the real inter-frame motion is 5 pixels·frame^{-1}. However, other estimated values occur in the histogram as well, which means the algorithm is not perfect and the estimation is not always correct.

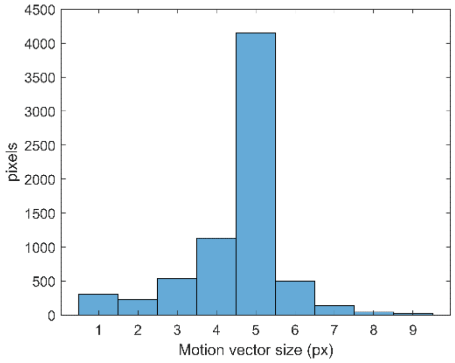
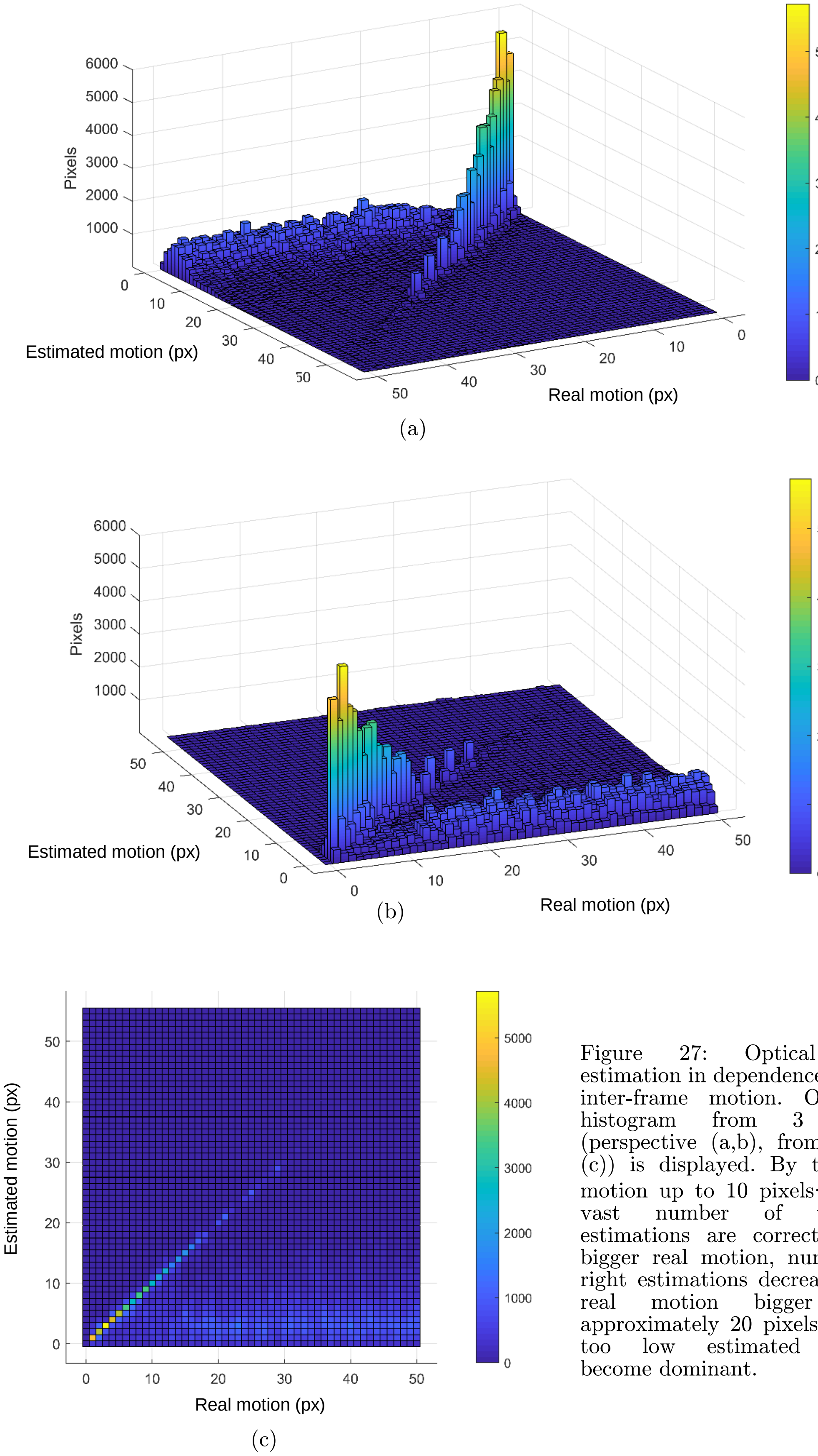


Figure 26: Histogram of motion vector sizes estimated in a sequence of frames of artificial vessel segment. Blood flow velocity descriptor is extracted from this histogram. In this particular case the descriptor equals 5, because most of the estimated motion vector sizes within the frame sequence rank among 4.5 and 5.5 pixels.

To see efficiency of the algorithm in a full range of requested velocities, we extracted estimations vectors for all videos. We created 2D histogram, where all the histograms of all estimations vectors are included (fig. 27). We did so in order to see the distribution of optical flow estimation in dependence on the inter-frame motion.

In fig. 27 one 2D histogram from three different views is displayed. In (a) and (b) perspective view is provided. In (c) the view from above is shown. In all cases *x* axes denotes to the real interframe motion and on *y* axis the particular estimations vector is displayed. We can see, that when the real motion is small, approximately up to 10 pixels·frame^{-1}, a significant amount of estimations vector elements are assessed correctly. Obviously, when the real motion is bigger, correct estimations occur as well, we can observe this up to the motion of around 30 pixels. However, the bins with correctly assessed values become less dominant with increasing velocity. To complete the information about how our algorithm is efficient or not, we provide an ideal histogram (fig. 28), which would describe the performance of a faultless algorithm, if it was included in our experiment. In contrast with the ideal implementation, real algorithm fails, when the real motion is bigger than approximately 20 pixels·frame^{-1}, because the bin with correctly estimated values become smaller than the faulty bins.



(a)

(b)

(c)

Figure 27: Optical flow estimation in dependence on the inter-frame motion. One 2D histogram from 3 views (perspective (a,b), from above (c)) is displayed. By the real motion up to 10 pixels·frame^{-1}, vast number of velocity estimations are correct. With bigger real motion, number of right estimations decreases. By real motion bigger than approximately 20 pixels, faulty too low estimated values become dominant.

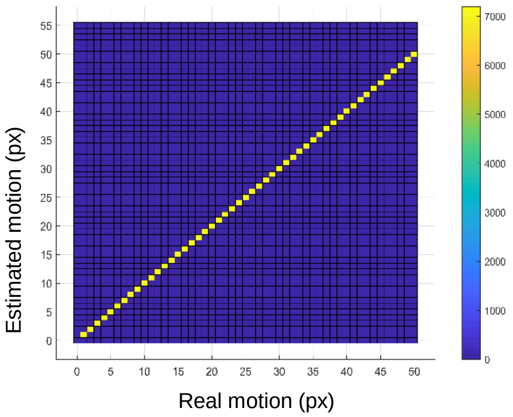


Figure 28: Ideal optical flow estimation in dependence on the inter-frame motion. This graph illustrates, how the optical flow would be estimated by the faultless algorithm. We can compare this graph with the performance of real algorithm (27c) to see the limits of the real implementation. Ideal algorithm would be able to estimate the motion up to 50 pixels·frame^{-1}, so that it could measure whole range of RBCs velocity. However, real algorithm fails when the real motion is bigger than 20 pixels.

9.2.2 Optical flow-based velocity descriptor

The goal of the implementation of velocity measurement algorithm is to obtain a descriptor of the flow in a whole video. To do so, we first create a descriptor of flow in a single vessel segment. It is the highest bin of the histogram of estimations vector, when the edges of histogram are half-integers (which means it is the most frequent estimation). Having a set of these individual descriptors (for each vessel segment of video, or group of segments) we can process it to create an overall velocity descriptor. Again, we choose the most frequent value to be the descriptor of video.

10 Temporal pixel intensity fluctuations

We are interested in temporal changes in intensity of each pixel of a vessel segment. Since the average brightness of each vessel segment is different, standard deviation is a suitable tool for compensating such variances [4]. So, we compute average value *s* of changes in temporal intensity of particular pixel (*I(x,y)*) as a standard deviation:

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(I(x,y) - \bar{I})^2} \tag{40}$$

where *N* stands for number of frames in video and $\bar{I}$ is the average intensity of pixels in a video segment within a video:

$$\bar{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{i=1}^{n} I(x_n y_n)}{n} \tag{41}$$

where *n* is the number of pixels of vessel segment.

As a descriptor of flow in a particular segment we choose the most frequent value *s* - the tallest bin in a histogram of intensity fluctuations *s.*

To describe whole video (or a group of vessels) we use the most frequent value of descriptors obtained from individual vessels. We display a histogram of individual vessel descriptors extracted from real video in figure 29. In this particular case the video descriptor equals 110. From now we will use an acronym PIF (pixel intensity fluctuations) for this descriptor.

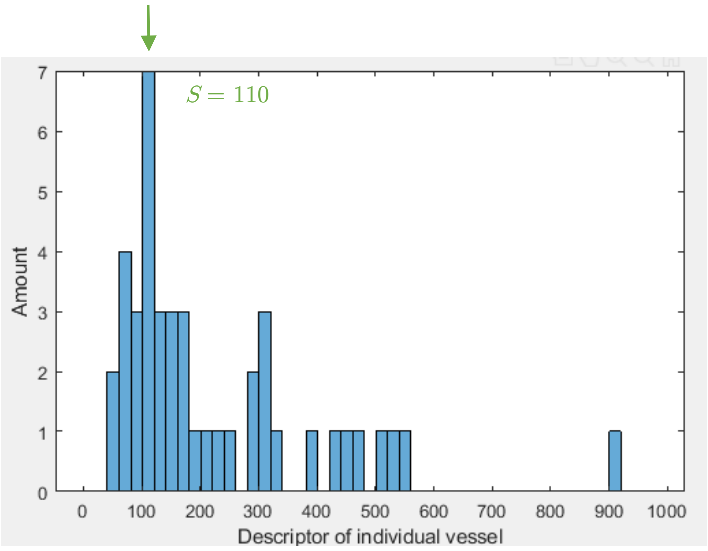
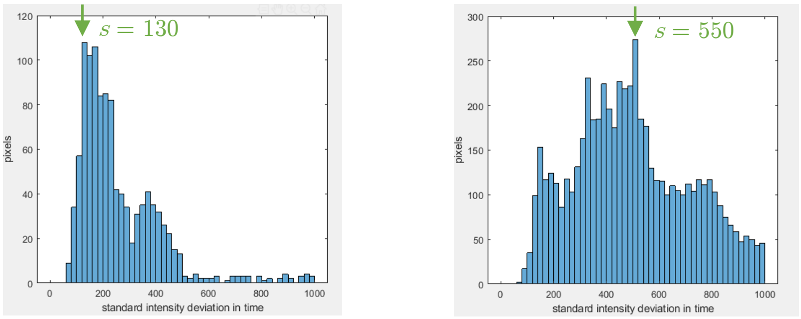


Figure 29: Histogram of descriptors of individual vessels from one video. Highest peak of the histogram is chosen to be the descriptor of the video (110).

10.1 Experiment with real vessel segments

We extracted 30 segments with good contrast and various flow velocities (visually apparent) from real videos and generated histograms of intensity fluctuations for each of them. Based on the visual assessment of several histograms we decided to set the edges to the multiples of 20. We display some histograms of intensity fluctuations from two real segments to show, that fluctuations in real vessels differ (fig. 30). We can see, that the highest peak in histogram is different in both cases (green arrow), so it makes sense to accept it as a distinctive descriptor of vessel segment. We provide values of descriptor of flow in each vessel segment in figure 31. Based on fig. 31, we believe, that pixel intensity fluctuations are different in real vessel segments with various flow velocity.



(a)

(b)

Figure 30: Comparison of histograms of pixel intensity fluctuations from two different real vessels. Difference of those vessels was apparent in video visually. Vessel (a) was thin, flow was intermittent and fast. Vessel (b) was thick, flow was continuous, velocity was heterogenous, RBCs were very dense.

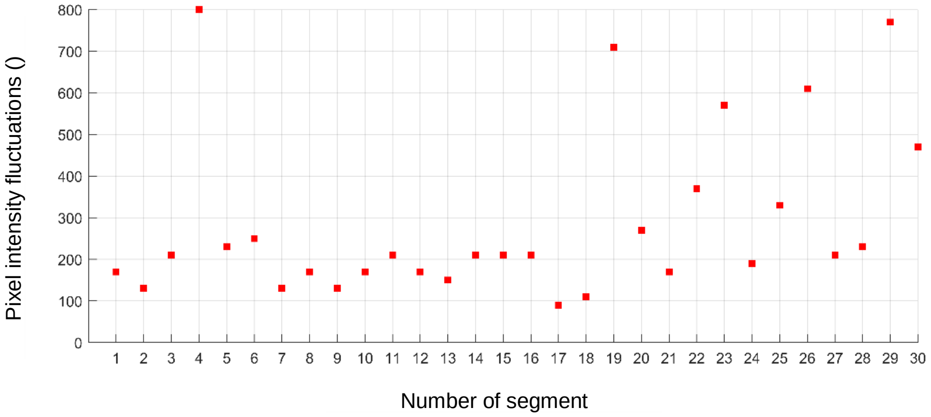


Figure 31: Intensity fluctuations of 30 real vessel segments with different velocities. In the graph we can see, that pixel intensity fluctuations are different in vessel segments with various flow velocity.

11 Implementation of conventional descriptors

11.1 TVD

TVD is implemented as

$$TVD = \frac{N_v}{N} \tag{42}$$

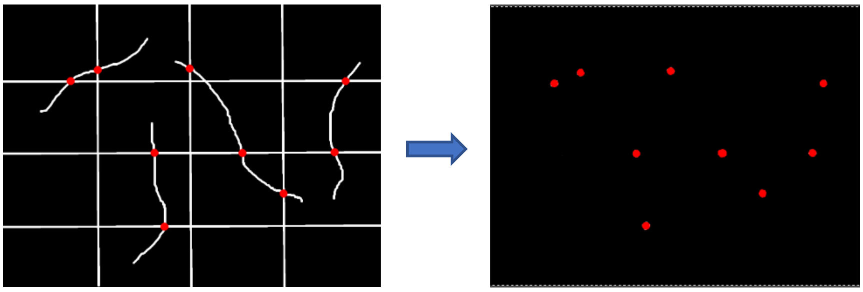
where N_v is the number of pixels in vessels and N number of pixels in a frame.

11.2 deBacker score

$$deBacker\ score = \frac{1000c}{3\sqrt{S}(m+n)} \tag{43}$$

where c is the number of crossing points, S is the area of one pixel, m,n are the dimensions of video in pixels.

We determine c as follows: A matrix of size $m\times n$ with three straight lines in x and y direction is created. Lines equal one and the background equals zero. We multiply matrix with lines with centerline image of a video, where ones label vessel segments centerlines and the rest equals zero. After multiplication the intersection of centerlines and lines equals one, rest equals zero. We sum up all the continuous objects to obtain number of intersections.



(a) Multipliacation of centerline image and deBacker lines image

(b) Crossing points

Figure 32: Determination of crossing points in deBacker score.

12 Classification of microcirculation

We defined three groups of vessels according to their diameter (based on the observation, so that all the groups have about the same amount of members in a video)

Table 5: Three groups of vessel segments in relation to their diameter, which we used in classification experiment.

vessel category	diameter (μm)
small	(0,13⟩
middle	(13,20⟩
large	>20

We extracted OFD, TVD, PIF and deBacker score for each group of vessels from each video in our dataset. We created description vector for each video. An example of such a vector is shown in table 6. We generated deBacker score for small+middle vessels (DB s+m) too (which means for vessel with diameter up to 20 μm), because AVA generates such parameter.

Table 6: Example of description vector, which is the outcome of video analysis. PIF: pixel intensity fluctuations, DB: deBacker score, TVD: total vessel density.

OFD all	OFD small	OFD middle	OFD large	PIF all	PIF small	PIF middle	PIF large	DB all	DB small	DB middle	DB large	DB s+m	TVD all	TVD small	TVD middle	TVD large
1	1	1	1	80	120	130	80	6.662	3.807	1.427	1.427	2.855	0.151	0.041	0.052	0.057

We generated description vectors from 75 on-pump videos and 76 off-pump videos. We created testing dataset so, that we chose one third from each group randomly. The rest was training dataset.

We trained SVM (support vectors machine) to discover a possible linear separability of data. We trained a linear classifier, because the dataset is small. We employed R statistical language [69] to do so. The aim of our investigation was to find out, whether any of parameters in description vector enables linear separation of pathological and physiological microcirculation. We did not combine the parameters, we trained a classifier for each of them. In each case we did 10-fold cross validation.

13 Results

13.1 Optical flow

We present the dependence of estimated velocity value on the real velocity of flow in figure 33. Green color marks the correct velocity estimation, red color codes the false estimation. The graph illustrates, that up to the velocity of 17 pixels/frame and for velocities 21, 25 and 29 pixels/frame the estimation was correct. Velocities in the range from 18 to 50 pixels/frame, except 21, 25 and 29 pixels/frame were not determined correctly.

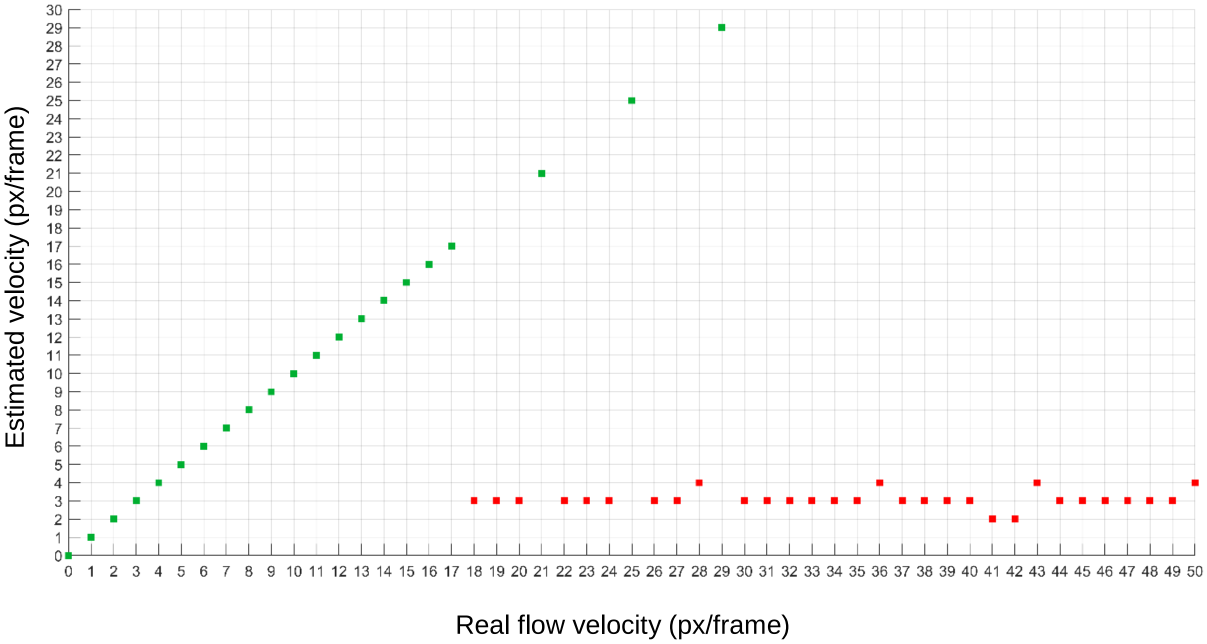


Figure 33: Performance of optical flow-based velocity estimation. Real flow velocity in artificial vessel segments on x axes, estimated velocity value on y axes. Green color marks the correct velocity estimation, red color codes the false estimation. The graph illustrates, that up to the velocity of 17 pixels/frame, velocity in artificial vessel segments is determined correctly by optical flow. In higher velocities it is mostly misleading.

13.2 Comparison of deBacker score with reference values

There are two deBacker score parameters, which both our and AVA measured on identical data - deBacker score for all vessels and deBacker score for small vessels with diameter up to 20 µm.

We provide comparison of both performances. We computed difference D of parameter values P:

$$D = \left| \frac{P_{AVA} - P}{P_{AVA}} \right| \cdot 100 \quad (\%), \tag{44}$$

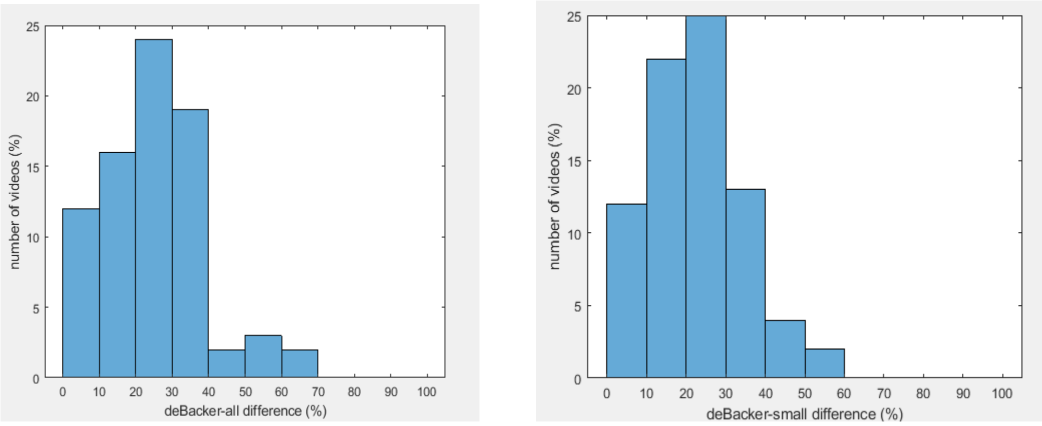
where P_{AVA} is the value AVA determined and P value obtained by our algorithm. The average difference $\overline{D}$ was computed

$$\overline{D} = \frac{1}{N} \sum_{n=1}^{N} D_n \tag{45}$$

where $N = 72$, which is is the amount of videos we analysed.

In parameter deBacker score, the average difference of values generated by both compared algorithms was 25.03 %. Histogram of differences is displayed in figure 34a.

In parameter deBacker-small, where only vessels with diameter up to 20 µm were included, the average difference was 22.42 %. Histogram is shown in figure 34b.



(a) deBacker score (all) difference (%)

(b) deBacker score (small) difference (%)

Figure 34: Histograms of differences in results achieved from commercial software and our algorithm. 72 videos were assessed. (a) parameter deBacker score (all), the most of the differences range between 0 and 40 %. (b) parameter deBacker score (small), the most of the differences range between 0 and 40 %.

13.3 Classification of pathological and physiological microcirculation

We trained SVM classifier for the parameters our software provides - TVD, deBacker score (DB), PIF, OFD. Each parameter was assessed in 4 groups of vessels according to their dimensions - small, middle, large and all.

Accuracy and p-value of all the classifiers is displayed in table 7. Based on table 7 we claim, that we found a significant difference between pathological microcirculation (patients on CPB) and physiological microcirculation for three of the measured parameters: pixel intensity fluctuations, PIF (all), (p=0.034), pixel intensity fluctuations, PIF (large), (p=0.001), deBacker score (all), (p=0.035). The difference between other measurements were not statistically significant.

Table 7: Accuracy of classification of testing data. In each column performance of one classifier is displayed. Such cases, where we found significant difference, are highlighted.

	OFD all	OFD small	OFD middle	OFD large	**PIF** **all**	PIF small	PIF middle	**PIF** **large**
accuracy	0.5	0.5	0.5	0.5	**0.64**	0.62	0.60	**0.72**
p-value	0.556	0.556	0.556	0.556	**0.034**	0.059	0.102	**0.001**

	DB **all**	DB small	DB middle	DB large	TVD all	TVD small	TVD middle	TVD large
accuracy	**0.64**	0.54	0.58	0.50	0.62	0.53	0.55	0.5
p-value	**0.035**	0.336	0.161	0.556	0.059	0.335	0.342	0.556

14 Discussion

Our work provides three significant findings concerning pixel intensity fluctuations, difference of on-pump and off-pump videos and optical flow. 1) We do not suppose optical flow computed by Lucas-Kanade method is sufficient for microcirculation analysis. 2) We suppose, that pixel intensity fluctuation is promising quantitative descriptor of microcirculation. 3) We found significant difference between on-pump and off-pump videos. All the statements will be discussed.

14.1 Optical flow

Our implementation of Lucas Kanade method in combination with multiresolution performs quite well on artificial data. In artificial videos with wide range of velocities (1-17 pixels/frame) the algorithm estimates velocity correctly. There is a hint of correct estimation even in higher velocities up to 30 pixels/frame (the descriptor would have to be improved). However, optical flow fails in real data. It seems it does not provide any relevant information about real videos. We see several causations. Firstly, there is lack of significant texture in vessel segments. The vessel content is just blurry without any corners or edges quite often. Moreover, when texture is present, it changes between frames, so there is no chance to find similarities. Furthermore, pictures are too noisy. We tried to filter the images by diffusion filter, but it did not bring any improvement in performance of optical flow. In addition, the content of extracted vessel segments might be insufficient in some cases because of wrong segmentation and straight segment extraction. We did not have enough time to check all the extracted areas, whether they really include just area of vessels. Finally, according to the experiment with artificial video, our implementation fails by higher velocities than 17 pixels/frame. In some cases this may be the reason of dysfunction in real videos, because velocities over 17 pixels/frame are supposed to be normal in capillaries.

14.2 Pixel intensity fluctuations

Pixel intensity fluctuations seem to be useful tool for quantitative microcirculation analysis, because it passed through all our experiments successfully. We found out it related with flow velocity in artificial vessels, it differed for real vessels with various flow velocity and it enabled statistically significant linear classification of pathological and physiological microcirculation either in all or just in large vessels. PIF just describes the video content without any estimations and assumptions, which we consider convenient. It is relatively resistant to noise and lack of contrast because of averaging. Nevertheless, we should mention, that one estimation in PIF determination process exists. It is the width of bins in histogram of parameter s for particular vessel. We accept the centre value of the tallest bin of histogram as the descriptor of particular vessel. We stated the width of bins to be 20 empirically, which is not an ideal solution. If the width of bin was different, PIF values would differ too.

More data from various patients would be needed to analyse, what really PIF describes. There is actually no reason not to accept it as a quantitative descriptor for scientific purposes. Informational value of the descriptor might be revealed in the future, when enough data from various patients would be gained for statistical analysis.

14.3 Difference of on-pump and off-pump videos

We found a significant difference in 3 from 16 parameters between on-pump and off-pump patients, which confirms the assumption, that the difference exists. However, the dataset was very small, just 76 off-pump and 75 on-pump videos. We used cross-validation when training classifiers to compensate the lack of data. Moreover, the videos are very diverse. We suppose, that there are off-pump and on-pump videos, but they have many subcategories. On-pump videos include patients both before and after surgery and in anaesthesia. All these videos might differ from each other. We can find literature, which claims, that there is significant difference between microcirculation of subjects with and without anaesthesia [8,67]. Considering off-pump videos, they were captured in different stages of surgeries. In [13] quality of microcirculation with and without CPB was compared, significant difference was registered in the beginning of surgery, but no variation was found in later stages.

We found difference in deBacker score and in PIF. DeBacker score relates to the amount of vessels and their distribution in captured area, but it does not provide any information about vessel content. In contrast, PIF describes flow, but it has nothing in common with amount of vessels or their dimensions. It means we found a difference both in vessel amount and flow. TVD relates to the area vessels occupy in captured area. We did not find any significant difference in this parameter.

14.4 Software

There are some flaws, which negatively influence the reliability of microcirculation descriptors. Firstly, stabilization of videos is never perfect. We controlled the stability visually and kept those videos, which we subjectively considered stable enough, but in some cases some vessel segments were not stable entirely. However, we could not have risked to reject more videos because of lack of data. Moreover, subjective evaluation of stability causes, that the quality differs within dataset. Automatic control of stability would be useful in this case.

Automatic vessel segmentation is performed by classifier, trained on 14 training pictures, which is not too much. Machine learning-based segmentation is attractive, but the creation of training and testing data is time-consuming. Looking at the automatic segmentation we should remember, that approximately 20 % of vessel pixels are classified wrong and that roughly 22 % of vessel pixels from original image are not included among vessel pixels of automatic segmentation. Another inaccuracy is caused by dividing segmentation objects into individual segments. Firstly, areas around bifurcations are tricky in our software. No border recognition is implemented, so sometimes a vessel segment ends in the area of another segment. Part of the information from such segment is misleading, because it contains information from two different vessels. Another distortion is caused by the fact, that we always separate vessels in bifurcations. In case there is a long thick vessel and there are three bifurcations along its length, we divide this vessel into three segments. Subsequently, the analysis contains data from this vessel three times instead of once.

14.5 Comparison with AVA

The difference of deBacker score determined by our software and deBacker score from AVA is roughly 25 %. We only compared 72 values, because there were no more suitable videos. We accept values from AVA as reference, because we assume it is more accurate software than ours. The difference is caused by the imprecisions of our software. DeBacker score value depends on the number and distribution of detected vessels. Inaccuracy in stabilization and subsequent automatic vessel segmentation causes, that not all vessels are detected, which influences deBacker score value. On top of that, we can not be sure, how precise AVA is in stabilization and vessel detection.

15 Conclusion

We created a functional software for automated analysis of microcirculation videos. Video stabilization, automatic vessel detection and their dimensions measurement are implemented. MatLab [68] was used to implement stabilization and dimensions measurement, Ilastik tool [24] is employed for vessel detection. The software provides quantitative analysis of microcirculation (implemented in MatLab). It determines two conventional (deBacker score, total vessel density (TVD)) and two novel descriptors (pixel intensity fluctuations (PIF)), (optical flow descriptor (OFD)) automatically. We compared our deBacker score values with those from commercial tool. The average difference was approximately 25 %.

We designed and constructed two novel microcirculation descriptors. OFD is based on the analyse of microcirculation velocity profile. Flow velocity is estimated by optical flow method. We discovered experimentally, that the descriptor estimates flow velocity in range 0-17 pixels/frame correctly in artificial videos. However, we found out it was not suitable for assessment of real videos due to insufficient texture and noise. For this reason we did not implement it as an automatic feature of the software. However, it can be added any time, if better quality videos would be analysed.

The second novel descriptor (PIF) is based on the analyse of profile of pixel intensity fluctuations in video. Despite OFD it performs well on real videos.

All the descriptors are assessed for small, middle and large vessels, just like for all the vessels at once. We determined these parameters for 151 videos, which of 76 were captured on normal subjects and 75 on subjects with CPB. We trained linear SVM classifier to distinguish on-pump and off-pump videos. We employed R to create the classifier. We found significant difference in 3 parameters: PIF (all vessels), (p=0.034), PIF (large vessels), (p=0.001), deBacker score (all vessels), (p=0.035).

The instructions for the use software are described in Appendix.

16 References

[1] DE BACKER, Daniel, Steven HOLLENBERG, Christiaan BOERMA, Peter GOEDHART, Gustavo BÜCHELE, Gustavo OSPINA-TASCON, Iwan DOBBE a Can INCE. How to evaluate the microcirculation: report of a roundtable conference. Critical Care. 11(5). DOI: 10.1186/cc6118. ISSN 13648535. Dostupné také z: http://ccforum.biomedcentral.com/articles/10.1186/cc6118

[2] JACKSON, William F., Steven HOLLENBERG, Christiaan BOERMA, Peter GOEDHART, Gustavo BÜCHELE, Gustavo OSPINA-TASCON, Iwan DOBBE a Can INCE. Microcirculation: report of a round table conference. Muscle. Elsevier, 2012, 2012, 11(5), 1197-1206. DOI: 10.1016/B978-0-12-381510-1.00089-2. ISBN 9780123815101. ISSN 13648535. Dostupné také z: https://linkinghub.elsevier.com/retrieve/pii/B9780123815101000892

[3] GOEDHART, P. T., M. KHALILZADA, R. BEZEMER, J. MERZA, C. INCE, Gustavo OSPINA-TASCON, Iwan DOBBE a Can INCE. Sidestream Dark Field (SDF) imaging: a novel stroboscopic LED ring-based imaging modality for clinical assessment of the microcirculation. *Optics Express*. Elsevier, 2007, 2012, **15**(23), 1197-1206. DOI: 10.1364/OE.15.015101. ISBN 9780123815101. ISSN 1094-4087. Dostupné také z: https://www.osapublishing.org/abstract.cfm?URI=oe-15-23-15101

[4] BEZEMER, Rick, Johannes G. DOBBE, Sebastiaan A. BARTELS, E. CHRISTIAAN BOERMA, Paul W. G. ELBERS, Michal HEGER a Can INCE. *Rapid automatic assessment of microvascular density in sidestream dark field images*. 2011, **49**(11), 1269-278. DOI: 10.1007/s11517-011-0824-1. ISSN 0140-0118. Dostupné také z: http://link.springer.com/10.1007/s11517-011-0824-1

[5] LEAHY, Martin J. *Microcirculation imaging*. Weinheim: Wiley-Blackwell, c2012. ISBN 978-3-527-32894-9.

[6] MASSEY, Michael J. a Nathan I. SHAPIRO. A guide to human in vivo microcirculatory flow image analysis. *Critical Care*. 2016, **20**(1). DOI: 10.1186/s13054-016-1213-9. ISSN 1364-8535. Dostupné také z: http://ccforum.com/content/20/1/35

[7] *Septic Shock* [online]. 2019 [cit. 2019-04-20]. Dostupné z: https://emedicine.medscape.com/article/168402-overview#a3

[8] INCE, Can, E. Christiaan BOERMA, Maurizio CECCONI, et al. Second consensus on the assessment of sublingual microcirculation in critically ill patients: results from a task force of the European Society of Intensive Care Medicine. Intensive Care Medicine. 2018, 44(3), 281-299. DOI: 10.1007/s00134-018-5070-7. ISSN 0342-4642. Dostupné také z: http://link.springer.com/10.1007/s00134-018-5070-7

[9] DOBBE, J. G. G., G. J. STREEKSTRA, B. ATASEVER, R. VAN ZIJDERVELD a C. INCE. *Measurement of functional microcirculatory geometry and velocity distributions using automated image analysis*. 2008, **46**(7), 659-670. DOI: 10.1007/s11517-008-0349-4. ISSN 0140-0118. Dostupné také z: http://link.springer.com/10.1007/s11517-008-0349-4

[10] LEONDES, Cornelius T. *Biomechanical systems technology*. Hackensack, N.J.: World Scientific, c2007.

[11] CHAO LIU, P. T., Hernando GOMEZ, Srinivasa NARASIMHAN, Artur DUBRAWSKI, Michael R. PINSKY, Brian ZUCKERBRAUN, Iwan DOBBE a Can INCE. Real-time visual analysis of microvascular blood flow for critical care: a novel stroboscopic LED ring-based imaging modality for clinical assessment of the microcirculation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015, 2015, 15(23), 2217-2225. DOI: 10.1109/CVPR.2015.7298834. ISBN 978-1-4673-6964-0. ISSN 1094-4087. Dostupné také z: http://ieeexplore.ieee.org/document/7298834/

[12] DE BACKER, Daniel, Marc-Jacques DUBOIS, Denis SCHMARTZ, Marc KOCH, Anne DUCART, Luc BARVAIS a Jean-Louis VINCENT. Microcirculatory Alterations in Cardiac Surgery: Effects of Cardiopulmonary Bypass and Anesthesia. The Annals of Thoracic Surgery. 2009, 88(5), 1396-1403. DOI: 10.1016/j.athoracsur.2009.07.002. ISSN 00034975. Dostupné také z: https://linkinghub.elsevier.com/retrieve/pii/S0003497509014088

[13] BIENZ, Marc, David DRULLINSKY, Louis-Mathieu STEVENS, David BRACCO a Nicolas NOISEUX. Microcirculatory response during on-pump versus off-pump coronary artery bypass graft surgery. *Perfusion*. 2016, **31**(3), 207-215. DOI: 10.1177/0267659115590481. ISSN 0267-6591. Dostupné také z: http://journals.sagepub.com/doi/10.1177/0267659115590481

[14] DI DEDDA, Umberto, Marco RANUCCI, Alberto PORTA, Vlasta BARI, Alice ASCARI, Angela FANTINATO, Ekaterina BARYSHNIKOVA a Mauro COTZA. The combined effects of the microcirculatory status and cardiopulmonary bypass on platelet count and function during cardiac surgery. *Clinical Hemorheology and Microcirculation*. 2018, **70**(3), 327-337. DOI: 10.3233/CH-180391. ISSN 13860291.

[15] DE BACKER, Daniel, Jacques CRETEUR, Jean-Charles PREISER, Marc-Jacques DUBOIS, Jean-Louis VINCENT, Angela FANTINATO, Ekaterina BARYSHNIKOVA a Mauro COTZA. Microvascular Blood Flow Is Altered in Patients with Sepsis. American Journal of Respiratory and Critical Care Medicine. 2002, 166(1), 98-104. DOI: 10.1164/rccm.200109-016OC. ISSN 1073-449X. Dostupné také z: http://www.atsjournals.org/doi/abs/10.1164/rccm.200109-016OC

[16] SPRONK, Peter E, Can INCE, Martin J GARDIEN, Keshen R MATHURA, Heleen M Oudemans-van STRAATEN, Durk F ZANDSTRA, Ekaterina BARYSHNIKOVA a Mauro COTZA. Nitroglycerin in septic shock after intravascular volume resuscitation. The Lancet. 2002, 360(9343), 1395-1396. DOI: 10.1016/S0140-6736(02)11393-6. ISSN 01406736. Dostupné také z: https://linkinghub.elsevier.com/retrieve/pii/S0140673602113936

[17] TRZECIAK, Stephen, R. Phillip DELLINGER, Joseph E. PARRILLO, et al. Early microcirculatory perfusion derangements in patients with severe sepsis and septic shock: Relationship to hemodynamics, oxygen transport, and survival. *Annals of Emergency Medicine*. 2007, **49**(1), 88-98.e2. DOI: 10.1016/j.annemergmed.2006.08.021. ISSN 01960644. Dostupné také z: https://linkinghub.elsevier.com/retrieve/pii/S019606440602141X

[18] SAKR, Yasser, Marc-Jacques DUBOIS, Daniel DE BACKER, et al. Persistent microcirculatory alterations are associated with organ failure and death in patients with septic shock*: Relationship to hemodynamics, oxygen transport, and survival. Critical Care Medicine. 2004, 32(9), 1825-1831. DOI: 10.1097/01.CCM.0000138558.16257.3F. ISSN 0090-3493. Dostupné také z: https://insights.ovid.com/crossref?an=00003246-200409000-00002

[19] OURSELIN, Sebastien, Martin A. STYNER, Yansong LIU, Eli SABER, Angela GLADING a Maria HELGUERA. *Measurement of blood flow velocity for in vivo video sequences with motion estimation methods*. 2014-3-21, , 90342W-. DOI: 10.1117/12.2043255. Dostupné také z: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2043255

[20] DEMIR, Sumeyra U, Roya HAKIMZADEH, Rosalyn Hobson HARGRAVES, Kevin R WARD, Eric V MYER a Kayvan NAJARIAN. An automated method for analysis of microcirculation videos for accurate assessment of tissue perfusion. BMC Medical Imaging. IEEE, 2012, 2018, 12(1), 593-597. DOI: 10.1186/1471-2342-12-37. ISBN 978-9-0827-9701-5. ISSN 1471-2342. Dostupné také z: http://bmcmedimaging.biomedcentral.com/articles/10.1186/1471-2342-12-37

[21] ja72. Extracting rotation, scale values from 2d transformation matrix. In: *Mathematics* [online]. 2014 [cit. 2019-05-07]. Dostupné z: https://math.stackexchange.com/questions/13150/extracting-rotation-scale-values-from-2d-transformation-matrix/13165

[22] Rotation (mathematics). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2019 [cit. 2019-05-07]. Dostupné z: https://en.wikipedia.org/wiki/Rotation˙(mathematics)

[23] Pyramid (image processing). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2019 [cit. 2019-05-08]. Dostupné z: https://en.wikipedia.org/wiki/Pyramid˙(image˙processing)#cite˙note-LinBre03-ScSp-11

[24] ilastik: Interactive Learning and Segmentation Toolkit C. Sommer, C. Strähle, U. Köthe, F. A. Hamprecht in: Eighth IEEE International Symposium on Biomedical Imaging (ISBI). Proceedings, (2011), 230-233

[25] Ray Leroy Khuboni. What are the state-of -the art techniques for image segmentation? In: *Researchgate* [online]. 2016 [cit. 2019-05-07]. Dostupné z: https://www.researchgate.net/post/What˙are˙the˙state-of-the˙art˙techniques˙for˙image˙segmentation

[26] Otsu, N., ”A Threshold Selection Method from Gray-Level Histograms.” IEEE Transactions on Systems, Man, and Cybernetics. Vol. 9, No. 1, 1979, pp. 62–66.

[27] A. Frangi, W. Niessen, K. Vincken, and M. Viergever. Multiscale vessel enhancement filtering. In LNCS, volume 1496, pages 130–137, Germany, 1998. Springer-Verlag.

[28] VANTARAM, Sreenath Rao a Eli SABER. Survey of contemporary trends in color image segmentation. *Journal of Electronic Imaging*. 2012, **21**(4). DOI: 10.1117/1.JEI.21.4.040901. ISSN 1017-9909. Dostupné také z: http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.JEI.21.4.040901

[29] CHANG, Jen-Mei, Nen HUYNH, Marilyn VAZQUEZ a Carolyn SALAFIA. Vessel enhancement with multiscale and curvilinear filter matching for placenta images. *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2013, 2013, , 125-128. DOI: 10.1109/IWSSIP.2013.6623469. ISBN 978-1-4799-0944-5. Dostupné také z: http://ieeexplore.ieee.org/document/6623469/

[30] KHAN, Khan Bahadar, Amir. A. KHALIQ, Abdul JALIL, Muhammad SHAHID a Paul J ATZBERGER. A robust technique based on VLM and Frangi filter for retinal vessel extraction and denoising: 1371-1945 (1950) : inventář : B/103. PLOS ONE. IEEE, 2018, 1965, 13(2), 125-128. DOI: 10.1371/journal.pone.0192203. ISBN 978-1-4799-0944-5. ISSN 1932-6203. Dostupné také z: http://dx.plos.org/10.1371/journal.pone.0192203

[31] OURSELIN, Sébastien, Martin A. STYNER, Tim JERMAN, Franjo PERNUŠ, Boštjan LIKAR a Žiga ŠPICLIN. Beyond Frangi: an improved multiscale vesselness filter. *PLOS ONE*. IEEE, 2018, 2015-3-20, **13**(2), 94132A-. DOI: 10.1117/12.2081147. ISBN 978-1-4799-0944-5. ISSN 1932-6203. Dostupné také z: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2081147

[32] FU, Weilin, Katharina BREININGER, Roman SCHAFFERT, Nishant RAVIKUMAR, Tobias WÜRFL, Jim FUJIMOTO, Eric MOULT a Andreas MAIER. Frangi-Net: an improved multiscale vesselness filter. *Bildverarbeitung für die Medizin 2018*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, 2018-02-21, **13**(2), 341-346. Informatik aktuell. DOI: 10.1007/978-3-662-56537-7˙87. ISBN 978-3-662-56536-0. ISSN 1932-6203. Dostupné také z: http://link.springer.com/10.1007/978-3-662-56537-7˙87

[33] STARING, Marius, et al. Pulmonary Vessel Segmentation using Vessel Enhancement Filters. 2012.

[34] STAAL, J., M.D. ABRAMOFF, M. NIEMEIJER, M.A. VIERGEVER a B. VAN GINNEKEN. Ridge-Based Vessel Segmentation in Color Images of the Retina. *IEEE Transactions on Medical Imaging*. Division of Image Processing (LKEB), Department of Radiology, Leiden University Medical Center, Leiden, The Netherlands, 2004, **23**(4), 501-509. DOI: 10.1109/TMI.2004.825627. ISSN 0278-0062. Dostupné také z: http://ieeexplore.ieee.org/document/1282003/

[35] GONZALEZ, Germán, François AGUET, François FLEURET, Michael UNSER a Pascal FUA. Steerable Features for Statistical 3D Dendrite Detection. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, 2009, , 625-632. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-04271-3˙76. ISBN 978-3-642-04270-6. Dostupné také z: http://link.springer.com/10.1007/978-3-642-04271-3˙76

[36] STEGER, C. An unbiased detector of curvilinear structures. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(2), 113-125. DOI: 10.1109/34.659930. ISSN 01628828. Dostupné také z: http://ieeexplore.ieee.org/document/659930/

[37] SUN K., SANG N., ZHANG T. (2007) Marked Point Process for Vascular Tree Extraction on Angiogram. *Conference on Computer Vision and Pattern Recognition,* pp 467–478

[38] YEDIDYA T., HARTLEY R. (2008) Tracking of Blood Vessels in Retinal Images Using Kalman Filter. *Digital Image Computing: Techniques and Applications*, pp 52–58

[39] XU J.,WU J,. FENG D., CUI Z (2009) Dsa Image Blood Vessel Skeleton Extraction Based on Anti-Concentration Diffusion and Level Set Method. *Computational Intelligence and Intelligent Systems* 51:188–198

[40] VASILKOSKI Z., STEPANYANTS A. (2009) Detection of the Optimal Neuron Traces in Confocal Microscopy Images. *Journal of Neuroscience Methods* 178(1):197–204

[41] TÜRETKEN, Engin, Germán GONZŒLEZ, Christian BLUM a Pascal FUA. Automated Reconstruction of Dendritic and Axonal Trees by Global Optimization with Geometric Priors. Neuroinformatics. 2011, 9(2-3), 279-302. DOI: 10.1007/s12021-011-9122-1. ISSN 1539-2791. Dostupné také z: http://link.springer.com/10.1007/s12021-011-9122-1

[42] Schindelin, J.; Arganda-Carreras, I. & Frise, E. et al. (2012), "Fiji: an open-source platform for biological-image analysis", Nature methods 9(7): 676-682, PMID 22743772, doi:10.1038/nmeth.2019 (on Google Scholar).

[43] C. Sommer, C. Strähle, U. Köthe, F. A. Hamprecht. Ilastik: Interactive Learning and Segmentation Toolkit. in: Eighth IEEE International Symposium on Biomedical Imaging (ISBI). Proceedings, (2011), 230-233.

[44] The GIMP team, GIMP 2.8.10, www.gimp.org, 1997-2014, retrieved on 31.07.2014.

[45] KUMAR, Rajeev a Abhaya INDRAYAN. Receiver operating characteristic (ROC) curve for medical researchers. 2011.

[46] Computational and Mathematical Methods in Medicine. 2017, 2017. ISSN 1748-670X. Dostupné také z: https://www.hindawi.com/journals/cmmm/2017/3762651/

[47] DUDA, Richard a Peter HART. Use of the Hough Transformation To Detect Lines and Curves in Pictures. Communications of the ACM. 1972, (15).

[48] Hough Transform. /www.mathworks.com/ [online]. [cit. 2019-04-27]. Dostupné z: https://www.mathworks.com/help/images/ref/houghpeaks.html

[49] PRATT, William K. Digital image processing: PIKS inside. 3rd ed. New York: Wiley, c2001. ISBN 04-713-7407-5.

[50] JEFFREY WOODWARD, Hany Farid. Video Stabilization and Enhancement. Dartmouth College Hanover NH 03755, 2007.

[51] Lucas BD, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. In: Hayes PJ, editor. Proceeding of the 7th International Joint Conference on Artificial Intelligence (IJCAI); Vancouver, BC, Canada: William Kaufmann; 1981. pp. 674–679.

[52] HORN, Berthold a Brian SCHUNCK. Determining Optical Flow. ARTIFICIAL INTELLIGENCE. 1981(17), 185-203.

[53] DOBBE, J. G. G., G. J. STREEKSTRA, B. ATASEVER, R. VAN ZIJDERVELD a C. INCE. Measurement of functional microcirculatory geometry and velocity distributions using automated image analysis. 2008, 46(7), 659-670. DOI: 10.1007/s11517-008-0349-4. ISSN 0140-0118. Dostupné také z: http://link.springer.com/10.1007/s11517-008-0349-4

[54] WANG, Zhongqiang, Lei ZHANG a Hua HUANG. High-Quality Real-Time Video Stabilization Using Trajectory Smoothing and Mesh-Based Warping. IEEE Access. 2018, 6, 25157-25166. DOI: 10.1109/ACCESS.2018.2828653. ISSN 2169-3536. Dostupné také z: https://ieeexplore.ieee.org/document/8341815/

[55] IRANI, Zhongqiang, Lei ROUSSO a Hua PELEG. Recovery of ego-motion using image stabilization. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94. IEEE Comput. Soc. Press, 1994, 1994, 6, 454-460. DOI: 10.1109/CVPR.1994.323866. ISBN 0-8186-5825-8. ISSN 2169-3536. Dostupné také z: http://ieeexplore.ieee.org/document/323866/

[56] MORIMOTO, C., R. CHELLAPPA a Hua PELEG. Fast electronic digital image stabilization. Proceedings of 13th International Conference on Pattern Recognition. IEEE, 1996, 1996, 6, 284-288 vol.3. DOI: 10.1109/ICPR.1996.546956. ISBN 0-8186-7282-X. ISSN 2169-3536. Dostupné také z: http://ieeexplore.ieee.org/document/546956/

[57] GUILLUY, Wilko, Laurent OUDRE a Azeddine BEGHDADI. Feature Trajectories Selection for Video Stabilization. 2018 26th European Signal Processing Conference (EUSIPCO). IEEE, 2018, 2018, 6, 593-597. DOI: 10.23919/EUSIPCO.2018.8553453. ISBN 978-9-0827-9701-5. ISSN 2169-3536. Dostupné také z: https://ieeexplore.ieee.org/document/8553453/

[58] MicroVision Medical: The new USB3 MicroScan is here. [online]. 2014 [cit. 2019-05-05]. Dostupné z: https://www.microvisionmedical.com/

[59] Optical flow [online]. 2013 [cit. 2019-05-11]. Dostupné z: https://www.sciencedirect.com/topics/engineering/optical-flow

[60] Optical flow [online]. 2019 [cit. 2019-05-11]. Dostupné z: https://en.wikipedia.org/wiki/Optical˙flow

[61] BAGNI, Daniele, Kannan PARI a Stephen NEUENDORFFER. Demystifying the Lucas-Kanade Optical Flow Algorithm with Vivado HLS. XAPP1300 (v1.0) February 3, 2017. 2017.

[62] UZ, Zühre, Can INCE, Philippe GUERCI, et al. Recruitment of sublingual microcirculation using handheld incident dark field imaging as a routine measurement tool during the postoperative de-escalation phase—a pilot study in post ICU cardiac surgery patients. *Perioperative Medicine*. 2018, **7**(1). DOI: 10.1186/s13741-018-0091-x. ISSN 2047-0525. Dostupné také z: https://perioperativemedicinejournal.biomedcentral.com/articles/10.1186/s13741-018-0091-x

[63] CERNY, V., Z. TUZEK a R. PARIZKOVA. *Orthogonal Polarization Spectral Imaging*. University Hospital Hradec Králové, Department of Anesthesiology and Intensive Care, Czech Republic, 2006.

[64] GRONER, Warren, James W. WINKELMAN, Anthony G. HARRIS, Can INCE, Gerrit J. BOUMA, Konrad MESSMER a Richard G. NADEAU. Orthogonal polarization spectral imaging: A new method for study of the microcirculation. Nature Medicine. University Hospital Hradec Králové, Department of Anesthesiology and Intensive Care, Czech Republic, 1999, 5(10), 1209-1212. DOI: 10.1038/13529. ISSN 1078-8956. Dostupné také z: http://www.nature.com/articles/nm1099˙1209

[65] AYKUT, Guclu, Gerke VEENSTRA, Claudia SCORCELLA, Can INCE a Christiaan BOERMA. Cytocam-IDF (incident dark field illumination) imaging for bedside

monitoring of the microcirculation. Intensive Care Medicine Experimental. 2015, 3(1). DOI: 10.1186/s40635-015-0040-7. ISSN 2197-425X. Dostupné také z: http://www.icm-experimental.com/content/3/1/4

[66] LEONDES, Cornelius T. *Biomechanical systems technology*. Hackensack, N.J.: World Scientific, c2007.

[67] TUREK, Zdenek, Roman SYKORA, Martin MATEJOVIC a Vladimir CERNY. Anesthesia and the Microcirculation. Seminars in Cardiothoracic and Vascular Anesthesia. 2009, 13(4), 249-258. DOI: 10.1177/1089253209353134. ISSN 1089-2532. Dostupné také z: http://journals.sagepub.com/doi/10.1177/1089253209353134

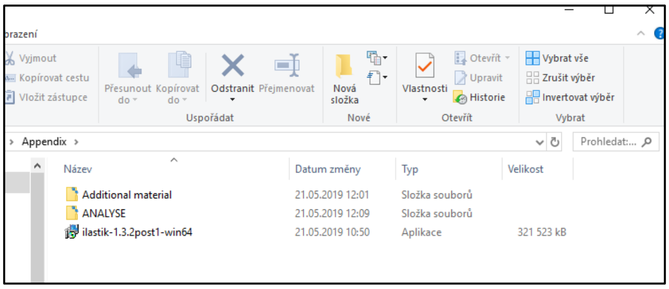
[68] MATLAB. Release R2018b, The MathWorks, Inc., Natick, Massachusetts, United States.

[69] R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Appendix

Description of „Appendix“ folder content (the folder is included on CD, which is attached to the printed thesis):

There are three subfolders and ilastik-1.3.2post1-win64.exe in Appendix folder:



Název	Datum změny	Typ	Velikost
Additional material	21.05.2019 12:01	Složka souborů	
ANALYSE	21.05.2019 12:09	Složka souborů	
ilastik-1.3.2post1-win64	21.05.2019 10:50	Aplikace	321 523 kB

subfolder Appendix-ANALYSE

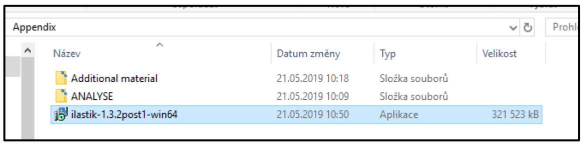
- It includes the software for automated analysis of microcirculation videos. Moreover, there are 4 exemplary microcirculation videos.

Instructions for automated analysis software

- The software was created in MatLab R2018b on Windows 10
- To run automatic analysis software, you need MatLab (with Image Processing Toolbox) and Ilastik 1.3.2 (tool for automatic image segmentation)

1) Install MatLab and Image Processing Toolbox
2) Install Ilastik:

- either from Appendix folder (ilastik-1.3.2post1-win64.exe):

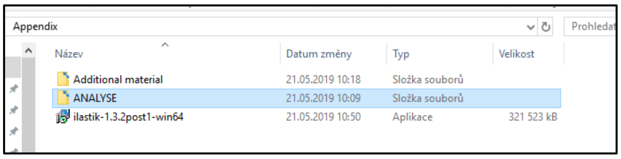


Název	Datum změny	Typ	Velikost
Additional material	21.05.2019 10:18	Složka souborů	
ANALYSE	21.05.2019 10:09	Složka souborů	
ilastik-1.3.2post1-win64	21.05.2019 10:50	Aplikace	321 523 kB

- or from website: https://www.ilastik.org/download.html

Run automatic analysis:

- Go to ANALYSE folder

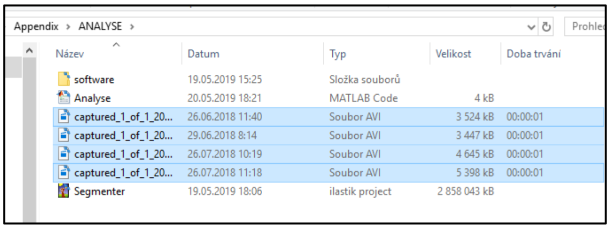


Appendix

Název	Datum změny	Typ	Velikost
Additional material	21.05.2019 10:18	Složka souborů	
ANALYSE	21.05.2019 10:09	Složka souborů	
ilastik-1.3.2post1-win64	21.05.2019 10:50	Aplikace	321 523 kB

- Put the videos you want to analyse into ANALYSE folder: (There are four exemplary videos right now). We use .avi videos, but another formats should be possible as well.

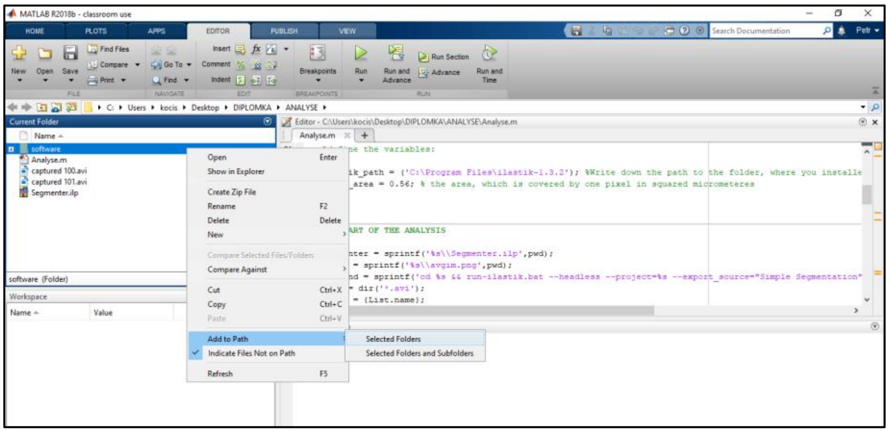
  Besides videos, there are: *Segmenter, software* and *Analyse*. The only thing you need to open is *Analyse*. Do not remove or manipulate *software* and *Segmenter*. *Software* includes Matlab functions, *segmenter* is a classifier for automated vessel segmentation.



Appendix › ANALYSE ›

Název	Datum	Typ	Velikost	Doba trvání
software	19.05.2019 15:25	Složka souborů		
Analyse	20.05.2019 18:21	MATLAB Code	4 kB	
captured_1_of_1_20...	26.06.2018 11:40	Soubor AVI	3 524 kB	00:00:01
captured_1_of_1_20...	29.06.2018 8:14	Soubor AVI	3 447 kB	00:00:01
captured_1_of_1_20...	26.07.2018 10:19	Soubor AVI	4 645 kB	00:00:01
captured_1_of_1_20...	26.07.2018 11:18	Soubor AVI	5 398 kB	00:00:01
Segmenter	19.05.2019 18:06	ilastik project	2 858 043 kB	

- Open Matlab script *Analyse.m* and add the folder *„software“* into Matlab path:



MATLAB R2018b - classroom use

C: › Users › kocis › Desktop › DIPLOMKA › ANALYSE ›

Editor - C:\Users\kocis\Desktop\DIPLOMKA\ANALYSE\Analyse.m

Current Folder: software, Analyse.m, captured 100.avi, captured 101.avi, Segmenter.ilp

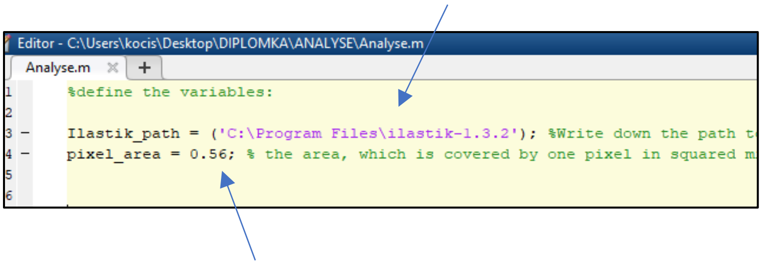
software (Folder)

```
ne the variables:

ik_path = ('C:\Program Files\ilastik-1.3.2'); %Write down the path to the folder, where you installe
_area = 0.56; % the area, which is covered by one pixel in squared micrometeres

ART OF THE ANALYSIS

nter = sprintf('%s\\Segmenter.ilp',pwd);
 = sprintf('%s\\avgim.png',pwd);
nd = sprintf('cd %s && run-ilastik.bat --headless --project=%s --export_source="Simple Segmentation"
= dir('*.avi');
 = {List.name};
```

You need to define two variables in *Analyse.m:*

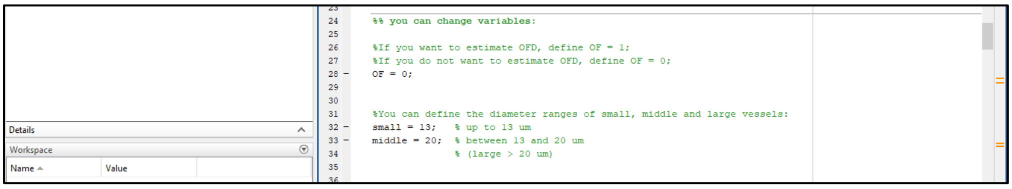
1) The path to the folder, where your Ilastik is installed (example: `'C:\Program Files\ilastik-1.3.2'`)

2) The real area, which is covered by one pixel of the video (in sqaured micrometres - µm^2) The preset value 0.56 is for MicroScan USB 3.



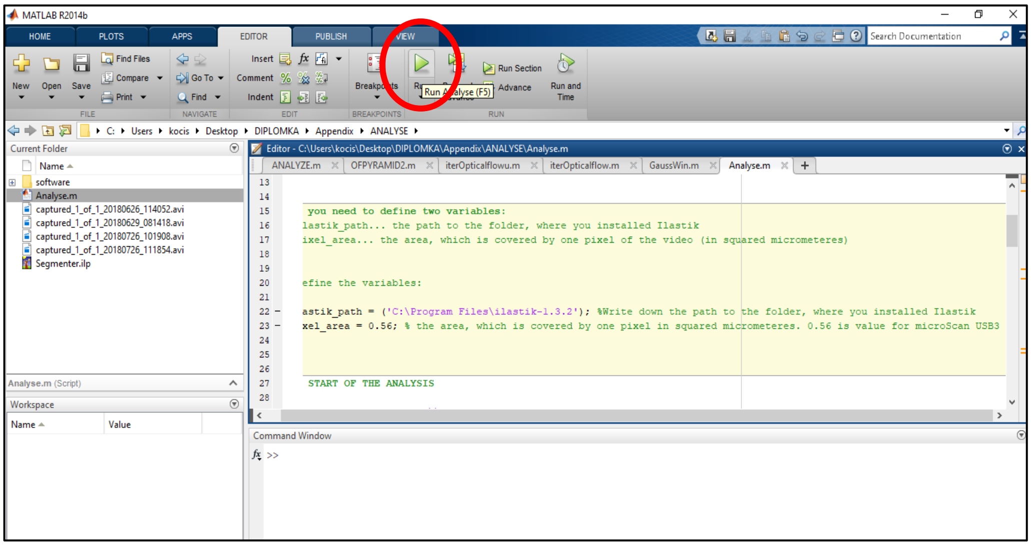
```
%define the variables:

Ilastik_path = ('C:\Program Files\ilastik-1.3.2'); %Write down the path t
pixel_area = 0.56; % the area, which is covered by one pixel in squared m
```

- You can choose, if you want to include OFD descriptor - define OF = 1 (yes), or OF = 0 (no)
- You can define the range of diameters for small, middle and large vessels (in micrometers)



```
%% you can change variables:

%If you want to estimate OFD, define OF = 1;
%If you do not want to estimate OFD, define OF = 0;
OF = 0;


%You can define the diameter ranges of small, middle and large vessels:
small = 13;   % up to 13 um
middle = 20;  % between 13 and 20 um
              % (large > 20 um)
```

Now you are ready to run Analyse.m and let the software work:



Editor - C:\Users\kocis\Desktop\DIPLOMKA\Appendix\ANALYSE\Analyse.m

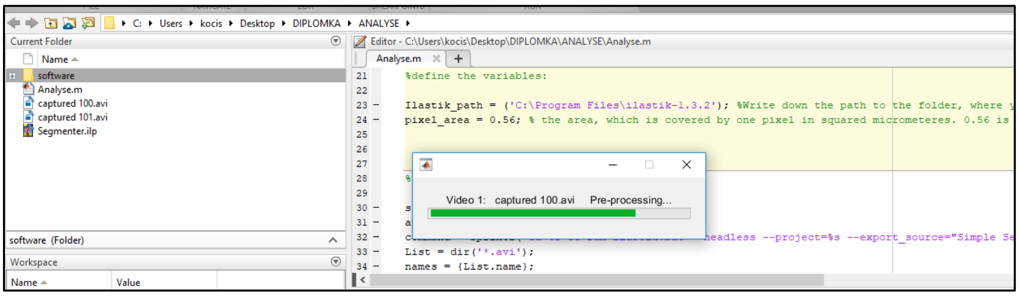
```
 you need to define two variables:
lastik_path... the path to the folder, where you installed Ilastik
ixel_area... the area, which is covered by one pixel of the video (in squared micrometeres)


efine the variables:

astik_path = ('C:\Program Files\ilastik-1.3.2'); %Write down the path to the folder, where you installed Ilastik
xel_area = 0.56; % the area, which is covered by one pixel in squared micrometeres. 0.56 is value for microScan USB3



 START OF THE ANALYSIS
```

What will happen:

- Each video will be processed one by one. It takes cca. 15 minutes to analyse 1 video, which is 1 second long (resolution 960x1280).

- There are 4 stages in the analysing process for each video:

    1) Pre-processing. This stage lasts approximately 30 seconds.



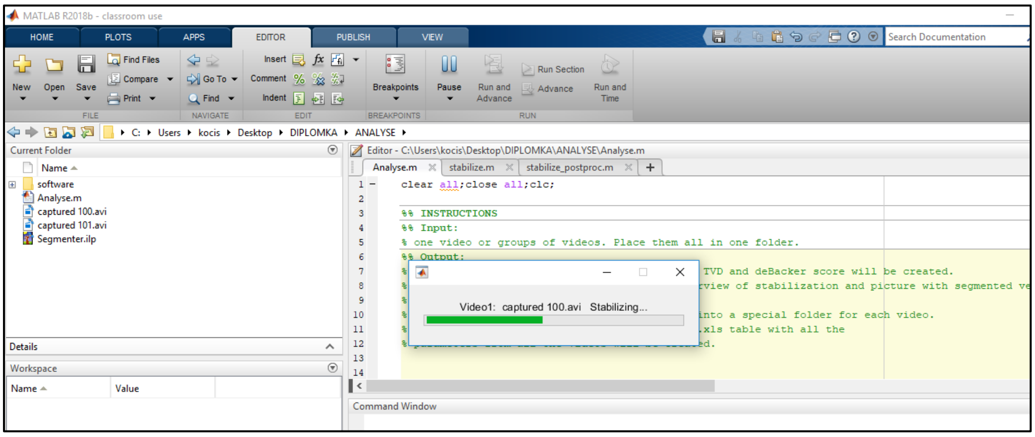
Editor - C:\Users\kocis\Desktop\DIPLOMKA\ANALYSE\Analyse.m

```
%define the variables:

Ilastik_path = ('C:\Program Files\ilastik-1.3.2'); %Write down the path to the folder, where y
pixel_area = 0.56; % the area, which is covered by one pixel in squared micrometeres. 0.56 is
```

Video 1: captured 100.avi Pre-processing...

```
List = dir('*.avi');
names = {List.name};
```

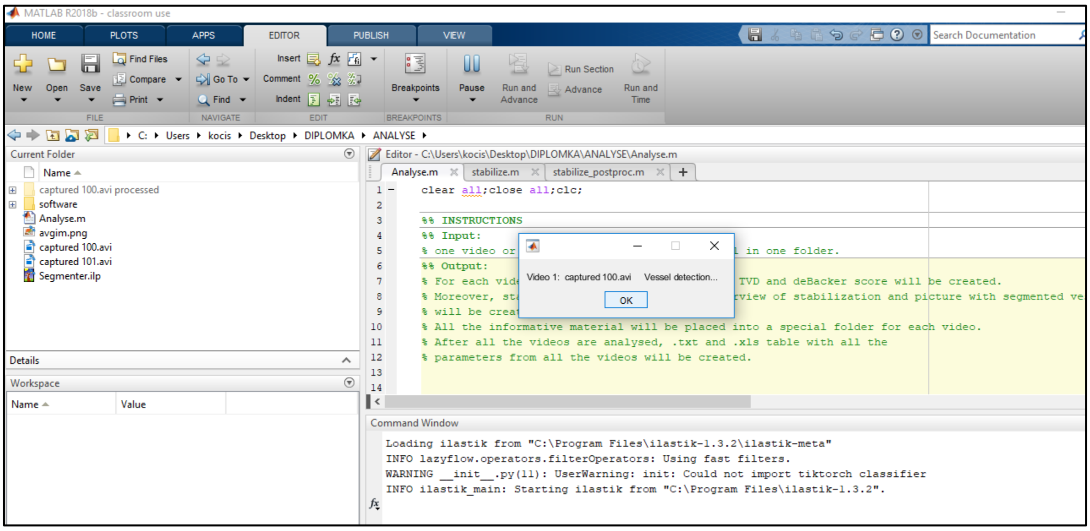
2) Stabilizing. This stage lasts approximately 5 minutes.



```
clear all;close all;clc;

%% INSTRUCTIONS
%% Input:
% one video or groups of videos. Place them all in one folder.
%% Output:
%                                    TVD and deBacker score will be created.
%                                   rview of stabilization and picture with segmented ve
%
%                                   into a special folder for each video.
%                                   .xls table with all the
%                                   ed.
```

Video1: captured 100.avi   Stabilizing...

3) Vessel detection. This stage lasts approximately 8 minutes.

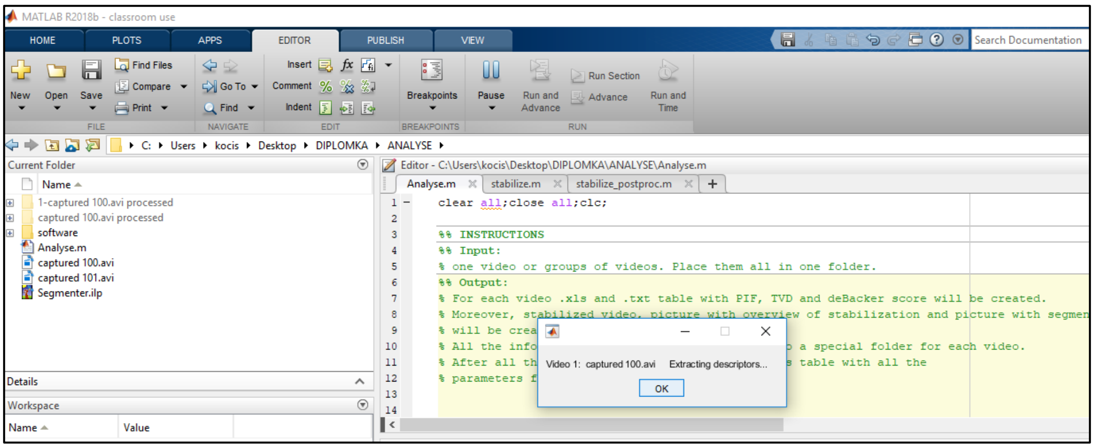


```
clear all;close all;clc;

%% INSTRUCTIONS
%% Input:
% one video or                       l in one folder.
%% Output:
% For each vide                      TVD and deBacker score will be created.
% Moreover, sta                      rview of stabilization and picture with segmented ve
% will be creat
% All the informative material will be placed into a special folder for each video.
% After all the videos are analysed, .txt and .xls table with all the
% parameters from all the videos will be created.
```

Video 1: captured 100.avi   Vessel detection...

```
Loading ilastik from "C:\Program Files\ilastik-1.3.2\ilastik-meta"
INFO lazyflow.operators.filterOperators: Using fast filters.
WARNING __init__.py(11): UserWarning: init: Could not import tiktorch classifier
INFO ilastik_main: Starting ilastik from "C:\Program Files\ilastik-1.3.2".
```

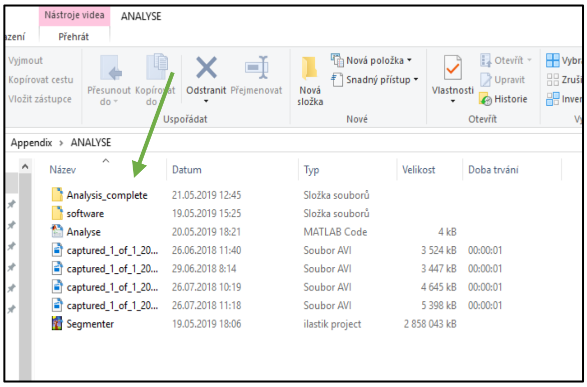
4) Extraction of microcirculation descriptors. This stage lasts approximately 2 minutes.



```
clear all;close all;clc;

%% INSTRUCTIONS
%% Input:
% one video or groups of videos. Place them all in one folder.
%% Output:
% For each video .xls and .txt table with PIF, TVD and deBacker score will be created.
% Moreover, stabilized video, picture with overview of stabilization and picture with segmen
% will be crea                                                    
% All the info                                        o a special folder for each video.
% After all th                                        s table with all the
% parameters f
```

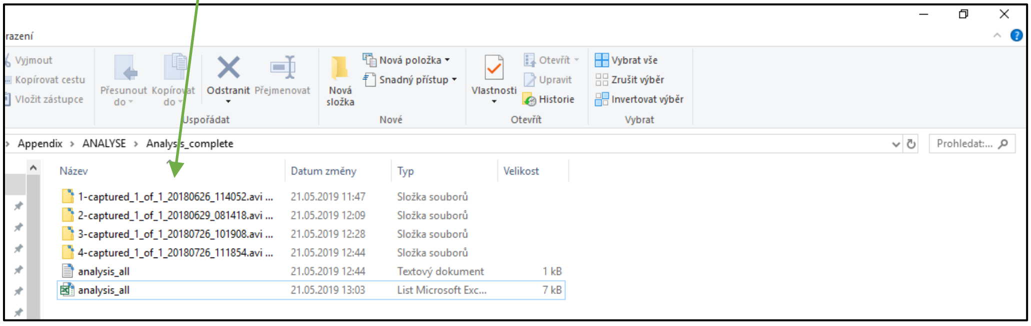
Video 1: captured 100.avi   Extracting descriptors...

After all the videos are processed, folder **Analysis complete** is created automatically



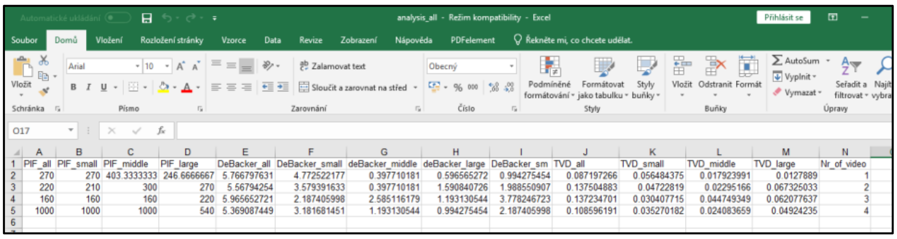
Název	Datum	Typ	Velikost	Doba trvání
Analysis_complete	21.05.2019 12:45	Složka souborů		
software	19.05.2019 15:25	Složka souborů		
Analyse	20.05.2019 18:21	MATLAB Code	4 kB	
captured_1_of_1_20...	26.06.2018 11:40	Soubor AVI	3 524 kB	00:00:01
captured_1_of_1_20...	29.06.2018 8:14	Soubor AVI	3 447 kB	00:00:01
captured_1_of_1_20...	26.07.2018 10:19	Soubor AVI	4 645 kB	00:00:01
captured_1_of_1_20...	26.07.2018 11:18	Soubor AVI	5 398 kB	00:00:01
Segmenter	19.05.2019 18:06	ilastik project	2 858 043 kB	

It contains a *folder* for each analysed video. There is information about analysis in each of these folders: stabilized video, picture with overview of stabilization, picture with segmented vessels and descriptors in .txt and .xls.



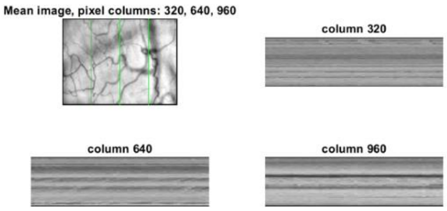
Appendix › ANALYSE › Analysis_complete

Název	Datum změny	Typ	Velikost
1-captured_1_of_1_20180626_114052.avi ...	21.05.2019 11:47	Složka souborů	
2-captured_1_of_1_20180629_081418.avi ...	21.05.2019 12:09	Složka souborů	
3-captured_1_of_1_20180726_101908.avi ...	21.05.2019 12:28	Složka souborů	
4-captured_1_of_1_20180726_111854.avi ...	21.05.2019 12:44	Složka souborů	
analysis_all	21.05.2019 12:44	Textový dokument	1 kB
analysis_all	21.05.2019 13:03	List Microsoft Exc...	7 kB

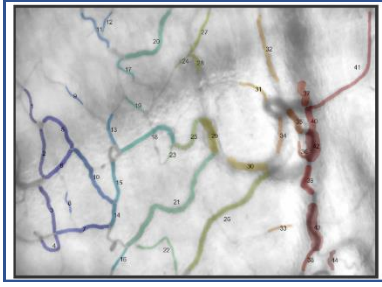


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PIF_all	PIF_small	PIF_middle	PIF_large	DeBacker_all	DeBacker_small	deBacker_middle	deBacker_large	DeBacker_sm	TVD_all	TVD_small	TVD_middle	TVD_large	Nr_of_video
2	270	270	403.3333333	246.6666667	5.766797631	4.772522177	0.397710181	0.596565272	0.994275454	0.087197266	0.056484375	0.017923991	0.0127889	1
3	220	210	300	270	5.56794254	3.579391633	0.397710181	1.590840726	1.988550907	0.137504883	0.04722819	0.02295166	0.067325033	2
4	160	160	160	220	5.965652721	2.187405998	2.585116179	1.193130544	3.778246723	0.137234701	0.030407715	0.044749349	0.062077637	3
5	1000	1000	1000	540	5.369087449	3.181681451	1.193130544	0.994275454	2.187405998	0.108596191	0.035270182	0.024083659	0.04924235	4
6														

descriptors

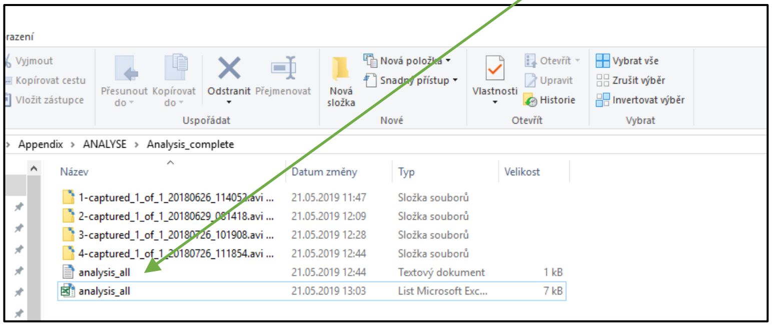


stabilization overview



segmentation overview

Moreover, folder **Analysis complete** contatins the *table* with all the descriptors from all the analysed videos both in .txt and .xls



Appendix › ANALYSE › Analysis_complete

Název	Datum změny	Typ	Velikost
1-captured_1_of_1_20180626_114052.avi ...	21.05.2019 11:47	Složka souborů	
2-captured_1_of_1_20180629_081418.avi ...	21.05.2019 12:09	Složka souborů	
3-captured_1_of_1_20180726_101908.avi ...	21.05.2019 12:28	Složka souborů	
4-captured_1_of_1_20180726_111854.avi ...	21.05.2019 12:44	Složka souborů	
analysis_all	21.05.2019 12:44	Textový dokument	1 kB
analysis_all	21.05.2019 13:03	List Microsoft Exc...	7 kB

subfolder Appendix-Additional material

- **Data for classification**: It is a dataset, which was gained from 151 videos. It was used to train linear SVM classifier to explore the differences between on-pump and off-pump videos

- **SVM**: Folder with R script (Classify.R), which contains linear SVM classifiers. It is pre-set for the .xls table which is produced by autmatic analysis. When you run it, the videos will be classified according to PIF and deBacker score. The table with predictions will be created. There are more detailed instructions in the folder.

- **Videos**: Those videos, from which data for classification were obtained.