VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

MULTIMODÁLNÍ REGISTRACE RETINÁLNÍCH SNÍMKŮ Z FUNDUS KAMERY A OCT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ONDŘEJ BĚŤÁK

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF BIOMEDICAL ENGINEERING

MULTIMODÁLNÍ REGISTRACE RETINÁLNÍCH SNÍMKŮ Z FUNDUS KAMERY A OCT

MULTIMODAL REGISTRATION OF FUNDUS CAMERA AND OCT RETINAL IMAGES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE AUTHOR	Bc. ONDŘEJ BĚŤÁK
VEDOUCÍ PRÁCE SUPERVISOR	Ing. JIŘÍ GAZÁREK

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské a ekologické inženýrství

Student:	Bc. Ondřej Běťák	***ID:***	70279
Ročník:	2	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Multimodální registrace retinálních snímků z fundus kamery a OCT

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s dostupnými prameny o zpracování barevných retinálních snímků, zejména snímků pořízených digitální fundus kamerou a OCT. Z dostupné literatury nastudujte současné metody registrace obrazů a podle pokynů vedoucího projektu vybranou metodu realizujte v programovém prostředí MATLAB. Metodu otestujte na dostupných obrazových datech z fundus kamery a OCT na ÚBMI.

DOPORUČENÁ LITERATURA:

[1] JAN J.: Medical Image Processing, Reconstruction and Restoration - Concepts and Methods, CRC New York, 2006.
[2] ZITOVÁ B., FLUSSER J.: Image registration methods: a survey, Image and Vision Computing, 2003.

Termín zadání:	15.10.2010	***Termín odevzdání:***	20.5.2011

Vedoucí práce: Ing. Jiří Gazárek

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Ondřej Běťák
Bytem: Střítež nad Bečvou 40, 756 52
Narozen/a (datum a místo): 13.2.1987 ve Valašském Meziříčí

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Technická 10, Brno 612 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Ivo Provazník, Ph.D, předseda rady oboru Biomedicínské a ekologické inženýrství
(dále jen „nabyvatel“)

Čl. 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

   ☐ disertační práce
   ☒ diplomová práce
   ☐ bakalářská práce
   ☐ jiná práce, jejíž druh je specifikován jako ....
   (dále jen VŠKP nebo dílo)

   Název VŠKP: Multimodální registrace retinálních snímků z fundus kamery a OCT
   Vedoucí/ školitel VŠKP: Ing. Jiří Gazárek
   Ústav: Ústav biomedicínského inženýrství
   Datum obhajoby VŠKP: 7. nebo 8. června 2011 *

   VŠKP odevzdal autor nabyvateli [*]:

   ☒ v tištěné formě – počet exemplářů: 2
   ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

[*] hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.

2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.

3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti

   ☒ ihned po uzavření této smlouvy
   □ 1 rok po uzavření této smlouvy
   □ 3 roky po uzavření této smlouvy
   □ 5 let po uzavření této smlouvy
   □ 10 let po uzavření této smlouvy
   (z důvodu utajení v něm obsažených informací)

4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.

2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.

3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.

4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 3. května 2011

………………………………………..   ………………………………………
Nabyvatel   Autor

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Multimodální registrace retinálních snímků z fundus kamery a OCT jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 3. května 2011 ....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Jiřímu Gazárkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 3. května 2011 ....
podpis autora

INTRODUCTION

This thesis is dealing with multimodal registration of retinal devices from different scanning devices. With this multimodal registration is possible to enhance some details on the images which are crucial to detect different types of eye diseases (such as glaucoma, nerve fibre layer degradation, vessel degradation, etc.). The fundus camera is very common in ophthalmology and the fundus devices are cheaper comparing to OCT or SLO devices. That is why fundus is very important even if it is not the best method to detect retinal diseases. On the other hand, the OCT is very sophisticated system which often contains also the SLO device, and is very helpful with finding retinal diseases, but also more expensive. The chapter one is focused on theoretical background of retina itself, retinal imaging, image processing, and image registration. The second chapter is focused on practical application of this knowledge in Matlab. The first step was to create manual registration of fundus images and SLO images. The registration of these two images from fundus and SLO is quite common. Another step was to create automatic registration of fundus and SLO images. For this registration it was necessary to use algorithm which can evaluate the similarities in both images, thus the correlation was used. Thanks to automatic and manual registration was possible to create hybrid type of registration – semi-automatic registration. In author's opinion this type of registration is the most effective one. It is very accurate and relatively fast with large amount of images. The final task was to create algorithm which can match the OCT B-scans with the lines in fundus image. The registration of fundus and OCT is not very usual. The results are consulted in the conclusion.

CONTENT

1 BACKGROUND INFORMATION

1.1 Biological background

1.1.1 Vision system

Vision system includes *bulbus oculi* and its auxiliary structures *organa oculi accessoria*. [4] Shape of *Bulbus oculi* approximately corresponds to a sphere. Its diameter is about 25 mm in adult human. The poles of the eye surface are connected by lines, known as meridians. In the direction of progress of the eyeball are vessels. *Linea visus* is an axis which connects observed point with the middle of *fovea centralis retinae* (yellow spot).

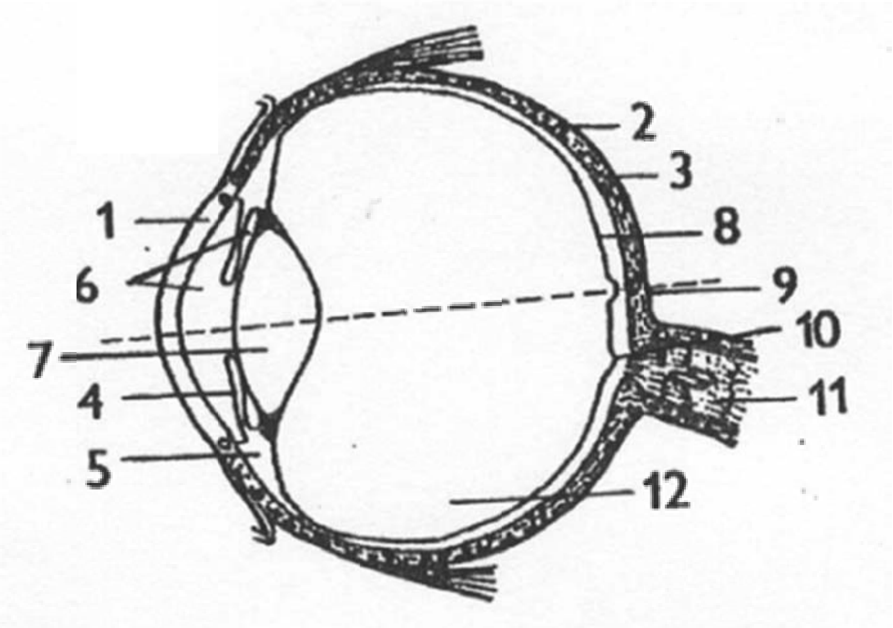

Fig. 1.1 – 1. cornea, 2. sclera, 3. chorioid (chorioidea), 4. iris, 5. ciliary apparatus (corpus ciliare), 6. front and rear eye chamber, 7. lens, 8. retinae (retina), 9. yellow spot (fovea centralis), 10. blind spot (loco caecus), 11. optic nerve (nervus opticus), 12. vitreous (corpus vitreum)

The eye movements are provided by extraocular muscles. Eyelids, conjunctiva and lacrimal apparatus protect the eye from mechanical damage, provide eye moistening and prevent inflammation. Lacrimal gland *(Glandula lacrimalis)* is placed externally under the eyepiece cap. Tears flow out by the lacrimal cochlea to the nasal cavity.

Rods and cones represent photosensitive elements of the retina. Yellow spot *(Fovea centralis)* is the point of sharpest vision and there are cumulated rods and cones. Cones are for daylight vision (cone vision) and rods are for monochromatic vision in the dusk or night (scotopic vision). The eye is able to recognize the visible part of spectrum which is in between 400 – 700 nm. There exist three different types of cones in the eye, which are sensitive to variable wave length. The range of these wave lengths can be divided into colours. Cones are sensitive to red, green and blue colour. Different intensity of stimulation of all cones causes the sensing of the whole colour spectrum.

1.2.3 Anatomy of retina

Retina is a light-sensitive tissue lining the inner surface of the eye. It is formed as pouch of diencephalon (the rear of the forebrain). Main function of retina is to capture the light signals incident on it through the lens. The rods and cones are placed on the inner side of retina. It also includes the yellow and blind spot (Fig. 1.2).

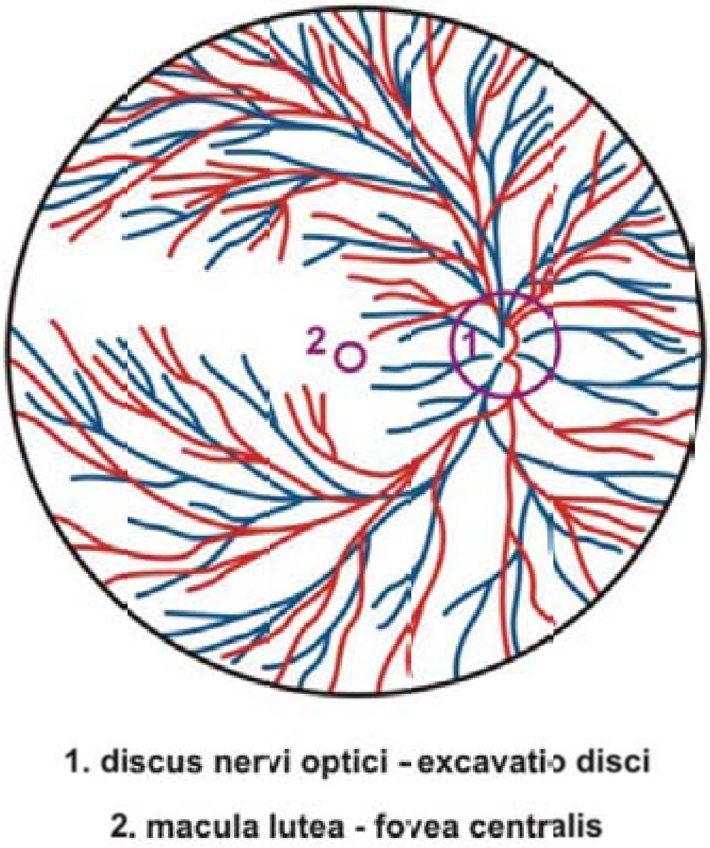
1
2
1. discus nervi optici - excavatio disci
2. macula lutea - fovea centralis

Fig. 1.2 – Cross section of the eye showing retinal vasculature: 1. marking vessels and exit of the optic nerve (blind spot), 2. yellow spot [4]

The retina can be further divided to these layers:

1. *stratum pigmenty retinae – pigmentosum* (Pigmentary epithelium of retina)
2. *stratum neuroepitheliale* (layer of rods and cons)
3. *stratum ganglionare retinae* (layer of bipolar neurons)
4. *stratum ganglionare nervi optici* (layer of multipolar neurons, theirs axons create *nervus opticus* (optic nerve))

[2] The retina covers the back of the eye making optical magnification constant of 3.5 degrees of scan angle of the eye per one millimetre. It allows image pre-processing of the captured image in the form of edge detection and colour analysis before the information is transferred along the optic nerve to the brain. Retina contains four main types of cells which are arranged in layers as described above. The dendrites of these cells occupy no more than 1 to 2

mm^2 of retina and their content is limited by spatial distribution of the layers. In the first layer are located about 125 million receptors. They contain the photosensitive pigment responsible for converting photons into chemical energy. Receptor cells, as already mentioned, are divided into two groups: rods and cones. Cones are responsible for colour vision and work only in daylight or in good light of the scanned image. On the other hand, in the dark the rods are exited. When the single photon is captured by the eye, it increases the membrane potential, which can be measured in the rods. This reaction of the eye (eye sensitivity) is the result of chemical cascade which works similar like a photomultiplier in which the single photon generates the cascade of electrons. All rods use the same pigment whereas cones use three different types of pigments.

On the inner side of retina is a layer of the ganglion cells (*stratum ganglionare retinae*), those axons form the optic nerve – output of retina. There are approximately one million of these cells. When the information is transferred through the area between the receptors and ganglion cells, there is a compression of the transferred information. Bipolar cells represent the first level of information processing in the visual system. Throughout this area are bipolar cells dissociated from receptors to the ganglion cells. Their response to the light represents the centre, eventually surroundings, of captured image (scene). It means that even a little dot on the retina evokes specific reaction, whereas surrounding area evokes opposite reaction. If the centre and the surrounding are illuminated at the same time, there is no response. Thus the bipolar cells exist in two types: on-centre and off-centre. The on-centre reacts to the bright illumination and off-centre to the dark.

On-centre response of bipolar cells is in direct contact with receptors. Off-centre response is delivered by horizontal cells which are parallel to the retina surface and can be founded between receptor and bipolar layer. Thanks to this, the off-centre layer has the opposite influence to the on-centre cells. Amacrine cells are also parallel to the retina surface. They can be founded in the different layer, between the bipolar cells and ganglion cells. Amacrine cells detect the motions on the captured image.

If the ganglion cells are triggered by bipolar cells, then they have also the on-centre and off-centre receptor field. In the on-centre of ganglion cells is field which illumination causes increasing the excitation of these cells. On the other hand, if the off-centre field is illuminated, it causes the decreasing of excitation of these cells. If the whole area is illuminated, then there is a little or none level of excitation, because mutual effect annuls the excitation. The fibres of optical nerve use the frequency encoding for scalar amount representation. Several of ganglion cells can receive information from the same receptor, because receptor fields of these cells overlap themselves. The maximal resolution is in the yellow spot. For recognizing two different points in the captured scene it is necessary that the distance between these two points has to be at least 0.5 minute of the visual angle. This separation corresponds to a distance on the retina of 2.5 µm, which is approximately the center-to-center spacing between cones.

1.2 Devices for retinal scanning

1.2.1 Fundus camera (Retinal Camera)

Fundus camera is a specialized microscope with an attached camera designed to photograph the interior surface of the eye – retina, blind and yellow spot. These photographs allow medical professionals to monitor progression of a disease – glaucoma, macular degeneration, etc. The design is based on principle of monocular indirect ophthalmoscopy. It replaced previously used ophthalmoscope [7].

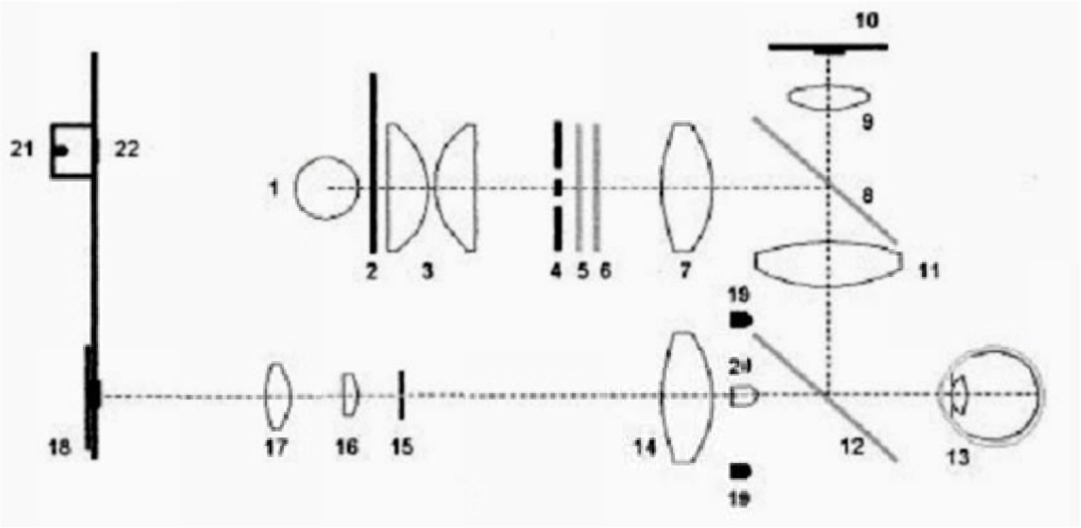

Fig. 1.3 – Principle of fundus camera; 1 – halogen lamp, 2 – shade, 3 – optics for focusing, 4 – aperture for illumination, 5 – green filter, 6 – red filter, 10 – (CCD) camera, 18 – CCD sensor, 19 – aperture for reflection measurements, 17 – optics for focusing an image, 20 – diode for aiming the eye [7]

The principle of scanning by fundus camera is implied on the figure 1.3. The procedure for acquiring the image starts with fixation of patient’s head to the special holder, which fixates the position of the eye. Then the device automatically or manually detects the pupil centre. In the next step the device automatically focuses to the retina – the shade opens (Fig. 1.3 – 2), several images is scanned by CCD sensor (Fig. 1.3 - 18) with synchronous shift of the lens (Fig. 1.3 - 17). Further the system evaluates the amount of high frequencies in scanned image. Well – focused image has many high frequencies. In the next step the device automatically sets the intensity of the patient’s eye illumination. The images scanned in previous step are evaluated and the average intensity of the image is calculated. The final image is generated by scanning of several images for different shifting around the position of the best focusing. It is also possible to scan images in horizontal shifting.

For better vessel resolution the scanning through green filter is used. The illumination is usually around 1 mW/cm^2. Total exposition period lasts approximately 250 ms. Field of view (FOV) is most often between 20° to 45°. The device is capable to compensate dioptric deviations of the patient’s eye in the range between +/- 20 D. For focusing the pupil, it’s diameter of at least 4 mm is needed. The resolution of CCD sensor is usually larger than 2.1 MPx. The overall magnification in the output image compared to the patient's eye is ten to thirty times. The output image is usually stored as a RGB image in standard JPEG or in a special format of manufacturer.

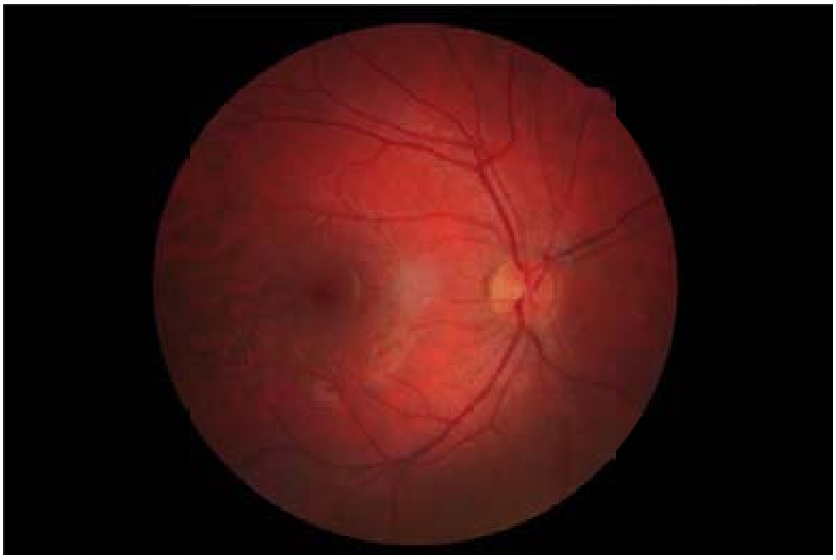

Fig. 1.4 – Fundus camera image

1.2.2 OCT (Optical Coherence Tomography) – time domain

OCT is non-invasive tomographic imaging and diagnostic method, which scanning the images of biological tissue (skin, mucosa, eye and teeth) in high resolution. [5] It belongs to a group of optical tomographic techniques. OCT scans the images in the transversal (cross) section. The method uses infrared radiation, which is able to penetrate to a depth of 1-3 mm and it has a high resolution. Based on the characteristics of the infrared radiation source, it may have resolution in microns. With this radiation, the computer can reconstruct the physical structures in two or three dimensions.

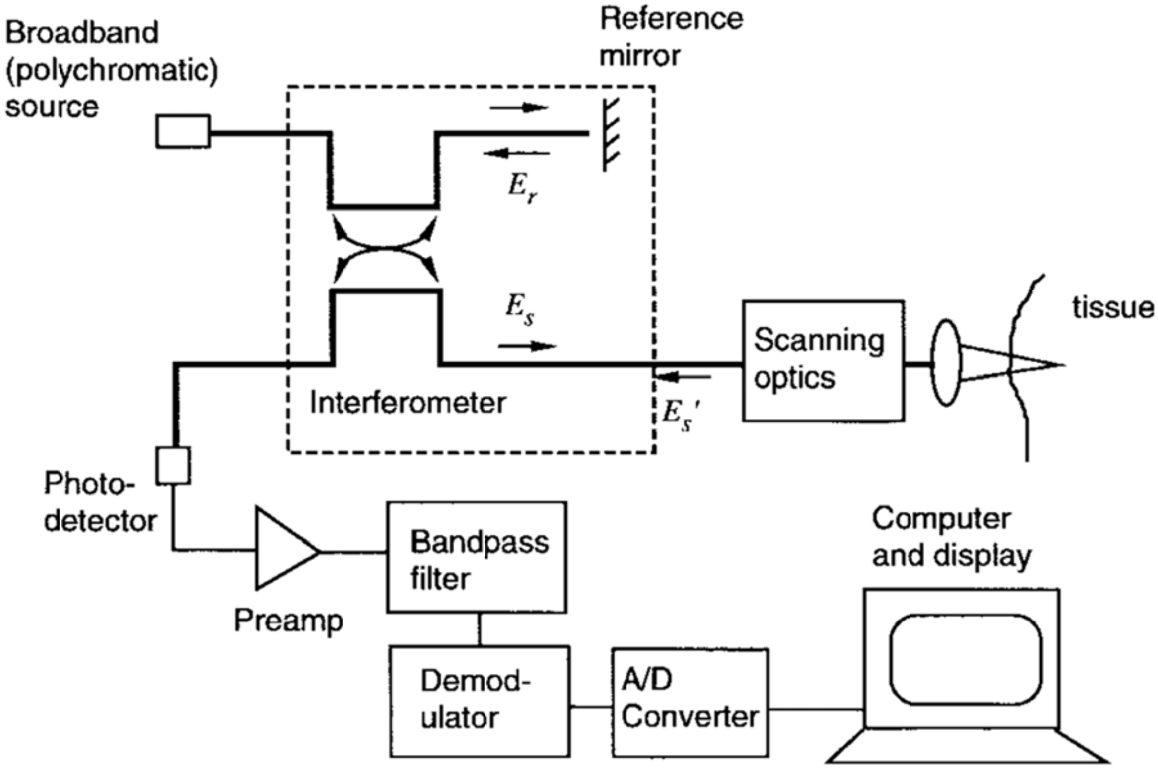

Fig. 1.5 – Component blocks of an OCT system [5]

Figure 1.5 shows the basic components of the OCT system. One of the most important blocks is the interferometer, which is illuminated by broadband light source. In this section, the interferometer is stripped of all unnecessary signals for further analysis. Interferometer divides the broadband source field to reference field $\boldsymbol{E_r}$ and sample field $\boldsymbol{E_s}$. [5] The sample field focuses through the scanning optics and objective lens to some point below the surface of the tissue. After scattering back from the issue, the modified sample field ***E'*** mixes with $\boldsymbol{E_r}$ on the surface of the photodetector. Given the assumption that the photodetector captures all light from reference and sample arms the intensity that impacts the photodetector is:

$$I_d = \left\langle \left| E_d \right|^2 \right\rangle = 0{,}5(I_r + I'_s) + \mathrm{Re}\left\{ \left\langle E*_r\,(t+r)E'_s\,(t) \right\rangle\ \right\} \tag{1}$$

“where I_r a I_s are mean intensities returning from the reference and sample arms of the interferometer. I_d is intensity impacts the photodetector. The second term in equation (1), which depends on the optical time delay τ, set by the position of the reference mirror, represents the amplitude of the interference fringes that carry information about the tissue structure. The nature of the interference fringes - or whether any fringes form at all - depends on the degree to which the temporal and spatial characteristics of $\boldsymbol{E'_s}$ and $\boldsymbol{E_r}$ match. Under the assumption that the tissue behaves as an ideal mirror that leaves the sample beam unaltered, the correlation amplitude depends on the temporal-coherence characteristics:

$$\mathrm{Re}\left[\left\langle E*_s\,(t+r)E'_s\,(t)\right\rangle\right] = \left|G(\tau)\right|\cos\left[2\pi\nu_0\tau + \Phi(\tau)\right] \tag{2}$$

where c is the speed of light, $\nu_0 = \dfrac{c}{\lambda_0}$ is middle frequency of the source, *G(τ)* is complex temporal-coherence function with argument Φ(τ). According to Wiener-Khinchin theorem, *G(τ)* is related to the power spectral density of the source , *S(ν)*, as

$$G(\tau) = \int_0^{\infty} S(\nu)e^{-j2\pi\tau}d\nu \tag{3}$$

It follows that the shape and the width of the emission spectrum of the light source are important variables in OCT because of their influence on the sensitivity of the interferometer. The relationship between *S(ν)* and *G(τ)* is:

$$S(\nu)=\frac{2\sqrt{\ln\frac{2}{\pi}}}{\Delta\nu}e^{-4\ln 2(\frac{\nu-\nu_0}{\Delta\nu})} \tag{4}$$

$$G(\tau)=e^{-\left(\frac{\pi\Delta\nu\tau}{2\sqrt{\ln 2}}\right)^2}e^{-j2\pi\nu_0\tau} \tag{5}$$

In these equations, the half-power bandwidth Δν represents the spectral width of the source in optical frequency domain. The corresponding measure of correlation width, derived from (5), is the correlation length, given by:

$$l_c=\frac{2c\ln(2)}{\pi}\frac{1}{\Delta\nu}\approx 0{,}44\frac{\lambda_0^2}{\Delta\lambda} \tag{6}$$

where Δλ is the full-width of the coherence function at half-maximum measured in wavelength units.” [5]

The principle which is described above, describes so-called Time domain OCT. The newest OCT devices functions in the spectral domain.

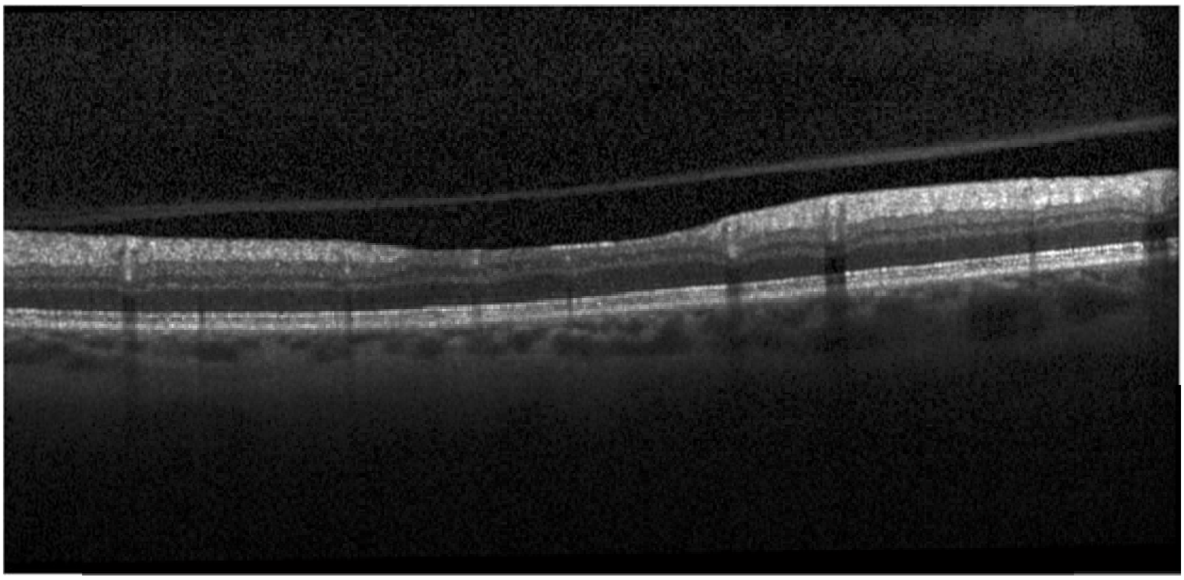

Fig. 1.6 – OCT image (B-scan)

1.2.3 OCT (Optical Coherence Tomography) – spectral domain

OCT enables micron scale, cross-sectional imaging of internal issue microstructure in real time. [13] Commercial ophthalmic OCT imaging systems have axial resolutions of 10 – 15 µm. Spectral (Fourier) domain detection methods for OCT were suggested in 90s of 20th century. In approximately 2005 became an active area of investigation. Studies have shown that spectral domain detection can enable dramatic improvements in imaging speed or detection sensitivity compared to standard detection techniques. Spectral time domain OCT detects the echo time delay and magnitude of light by measuring the interference spectrum of the light signal from the tissue. The principle is shown in figure 1.7.

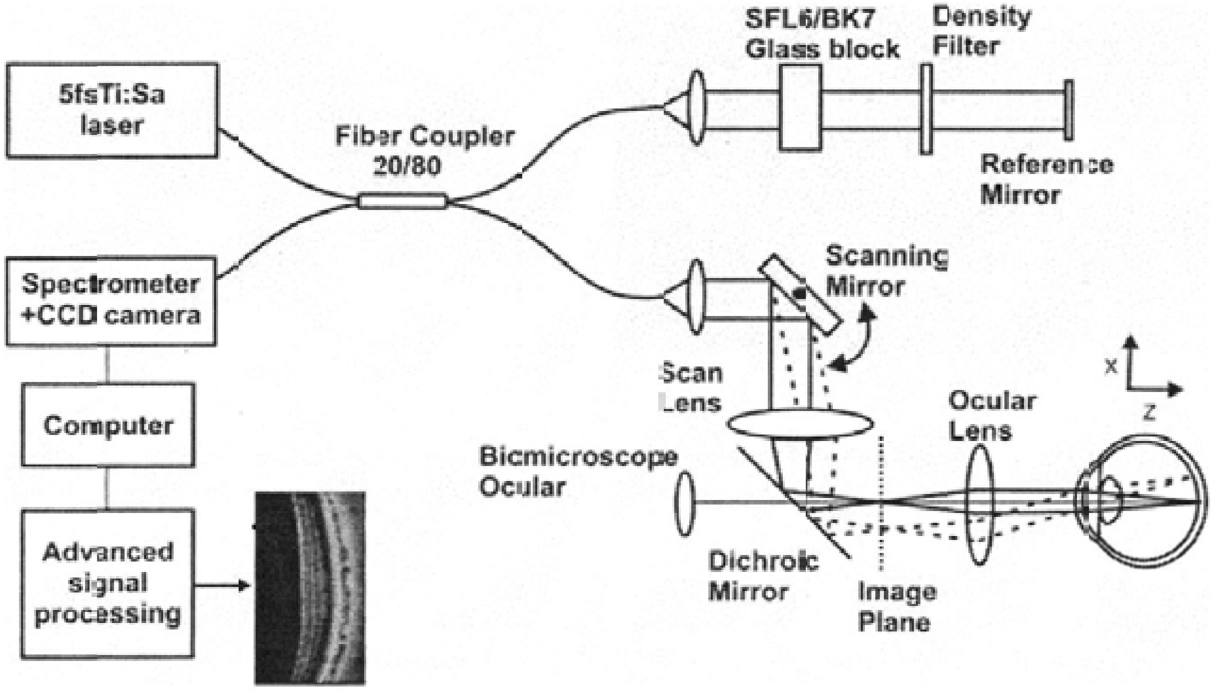

Fig. 1.7 – Schematic of high speed, ultrahigh resolution OCT system using spectral (Fourier) domain detection, the echo time delay and magnitude of reflected light is detected by measuring and Fourier transforming the spectrum of the interference signal [13]

[13] The detection uses a spectrometer in conjunction with a high speed, high dynamic range CCD camera, photodiode array, or linescan camera. The camera records the spectrum of the interference pattern and the echo time delays of the light signal can be extracted by Fourier transform. Because spectral domain OCT measures all of the reflected light at once, rather than

light which returns at a given echo delay, there is a dramatic increase in detection sensitivity. Thanks to this, it is possible to get high speed retinal imaging.

The newest types of OCT devices combines OCT and SLO scanning. First of all, the SLO image is captured, then the user choose the area in SLO image to be scanned with OCT.

1.2.4 SLO (Scanning laser ophthalmoscopy)

[14] Scanning laser ophthalmoscopes provide non-invasive image sequences of the retina with high spatial and temporal resolution and it is a method of examination of the eye which uses the technique of confocal laser scanning microscopy for diagnostic imaging of retina or cornea of human eye. SLO scans images with a laser beam and typically provide standard video outputs (30 Hz) that can be captured by computer systems. The SLO imaging system focuses the laser onto a spot at the retina, on the order of 10 micrometre diameter, that lies about 24 mm beyond the pupil.

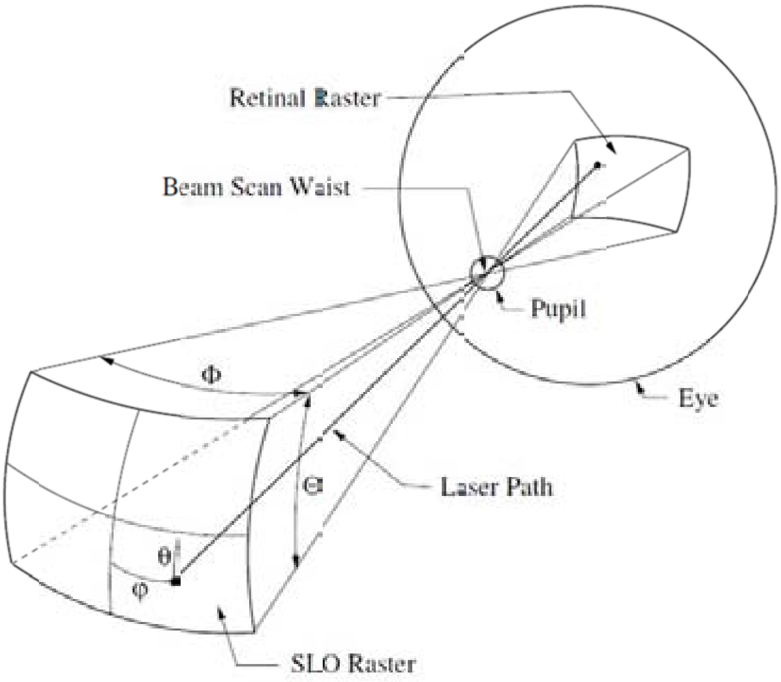

Fig. 1.8 – Representative SLO geometry [14]

"A laser beam of 1 mm diameter is pre-shaped by a lens (L)₁ so as to be in focus at the retina (figure 1.9). The beam passes through a 2 mm aperture (M_1) which is used as beam separator

between the illuminating and reflected light. The laser beam is then deflected horizontally by a rotating polygon mirror (M_2), to form a line scan. The number of facets of the polygon determines the angle of deflection which, in turn, determines the angle scanned in the horizontal direction. A galvanometric mirror (M_3) deflects the beam vertically to form a two-dimensional raster. The two-dimensional raster is focused by a spherical mirror (M_4) to a single spot at the position of the patient's lens. The optics of the eye then focuses this on to the retina. The light reflected from the retina (dotted line in figure 1.9) emerges from a larger exit aperture, travels back along the same path as the illumination beam and is descanned by the two scanning mirrors. The light is collected by the beam separator and focused by a lens (L_2) onto a photodetector. The signal from the photodetector can then either be recorded on video-tape or fed to a frame-grabber interfaced to a computer." [15]

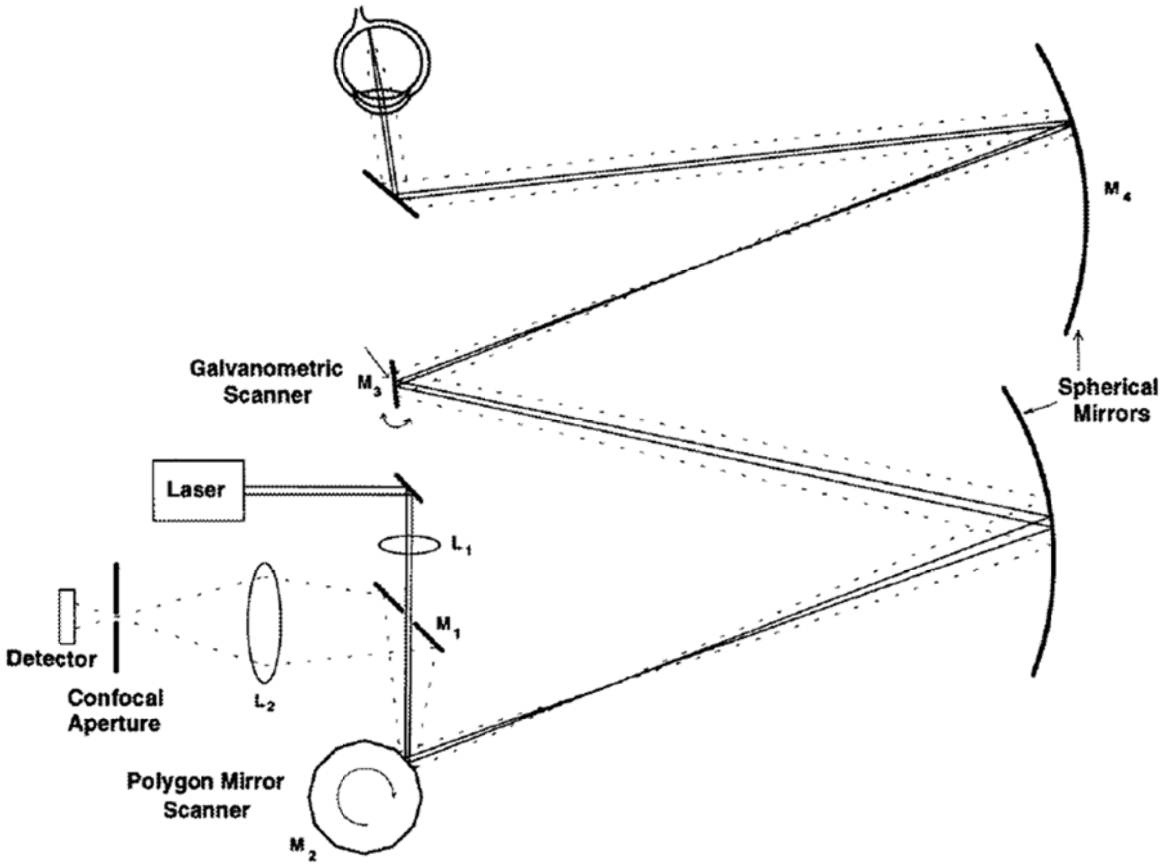

Fig. 1.9 – An optical diagram of the SLO

1.3 Image processing

Image processing is based on signal processing. In this case, the input is an image and the output is image with marked or highlighted elements. These elements are the reasons why the image is processed. There are many different types of methods, thus only used methods in this thesis will be noticed.

1.3.1 Edge detection

Detecting edges in an image is one of the fundamental pillars of the whole image processing [1]. The edge in the image is possible to define as a quick change of intensity in the image between the neighbour pixels. The edge can be part (or another edge) of bigger object in the image, feature, or it can indicates object in itself (e.g. line or point). Unfortunately, it often happens that local changes in intensity are due to noise, and detect as edge. Thanks to various methods of image processing, is possible to eliminate the noise. With global edge detection applied to the image, the secondary binary image is created. This binary image contains only two values – ones and zeroes. Ones are represent as white pixels, zeros as black pixels. Generally speaking, the edge operator puts the edges in place of the maximum response (derivation: responding to rapid change, it is necessary to differentiate according x and y to respond to the horizontal and vertical edges, Laplacian: the edge is where the function crosses zero).

Gradient-based detectors

These detectors are based on the evaluation of intensity changes in the neighbouring pixels and can be described as follows [1]:

$$g_{i,k} = \sqrt{(\Delta_x f_{i,k})^2 + (\Delta_y f_{i,k})^2} \tag{7}$$

where $g_{i,k}$ is a pixel in the image (by i and k coordinates) and $f_{i,k}$ is a pixel of x and y coordinates, from which is $g_{i,k}$ calculated. This operator is based on absolute value, which means that it is isotropic (independent on the direction). The equation (7) can be further approximated as [1]:

$$g_{i,k} = \max(\Delta_x f_{i,k}, \Delta_y f_{i,k}), g_{i,k} \approx \left|\Delta_x f_{i,k}\right| + \left|\Delta_y f_{i,k}\right| \tag{8}$$

As seen from equation (8), these operators are slightly anisotropic (direction-dependent). The first operator emphasizes horizontal and vertical edges, while the second operator highlights angled edges. The decision whether the pixel belongs to a certain edge, is based on comparison of the local gradient and the selected threshold, which is the only parameter of edge detection, given by absolute gradient formula. The threshold determinates whether the pixel belongs to edge representation of the image or not. Higher threshold causes that only the significant edges are highlighted (brighter or darker). If there is not known threshold at disposal, the threshold is set based on visual evaluation of the resulting edge representation.

A similar result as the operator (8) provides the Roberts operator, which calculates the difference in both directions, and the bigger value is considered to be the result. This operator uses the convolution mask, given by [1]:

$$\max\left(h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right) \tag{9}$$

The absolute gradient provides usually relatively thick edges, which is possible to process with operator, which causes the thinning of the edges (erosion operators).

If the partial differential is known, it is possible to determine the direction of local edge using equation [1]:

$$\theta_{i,k} = \arctan \frac{\Delta_y f_{i,k}}{\Delta_x f_{i,k}} \tag{10}$$

The calculation of formula (10) can be approximated by the corresponding value in two dimensional LUT (Lookup table). As a result, parametric edge representation becomes vector: for every positive edge detected pixel is possible to determine the local direction, which can be useful at a higher level of analysis, where the edge consistency tests based on the context.

The edge detection, including the rough estimate edge directions, can be eventually made by so called compass detectors. They are based on the repeated convolution of an image with all eight masks and the set of directional masks, which approximate directional derivatives of the weighted average difference. These masks may look like this [1]:

$$h_0 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$
$$h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_3 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \tag{11}$$

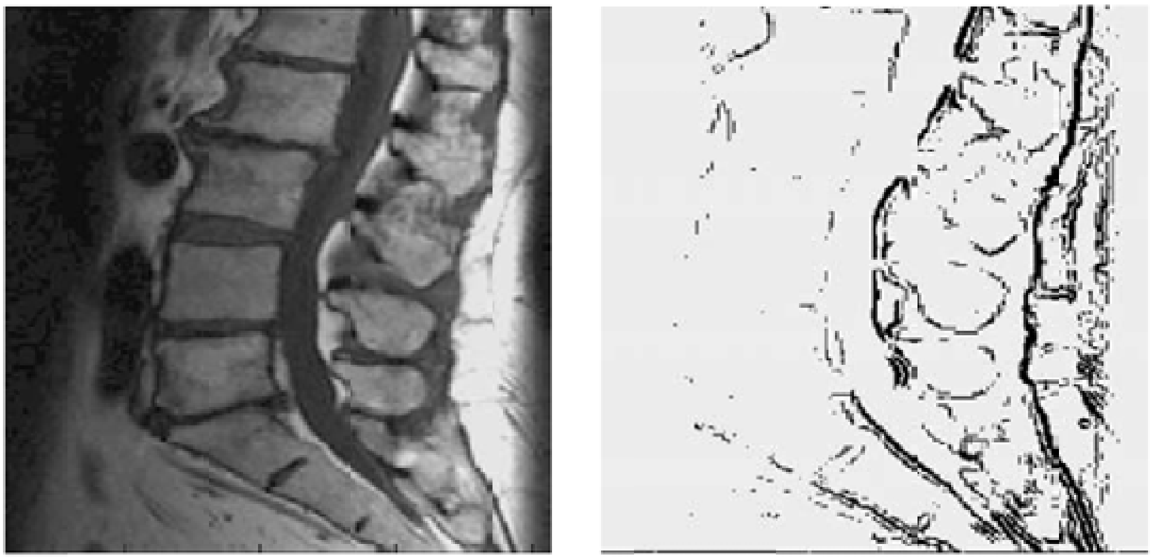

Fig. 1.10 – Original image and its rough edge representation [1]

Each mask gives different parametric image, which highlights a different orientation of the edge. With these masks, it is possible to find maximum for every pixel (i,k) and further threshold them [1]:

$$g_{i,k} = (\max(\left|{}_j g_{i,k}\right|) \geq T), \theta_{i,k} = j_{\max} 45^\circ \tag{12}$$

where $j_{\max}$ is mask's order, which gave the maximum absolute response and T is the chosen threshold. The masks in equation (11) represent so called Sobel operator. Other commonly used operators are e.g. Prewitt operator, Kirsch operator or Canny operator.

Laplacian-based Zero-Crossing detectors

Another commonly used method for edge detection is Laplacian-based detectors. The usual procedure for edge detection using this method, is transform the image into Laplacian image in the first step. Next step is finding the zero-crossings. The procedure provides thinner and usually more precise edges than gradient-based methods.

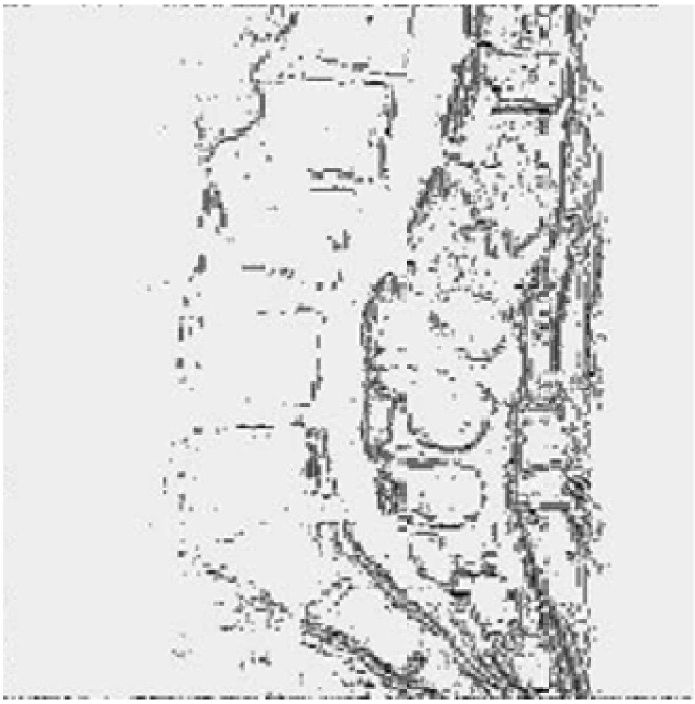

Fig. 1.11 - Raw edge representation based on detection of zero-crossings in the Laplacian [1]

To make this method effective enough, it is used instead of searching for places where mask crosses the zero, mask, which looks for neighbour places with opposite signs. This can be done with the mask [1]:

$$m = \begin{bmatrix} 0 & 1 & 0 \\ 1 & x & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{13}$$

This mask selects the four neighbouring pixels of the tested pixel. "*The further operations are logical:*

1. *if at least one of the neighbours has a different sign than others, and*
2. *if the difference between the extreme differently signed values of neighbours is greater than a chosen threshold T, and*
3. *if the central value at the cross position lies in between the extreme differently signed values of neighbours,*

then the pixel (in a corresponding new matrix of the edge representation) is marked as an edge (e.g., 1). Once any of the conditions is not fulfilled, the pixel is abandoned as a nonedge (0 marked). The procedure again has a single threshold parameter." [1]

The success of this method can be verified with a sufficiently high value of the gradient of the original image. Another test consists in the effective control of the slope between the neighbouring extremes of the Laplacian image with regard to the expected sharp edges that are detected.

1.3.2 Gradient

In vector calculus, the gradient of a scalar field is a vector field which points in the direction of the greatest rate of increase of the scalar field, and whose magnitude is the greatest rate of change [9]. The gradient (or gradient vector field) of a scalar function f(x_1, x_2, x_3, x_4, …, x_n) is denoted ∇f or $\nabla \overline{f}$ where ∇ (the nabla operator) denotes the vector differential operator. The notation *grad(f)* is also used for the gradient. The gradient *f* is defined to be the vector field whose components are the partial derivatives *f*. That is [9]:

$$\nabla f = (\frac{\partial f}{\partial x_1},...,\frac{\partial f}{\partial x_n}) \tag{14}$$

Here the gradient is written as a row vector, but it is often taken to be a column vector. When a function also depends on a parameter such as time, the gradient often refers simply to the vector of its spatial derivatives only. The gradient of a vector **f =** (f_1,f_2,f_3) is [9]:

$$\nabla f = \frac{\partial f_j}{\partial x_i} e_i e_j \tag{15}$$

or the transpose of the Jacobian matrix [9]:

$$\frac{\partial(f_1, f_2, f_3)}{\partial(x_1, x_2, x_3)} \tag{16}$$

In image processing the gradient is often used as the pre-step for edge representation. The gradient (row-wise or column-wise) is used to an image and then the threshold is set. The values which are smaller (or bigger) than current threshold are denoted as 1, the other values as 0. This method is described in detail in Chapter 1.3.1.

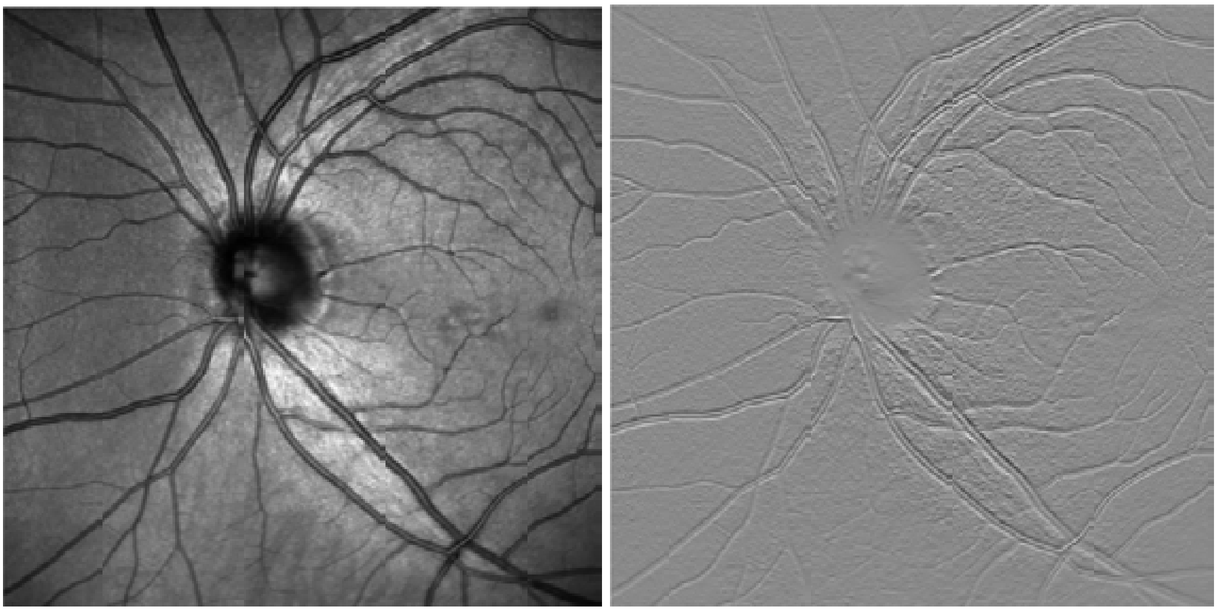

Fig. 1.12 – original OCT (SLO) image (left) and its gradient representation (right) created by MATLAB “gradient” function

1.3.3 Median filtering – noise reduction

Each real image is affected by some noise. This noise leads to inaccuracies in further image processing. For example if there is the edge detection in very noisy image, it may happen that edge detector detects false edges – edges caused by noise, but they are not real edges.

There are several types of noise. For example “the speckle noise” is typical for images for ultrasound scanning devices. Roughly speaking, the noise is caused by diffuse re-processing of signals from multiple distributed targets. For suppressing the noise, the median filter is often used. Generally speaking, the median filter removes extreme values. The median is created as follows: first of all the elements are sorted by value, then the value, which is exactly in the middle of sorted vector is picked and this value is called “median”. If the vector has even number of elements, the arithmetical mean of two values in the middle is picked.

Example a) odd number of elements in the vector

v_1 = [4 6 2 1 3] is sorted as v_1’ = [1 2 3 4 6], median is 3; arithmetical mean is 3.2

Example b) even number of elements in the vector

v_2 = [4 6 2 1 3 4] is sorted as v_2’ = [1 2 3 4 4 6], median is 3.5, arithmetical mean $3.\overline{3}$

[10] The median filter is to run through the signal entry by entry with the median of neighbouring entries. The pattern of neighbours is called the “window”, which slides, entry by entry, over the entire image (or signal). The median operation is nonlinear. The median filter does not possess the superposition property and traditional impulse response analysis is not strictly applicable. The impulse response of median filter is zero.

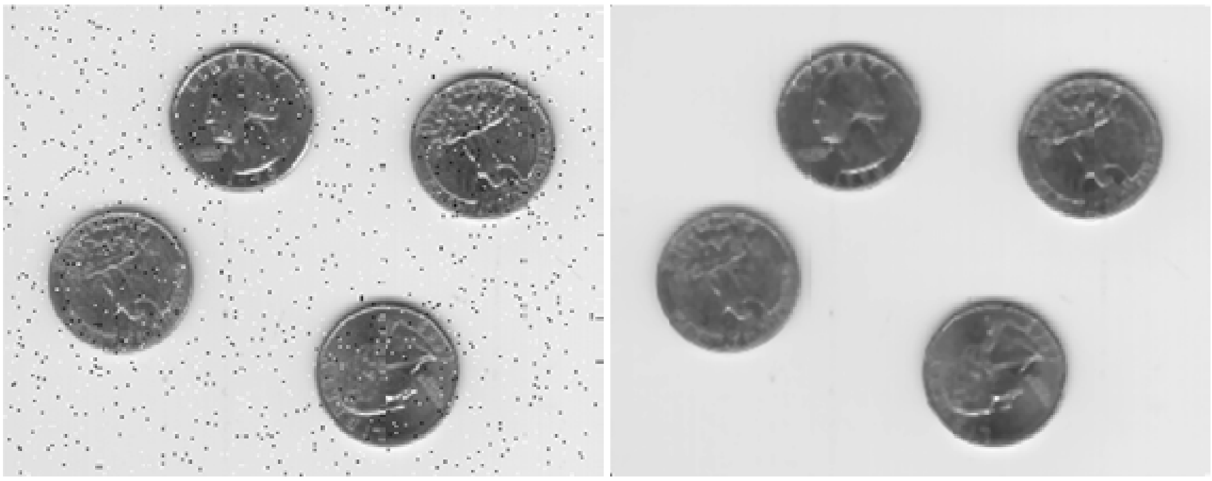

Fig. 1.13 – The noisy original image (left), the image filtered by median filter (right) [8]

1.3.4 Skeletonization

The skeleton [11] is important for recognition and representation of various shapes and features in an image. It contains features and topological structures of an original image. But compared to the original image, the skeleton representation is binary image. Its values are only 1 and 0. The process of creating the skeleton image is called skeletonization, because its main goal is to find object’s skeleton. The skeleton usually emphasizes geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width. Together with the distance of its points to the shape boundary, the skeleton can also serve as a representation of the shape (they contain all the information necessary to reconstruct the shape) [11].

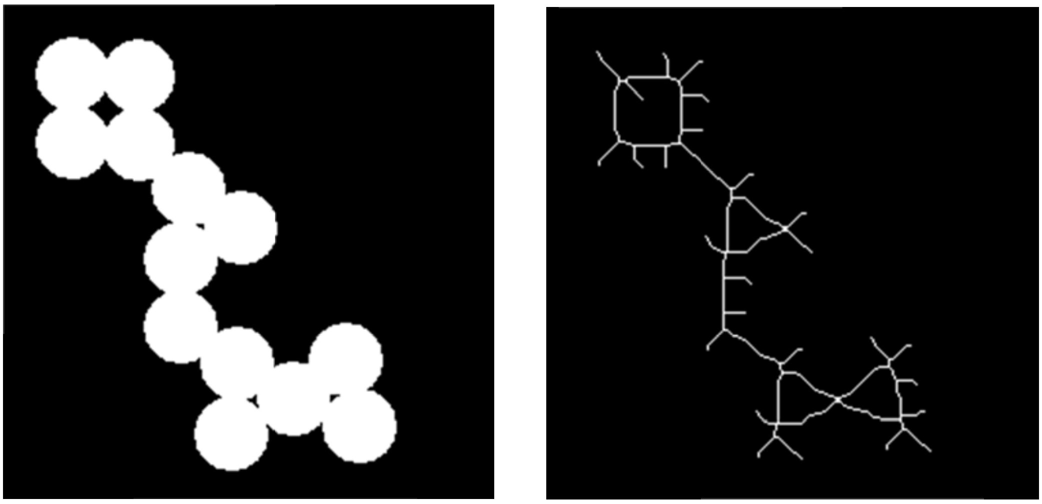

Fig. 1.14 – The original image (left) and its skeleton representation (right)[8]

1.3.5 Correlation

Correlation is a mutual relationship between two variables or processes. If one of them changes, the second one also correlatively changes. If the two processes show a correlation, it is likely depend on each other. The relationship between the variables A and B can be positive if applies y = k.x, or negative y = -k.x. The value of the correlation coefficient of -1 indicates an indirect dependent (anti-correlation), so the more the value increases in the first group of variables, the more it decreases the value of the second group of variables, such as the relationship between the elapsed and remaining time. The value of the correlation coefficient of +1 indicates a direct dependence, such as the relationship between speed and frequency speed of bicycle wheel. If the correlation coefficient is 0, then the variable is no statistically detectable linear dependence. But even with zero correlation coefficient of each variable may the variables may depend on each other. This only means that this relationship can't be expressed as a linear function. In this paper, the variables A and B represents matrices 100x100px of original image (SLO image) and correlated image (fundus image). The correlation coefficient is computed using [8]:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\sum_m \sum_n (A_{mn} - \bar{A})^2 \sum_m \sum_n (B_{mn} - \bar{B})^2}} \tag{17}$$

where $\bar{A}$ is mean of the values of matrix A, and $\bar{B}$ is mean of the values of matrix B. The MATLAB function *corr2* is used for computing the correlation coefficients in this thesis.

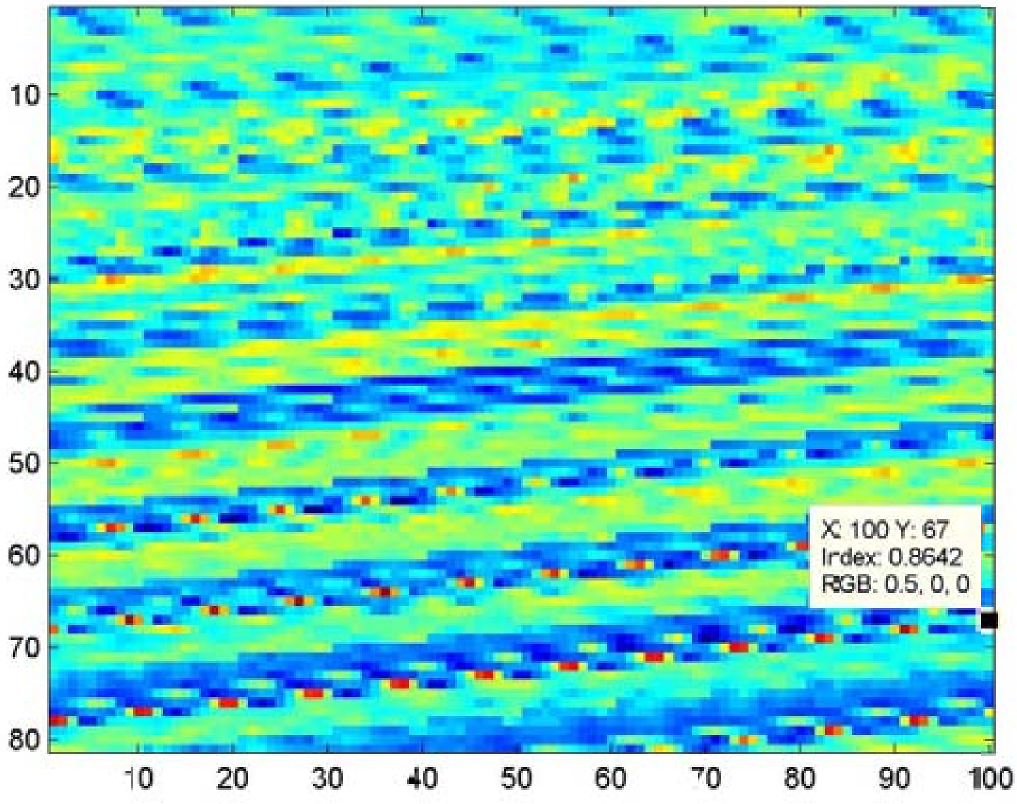

Fig. 1.15 – Image of correlation coefficient matrix with marked highest value (the red pixels represent values nearing 1, the blue pixels represent values nearing -1) which is used for finding corresponding image features in this thesis

1.4 Image registration

1.4.1 Background information

[6] Image registration is the process of overlaying two or more images of the same scene. These images are captured at different time, different angle, from another device or from different location. Thanks to this, it is possible to reconstruct, or highlight necessary information, which is inadequate from one image. The image registration is crucial step in image analysis, in which the final information is obtained by combining data from different sources, such as image fusion, change detection, and multichannel image restoration. Typically, the registration is required in remote sensing (environmental monitoring, weather forecasting, creating super-resolution images, integrating information into geographic information systems (GIS), in medicine (combining computer tomography (CT) and NMR data to obtain more complete information about the patient, monitoring tumour growth, verification of treatment), etc.). Generally speaking, the image registration can be divided into four subcategories according to the image acquisition.

1.4.2 Image acquisition methods

Registration of images taken from various points (multi-view analysis)

Images of the same scene are taken from various points (places). The aim is to get bigger 2D view or 3D representation of the scene. It is used for example in remote sensing from satellites or computer vision (face recognition).

Registration of images taken at different times (multi-temporal analysis)

Images are recorded in the same scene several times over time. It is recorded from the same position, only changing environmental conditions (light, seasons, etc.). The aim is to find changes that are reflected on an object, which is recorded over time (e.g. decomposition of a material, monitoring the growth of tumours, etc.).

Registration of images with different scanning techniques (multi-modal analysis)

Images of the same scene are taken from the different sensors. The aim is to combine information from different sources in order to obtain more comprehensive and detailed representation of the visual scene. For example, different combinations of sensors for recording the human body anatomy (MRI, ultrasound or CT) or a combination of images, OCT fundus cameras to detect diseases of the eye.

Registration of the model and the scene

In this case is registered model of the scene and the real the image scenes. The model may be a computer template scenes sculpted environment map, etc. The aim is to locate the image obtained with the model and compare them. Used for example in remote sensing from satellites to create maps, or layers in the GIS (geographic information system). In medicine, for example for comparing the patient's state bodies with the models or the models corresponding healthy organs.

1.4.3 Most common procedure of image registration

Given the diversity of captured images, which are required to register, it is not possible to use the universal method that would suit any requirement for registration of images. Therefore, all these methods will find their application in practice. The most common procedure for image registration consists of the following steps [6]:

1. Feature detection

Salient and distinctive objects (closed-boundary regions, edges, contours, line intersections, corners, etc.) are manually or automatically detected. For further processing of these features can be illustrated by their point representatives (centres of gravity, line endings, distinctive points), which are called control points (CPs) in the literature.

2. Feature matching

This step is an effort to find the correspondence between the features in the sensed image and those detected in the reference image is established. Various feature descriptors and similarity measures along with spatial relationships among the features are used for that purpose.

3. Transform model estimation

In this step, the estimated type and parameters of the so-called mapping functions and the fitting of the scanned image with reference image are estimated. Mapping function parameters are calculated by the estimated average common features of both images.

4. Image resampling and transformation

The scanned image is transformed by means of the mapping functions. Image values in non-integer coordinates are computed by the appropriate interpolation technique.

1.4.4 Spatial transformation types

The spatial transform [12] represents the spatial mapping of pixel values from the moving image space to pints in the fixed image space. The type of the transform is chosen in order to compensate the distortion between images. The can be several types of distortion due to sequential image scanning. There can be also distortions caused by eye movement during the scanning process. [8] There are many transformation types for image registration. Because this thesis has the practical part in Matlab, the Matlab's *cp2tform.m* function was used. This function takes pairs of control points and uses them to infer a spatial transformation. The *Input Points* is a matrix containing the x and y coordinates of control points in the image which is transformed. The *Base Points* is matrix a containing the x and y coordinates of control points specified in the base image. The *Transform type* specifies the type of spatial transformation to infer. The transformation types which are related to thesis are mentioned.

Non-reflective similarity

[8] This type of transformation is used when shapes in the input image are unchanged, but the image is distorted by some combination of translation, rotation, and scaling. Straight lines remain straight, and parallel lines are still parallel. Non-reflective similarity transformation can include rotation, scaling and translation. At least two pairs of control points are needed.

The transformation equation is:

$$\begin{pmatrix} u & v \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} s.\cos\theta & -s.\sin\theta \\ s.\sin\theta & s.\cos\theta \\ t_x & t_y \end{pmatrix} \tag{18}$$

where *x,y* are pairs of control points from output space, *u,v* are pairs of control points from input space, *s* is scaling, *t* is translation (*t_x* for x axis, *t_y* for y axis), *θ* is angle of rotation

Similarity

[8] The “similarity” type of transformation has the same features as the “Non-reflective similarity” type, but there is the addition of optional reflection. Thus, this type of transformation can include rotation, scaling, translation and reflection. At least three pairs of control points are needed. The transformation equation is:

$$\begin{pmatrix} u & v \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} s.\cos\theta & a.s.\sin\theta \\ s.\sin\theta & a.s.\cos\theta \\ t_x & t_y \end{pmatrix} \tag{19}$$

where *a* determines if there is a reflection or not (for a = -1 reflection is included, for a = 1 reflection is not included in transformation)

Affine

[8] The affine transformation is used when shapes in the input image exhibit shearing. The x and y dimensions can be scaled or sheared independently and there can be a translation. Linear conformal transformations are a subset of affine transformations. Straight lines remain straight, and parallel lines remain parallel, but rectangles become parallelograms. At least three control-point pairs are needed to solve for the six unknown coefficients (equation 20).

The equation is [8]:

$$\begin{pmatrix} u & v \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} A & D \\ B & E \\ C & F \end{pmatrix} \tag{20}$$

where A,B,C,D,E,F are unknown coefficients, the equation (20) can be rewritten as:

$$U = X.T \tag{21}$$

where T is the matrix of 6 unknowns (A,B,C,D,E,F), X is matrix of the input control-point pairs and U is the vector of [u v]. With three or more correspondence points is possible to solve T, with $T = \frac{X}{U}$ which gives the first 2 columns of T and third column must be [0 0 1]' (transposed matrix of [0 0 1]).

Projective

[8] This type of transformation can be used when the scene appears tilted. The quadrilaterals map is matched with quadrilaterals. Straight lines remain straight. Affine transformations are a subset of projective transformations. At least four control-point pairs are needed. This equation for this transformation is:

$$\begin{pmatrix} up & vp & wp \end{pmatrix} = \begin{pmatrix} x & y & w \end{pmatrix} \begin{pmatrix} A & D & G \\ B & E & H \\ C & F & I \end{pmatrix} \tag{22}$$

where $up = u.wp$ and $vp = v.wp$, T is matrix 3x3 of unknown elements (A,B,…,I) of equation (22). Assuming: $v = \frac{(Ax + By + C)}{(Gx + Hy + I)}$ $u = \frac{(Dx + Ey + F)}{(Gx + Hy + I)}$ is possible to solve nine elements of T. The detail procedure can be found in [8].

Polynomial

[8] This transformation type is needed when objects in the image are curved. The higher the order of the polynomial, the better the fit, but the result can contain more curves than the base image. For second order, at least six control-point pairs are need. Polynomial functions of x and y determine the mapping. For third order, at least ten control-point pairs are needed. Following equation is for the second order of polynomial function:

$$\begin{pmatrix} u & v \end{pmatrix} = \begin{pmatrix} 1 & x & y & x.y & x^2 & y^2 \end{pmatrix} \begin{pmatrix} A & G \\ B & H \\ C & I \\ D & J \\ E & K \\ F & L \end{pmatrix} \tag{23}$$

Considering the fact, that polynomial transformation type is not used in this thesis, further details are not mentioned. They can be founded in [8].

2 PRACITAL APPLICATION

Part of this thesis is to design an application in Matlab for image registration according to submission of the thesis. First step was to create program for pre-processing the images from Fundus camera and SLO image.

2.1 Pre-processing of fundus and SLO images

This pre-processing of fundus image is performed by *fce_cteni_fundus_man.m* function in Matlab for manual registration and by *fce_cteni_fundus.m* for automatic registration.

2.1.1 Input image

First of all, the fundus image is loaded. Because the fundus images are often taken through the red filter, it is necessary to remove the red (R) component of RGB image. When the red component is not removed and the RGB fundus image is transformed to grayscale fundus image, image becomes very bright with lack of contrast. That is why only green and blue component of RGB image is used.

2.1.2 Pre-processing

The image is divided by 255 to get the values in image between 0 and 1. The next step is to find edges of the eye of the fundus image. It is performed by finding the big differences of value of two neighbour pixels. After edges are found, the image is cut to show only the square of the eye. When the registration program starts, the user is asked if he wants to have original round shape of the eye, or if the inner square of the eye should be chosen (figure 2.2). If the user presses “1”, the round shape is used (figure 2.6), otherwise the inner square of the eye is cut and used (figure 2.5).

2.1.3 Output image

The output image of fundus is square, grayscale image with only G and B components of RGB spectrum. Then it is resized to 1000x1000. Image is figured on figure 2.2. The image of SLO is not modified, only resized to 1000x1000 pixel size.

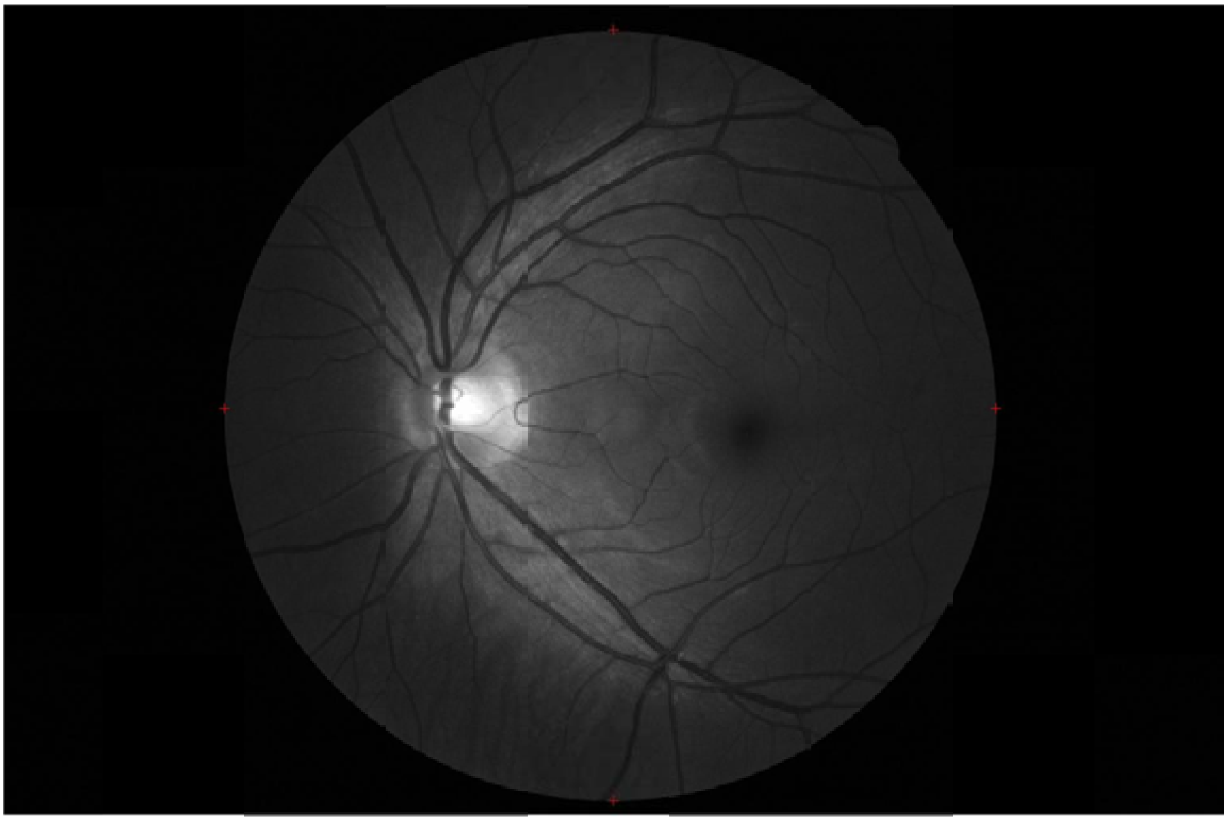

Fig. 2.1 – Fundus image with highlighted edges

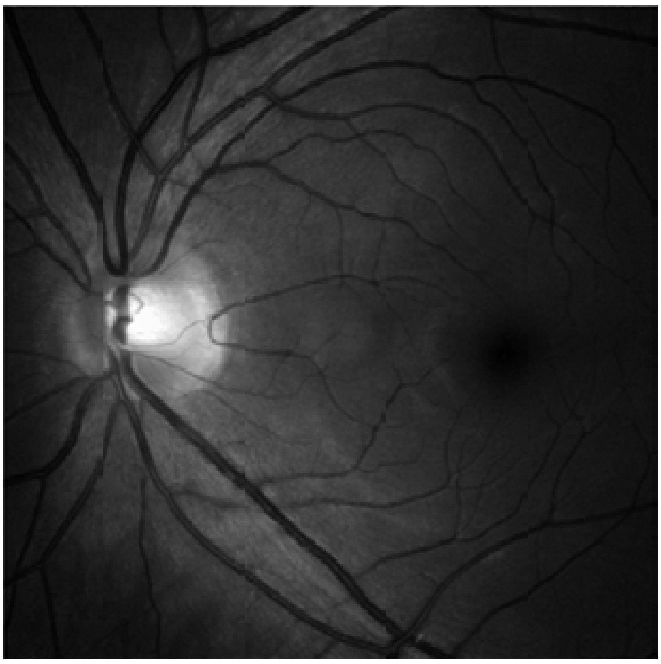

Fig. 2.2 – Final form of pre-processed fundus image

2.2 Manual registration of fundus and SLO images

In this step, the SLO image and the fundus image are registered. First of all, the number of registration points is set. The default setting is 4 (because of the used spatial transformation method), and the spatial transformation method is “affine” (for details see chapter 1.4.4).

2.2.1 Choosing the input points

The input points have to be chosen very carefully. It is good to mark the most significant points in the image (for example: the vessel crossings, branching of vessels, etc.). It is not recommended to mark the points in the blind spot (*loco caecus*), because on the fundus image the blind spot represented mostly as white points, but in SLO image is represented mostly as black points and thus it is difficult to choose the exact same points.

2.2.2 Choosing the reference (base) points

The base points have to match exactly (in significance of points) the input points otherwise the registration result would not be accurate. It means the exact crossings have to be marked on the input and reference image (see figure 2.3).

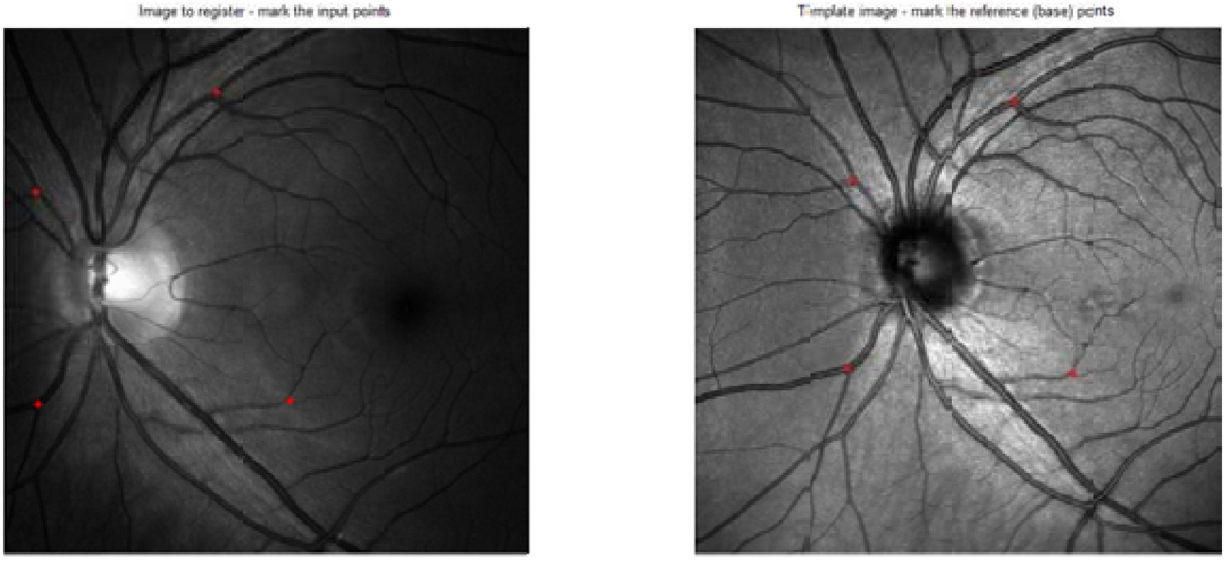
Image to register - mark the input points
Template image - mark the reference (base) points

Fig. 2.3 – Choosing the input and reference (base) points of fundus image (left) and SLO image (right)

2.2.3 Registration result

After the spatial transformation which is provided by cp2tform function (for details see chapter 1.4.4) the images are registered. As the figure 2.4 shows, the images are in this step registered. The fundus image represents smaller area than the SLO image that is why the fundus image is registered to the SLO image and not vice versa. The left image on figure 2.4 shows the connection of the vessels on the edge of fundus image whereas the right image shows the overlap of both images. Better view of registration result provides the figure 2.5. For this image was used the function *MakeChessBoard.m* which was created by Doc. Kolář of Department of biomedical engineering of Faculty of electrical engineering and communication of Brno University of Technology. This function creates a “chessboard” from both images. The size of each “window” of the chessboard is 100x100 pixels. The figure 2.5 shows how good are the vessels of both images connected in the different parts of the image or how far was the registration successful. The possible error is caused by user who did not select the exact points in both images.

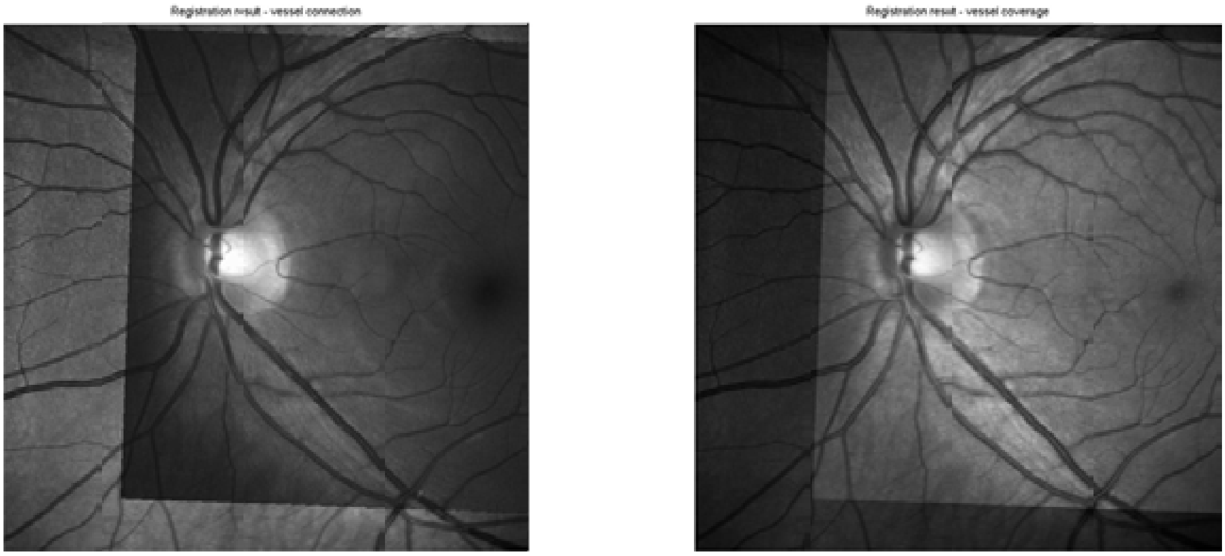
Registration result - vessel connection
Registration result - vessel coverage

Fig. 2.4 – Registration result – vessel connection (left), vessel coverage (right)

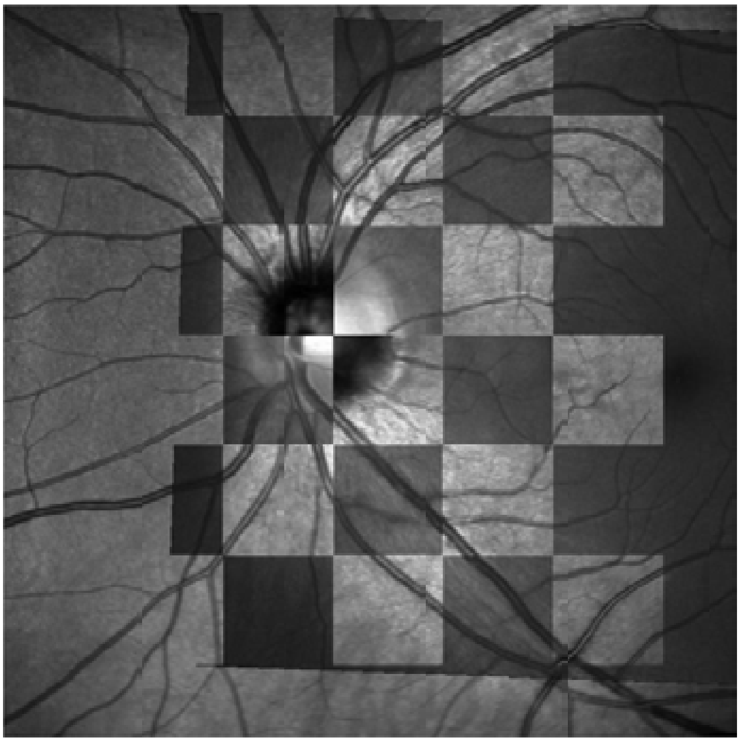

Fig. 2.5 – The "Chessboard" representation of result using the inner square of fundus image

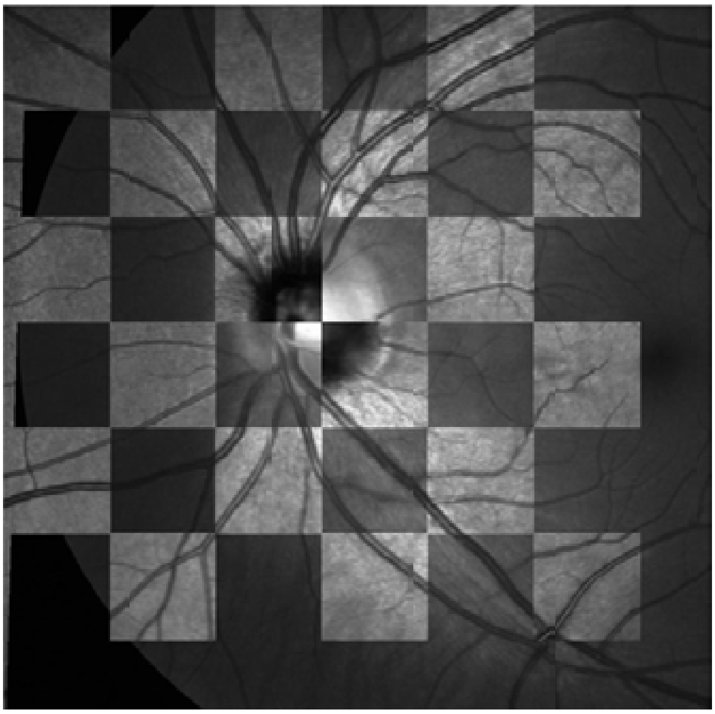

Fig. 2.6 - The "Chessboard" representation of result using original eye's round shape

2.2.4 Manual registration block diagram

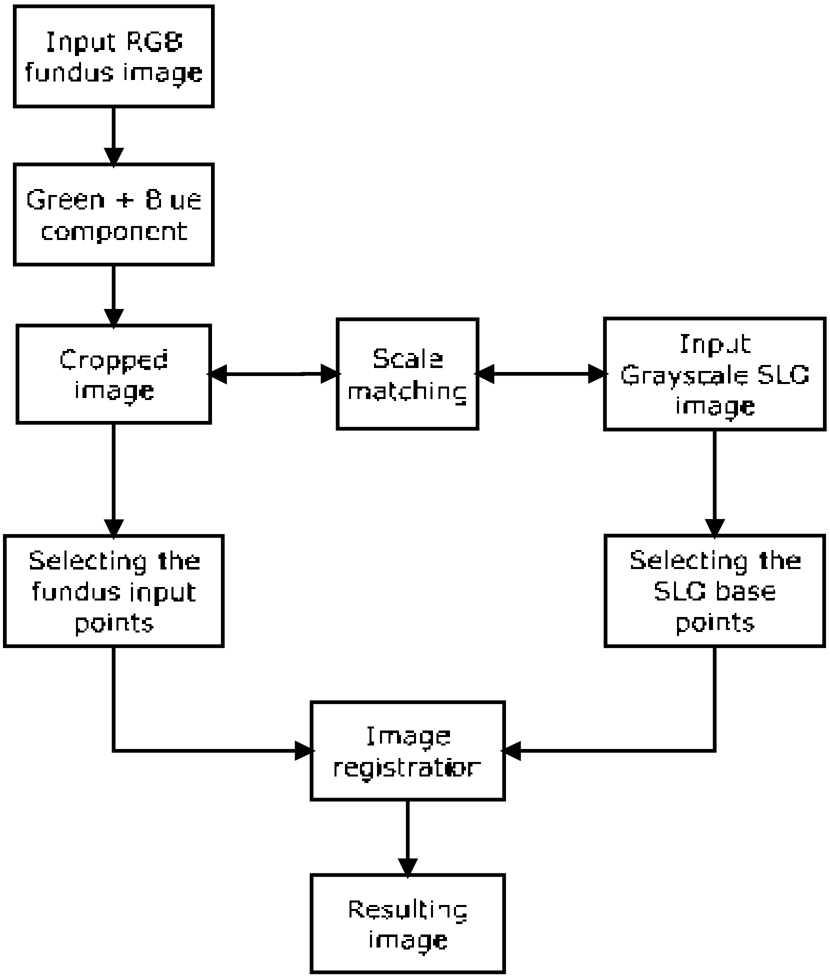

Fig. 2.7 – Manual registration block diagram

2.3 Automatic registration of fundus and SLO images

In automatic registration the input and base points of images are selected automatically. First step is the same as in the manual registration. Images have to be pre-processed by functions mentioned in chapter 2.1.

2.3.1 Choosing the reference points

First step is to automatically select significant points which will be the input points. This is done by following procedure. The gradient of reference image is created (more details about gradient can be found in chapter 1.3.2). The threshold is set according to the image maximal intensity. The values above this threshold are set to 1 and below the threshold to 0. Because of it, the binary representation of the reference image (SLO) is created. To find required input points is necessary to thinner the vessels and other structures represented in binary image as ones. This can be done by using skeletonization in the reference image (chapter 1.3.4). In this program the skeletonization is performed by Matlab's *bwmorph.m* function. After the skeletonization the image can be very noisy, so it is recommended to filter it by median filtering (chapter 1.3.3). For this the *medfilt2.m* function is used. Then the different regions of the reference image are searched to find significant points (such as vessels, vessel crossings, etc.). The four areas of approximately 100x100 pixels are searched and the algorithm marks the ones which neighbour each other. The algorithm for searching one area looks like this:

```
for i = 100:200
    for j = 300:400
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+2,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1))) && sw == 1;
            cnt = cnt + 1;
            if cnt == round(cnt1/2)
                sw = 0;
                plot(j,i,'g+');
            end
            oct_points(1,2) = j-win_size/2;
            oct_points(1,1) = i-win_size/2;
        end
    end
end
```

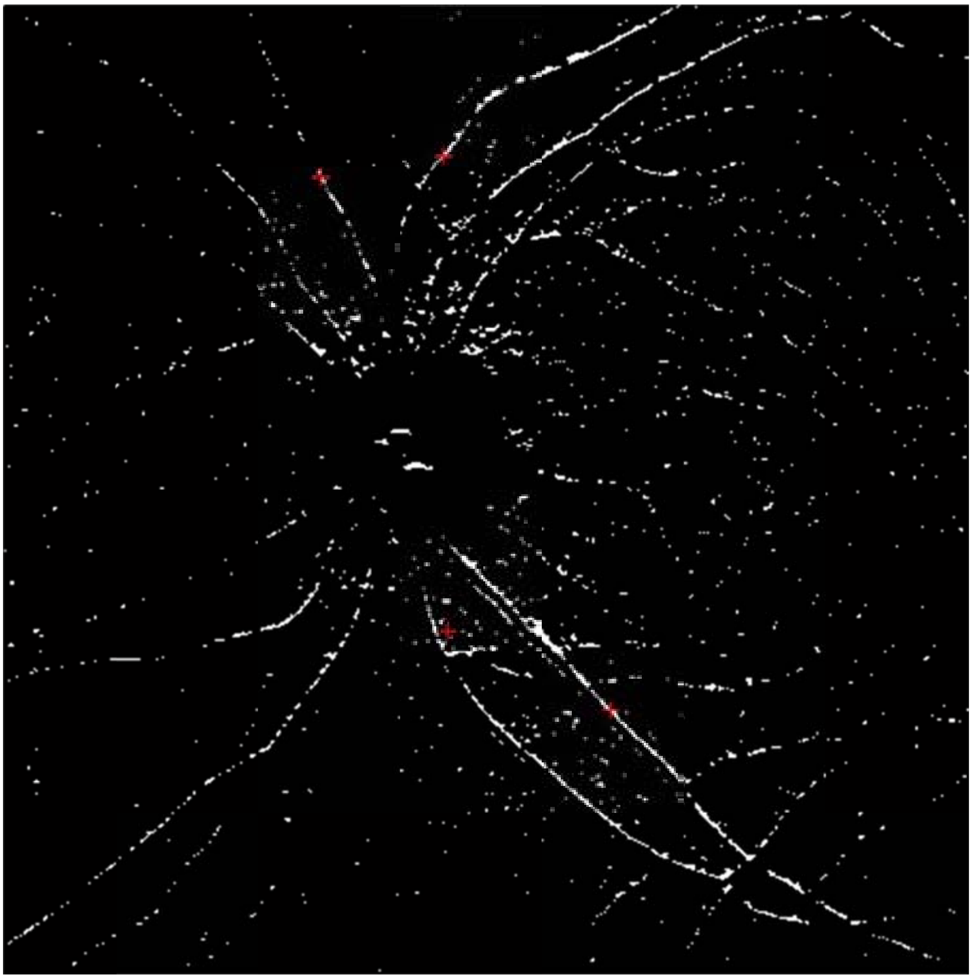

Fig. 2.8 – The binary image with marked reference points (red pluses)

2.3.2 Finding the similar windows

After the points are found, the 100x100 pixel windows of SLO are cut around them. The four found points are always in the middle of cut windows. These windows are in next step compared by correlation (chapter 1.3.5) to the all areas of fundus image. Depending on where is the maximum coefficient of correlation, the most similar window is found. The size of the window 100x100 pixels was chosen because it is the best compromise between the speed of the algorithm and the ability to find exact match. The user can change the size of the window in the program. If the window is bigger, the algorithm is faster, but may have difficulties to find the exact matches. The algorithm which performs correlation looks like this:

```
for i = 1:step:X-win_size
    for j = 1:step:Y-win_size
        if counter2 == Y/step;
            counter2 = 0;
            counter = counter+1;
        end
        counter2 = counter2 + 1;
        cor_coef(counter,counter2) = 
corr2(oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1),fundus(i:i+win_size-
1,j:j+win_size-1));  % picking up the correlation coefficients
    end
end
```

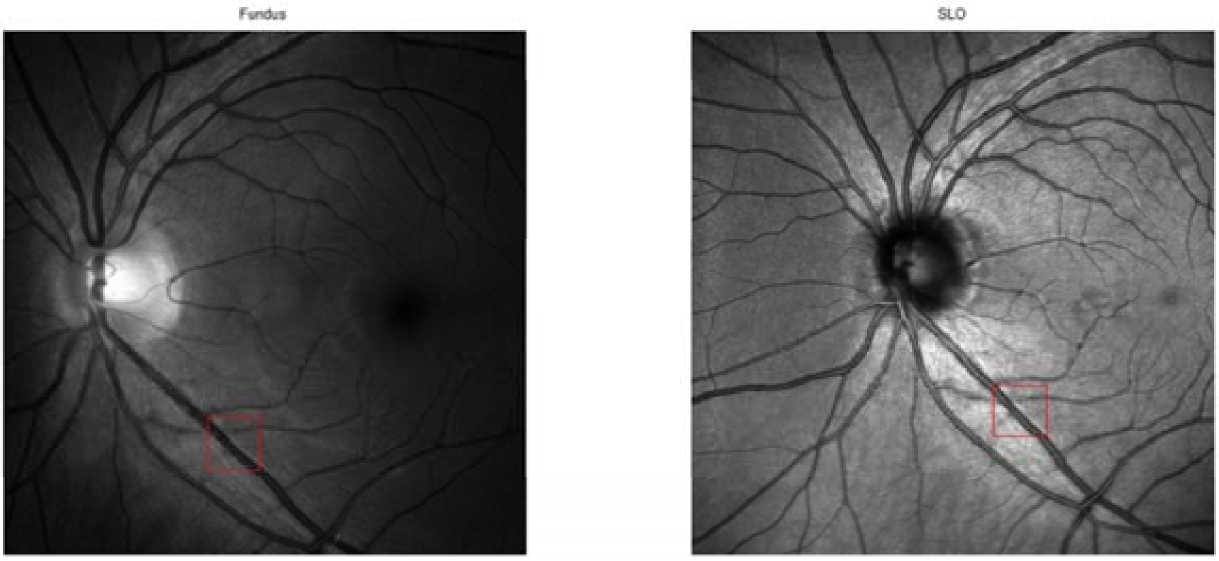

Fig. 2.9 – The window selected by algorithm in SLO image (right), and the window found by correlation in fundus image (left)

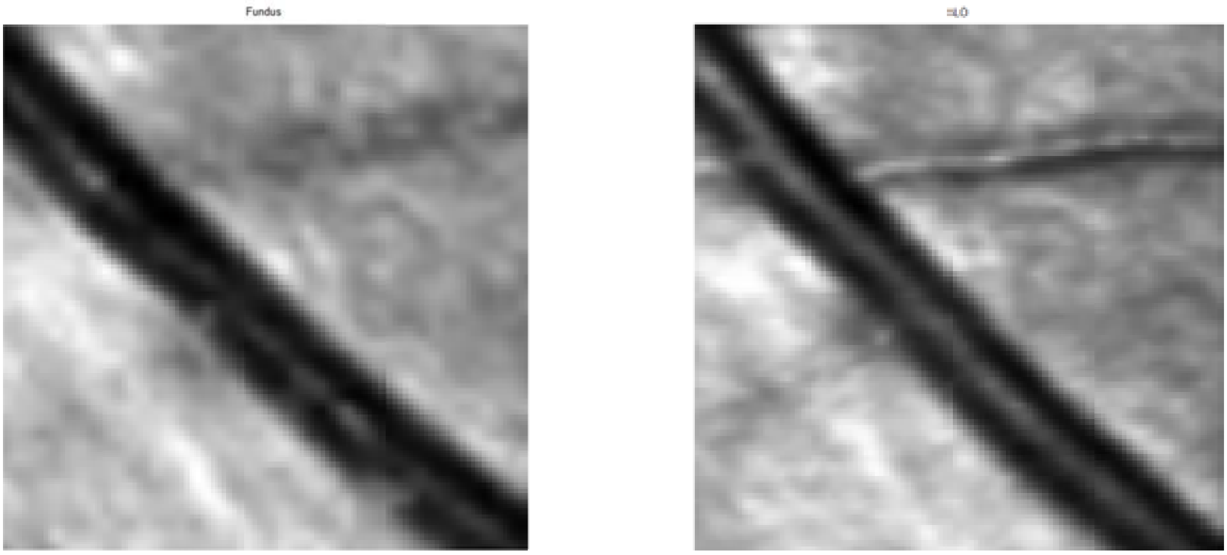

Fig. 2.10 – Detail on the original window (right) - SLO, and found window (left) - fundus

2.3.3 Finding the minimal value of found windows

Because the size of correlation window is relatively big, there can be relatively big error in the correlation. The spatial transform algorithm needs very precise matches in both images. That is why another correlation is needed. In the SLO small 100x100 pixel window, the minimal value is chosen (figure 2.10), and another smaller window (20x20 pixels) surrounding the minimal value is created. This 20x20 pixels window is again correlated with smaller windows (20x20 pixels) in fundus 100x100 pixels window (figure 2.11). This operation is repeated for four times in each found 100x100 window (because at least 4 points are needed for the spatial transformation). Thanks to this, the found points more match to the reference points.

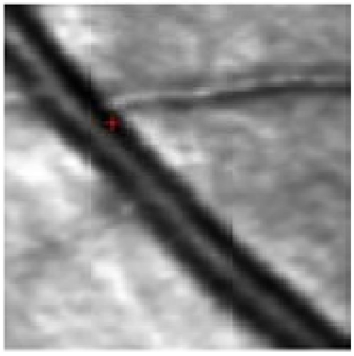

Fig. 2.11 – 100x100 pixels window of SLO image with minimal value marked

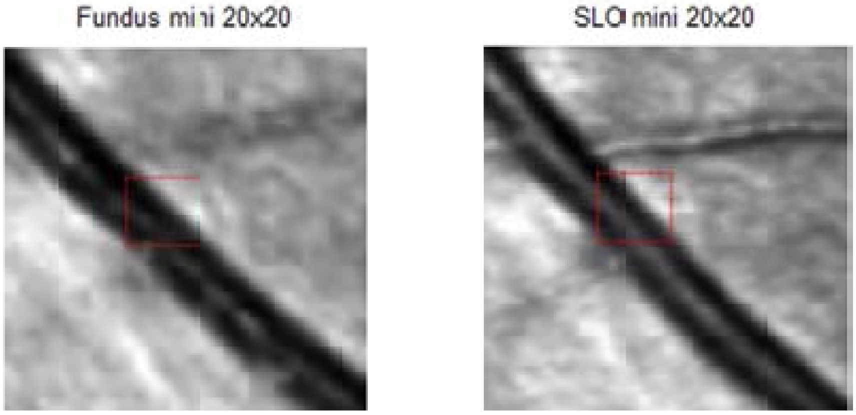

Fig. 2.12 – 100x100 pixels window with marked 20x20 pixels window surrounding the minimal value (right) and its marked match (left) in fundus 100x100 pixels window

2.3.4 Automatic registration result

All needed points are now found and set, thus the next step is the spatial transformation. The results of the automatic image registration are very similar to manual registration, but in some cases, when for example the images are damaged or heavily distorted it is better to use manual registration. If the figures 2.14 and 2.5 are compared, it is possible to see that manual registration have slightly better results in right top corner of the result image. It is possible to enhance the automatic registration by choosing the smaller correlation windows, or lowering the thresholds for the binary representation. Another option is to try different type of spatial transformation but it turned out to be not so effective. The evaluation of automatic registration is in the supplement A. If the number of average deviation is higher than 2, registration was not successful. The best result provides so called semi-automatic registration.

2.3.5 Evaluation of automatic registration result

In the supplement A can be found the evaluation of automatic registration result. For the evaluation the manual and automatic registration were used. The manual registration is the reference registration. First of all, the automatic registration result is subtracted from manual registration result. If there is a deviation in the same pixel in both images, its absolute value is transformed to percentage and then the average and maximum deviation is found. If the average deviation is bigger than 2%, the registration was not successful. The maximal deviation determines the biggest error in the registration result but it is not dependents whether the registration was successful or not. The equation for counting the deviation could be written as:

$$C = \frac{|A - B|}{255}.100 \tag{24}$$

where C is the result matrix of subtracted images, A is the matrix of manual registration result and B is the matrix of automatic registration result.

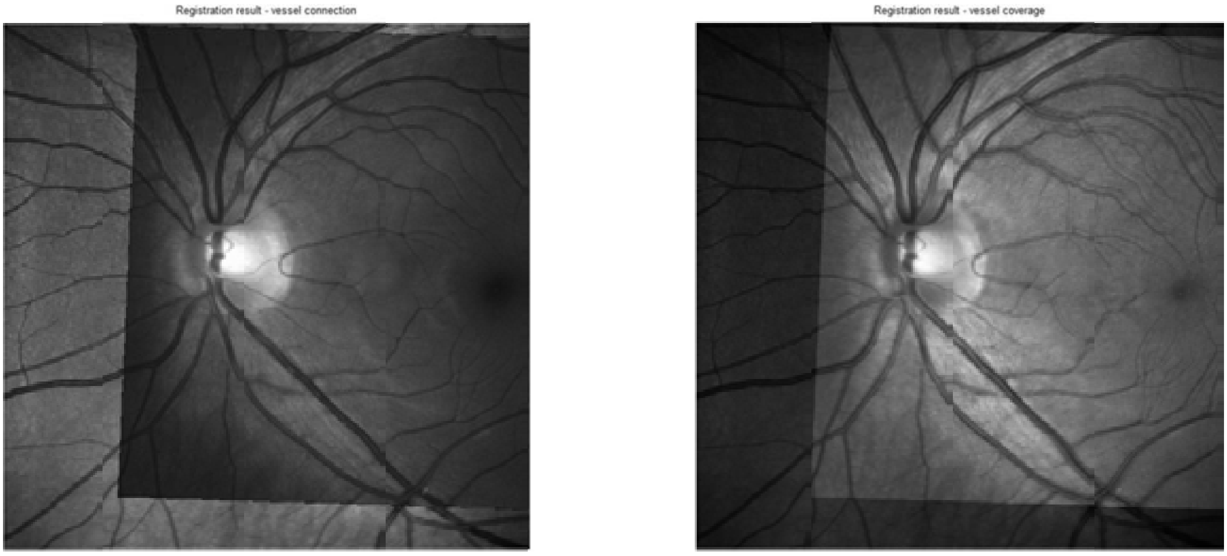
Registration result - vessel connection
Registration result - vessel coverage

Fig. 2.13 – Registration result – vessel connection (left), vessel coverage (right)

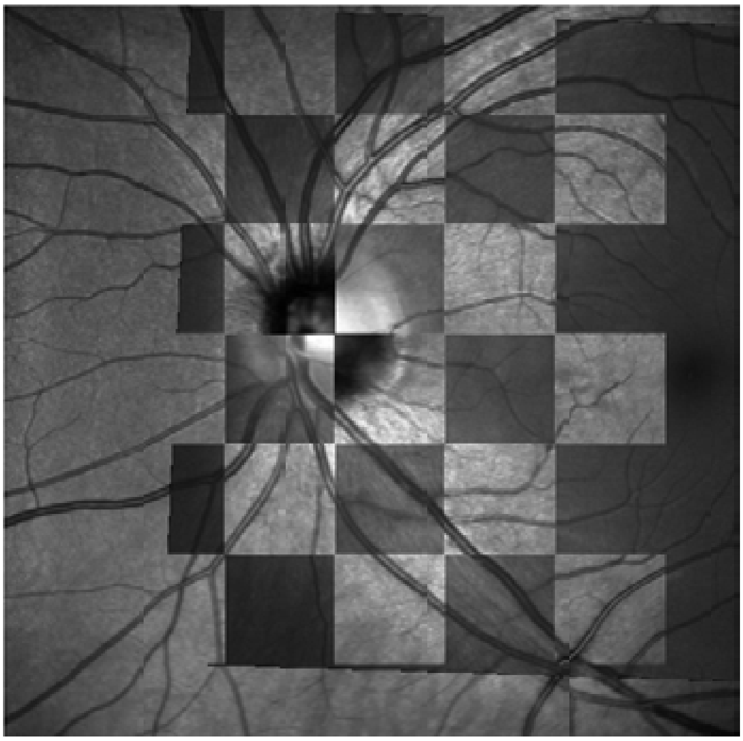

Fig. 2.14 - The "Chessboard" representation of the automatic registration result

2.3.5 Automatic registration block diagram

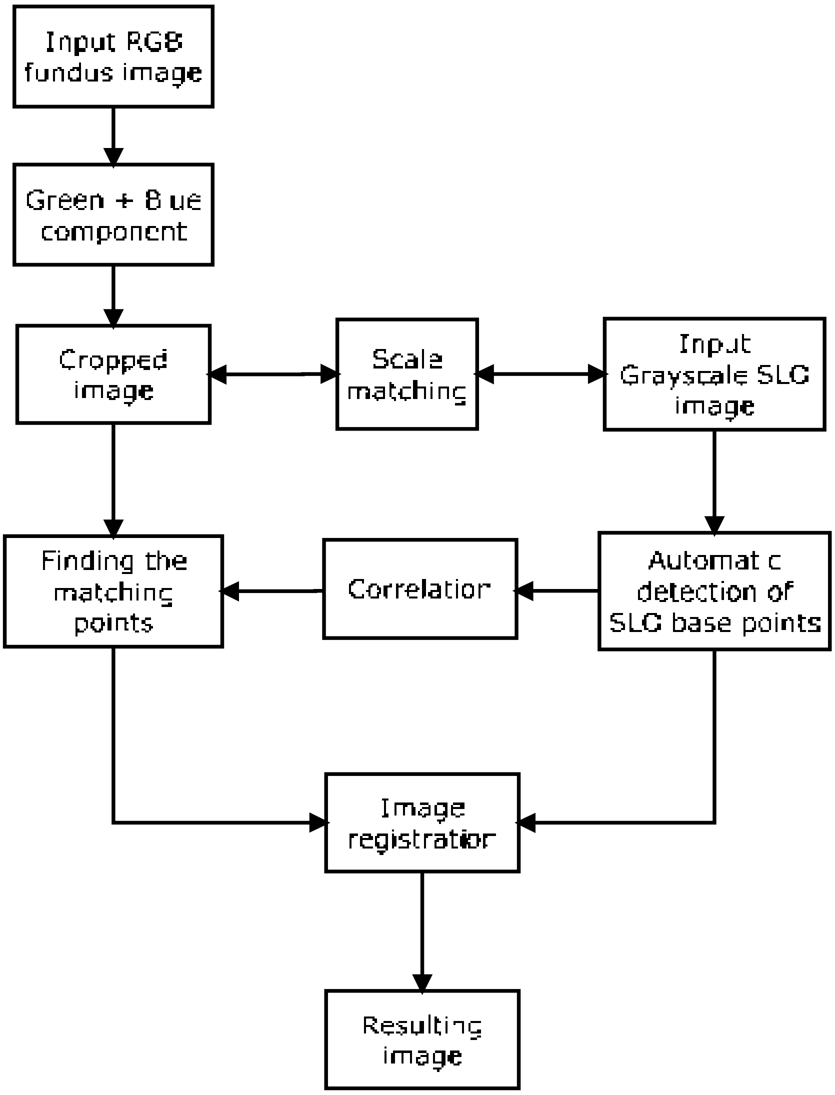

Fig. 2.15 – Automatic registration block diagram

2.4 Semi-automatic registration of fundus and SLO images

2.4.1 Semi-automatic registration result

This type of registration combines advantages of two previous types (chapter 2.2 and chapter 2.3). The user manually selects the reference points in the SLO image and the input points are found automatically. Thanks to this, the match of the points is nearly 100 %, because the error of selecting not the exact points by user is eliminated. It is recommended to mark points which represent vessels crossings or thick vessels. The user only has to be aware of the limitation of the fundus image, which represents smaller area than the SLO. Because the semi-automatic registration is combination of two previous methods, the procedure is practically the same, thus it will be not mentioned. This method is the most accurate.

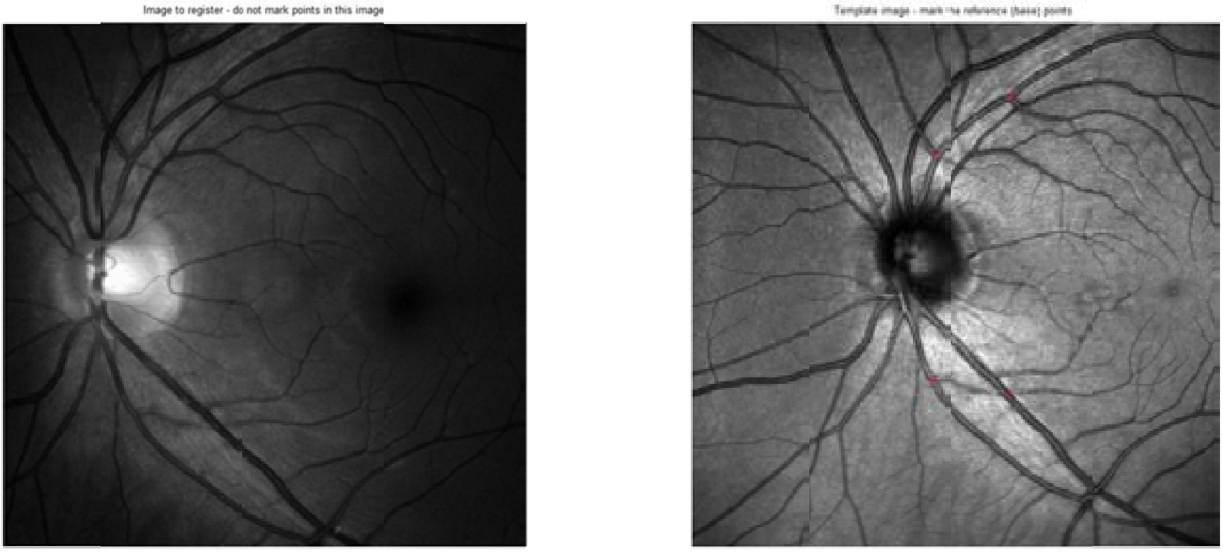
Image to register - do not mark points in this image
Template image - mark the reference (base) points

Fig 2.16 – Selecting the reference points in the SLO image

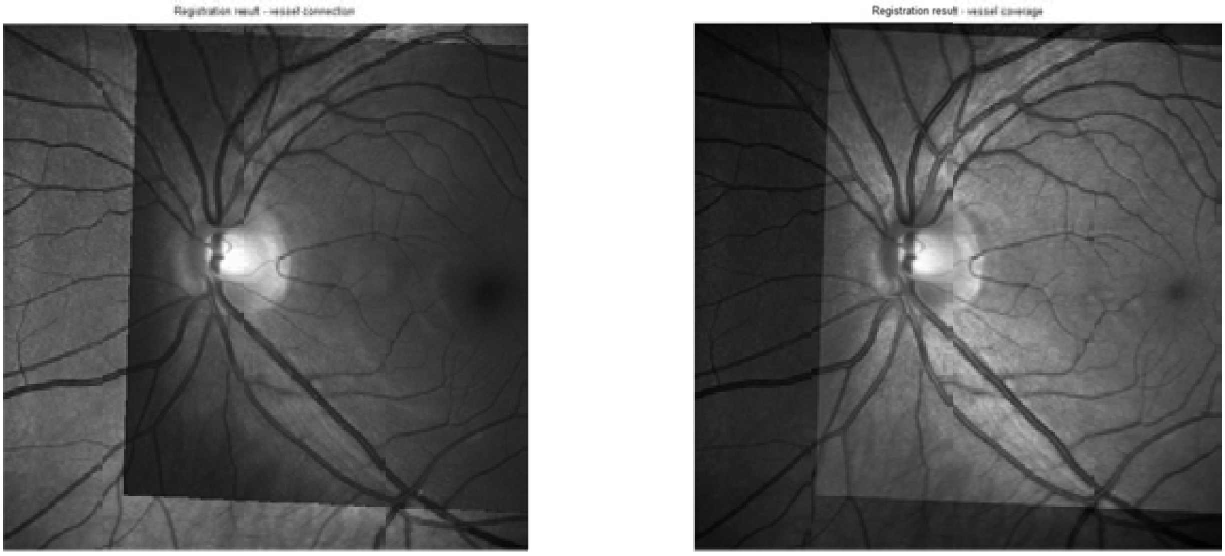
Registration result - vessel connection
Registration result - vessel coverage

Fig. 2.17 – Registration result of semi-automatic registration – vessel connection (left), vessel coverage (right)

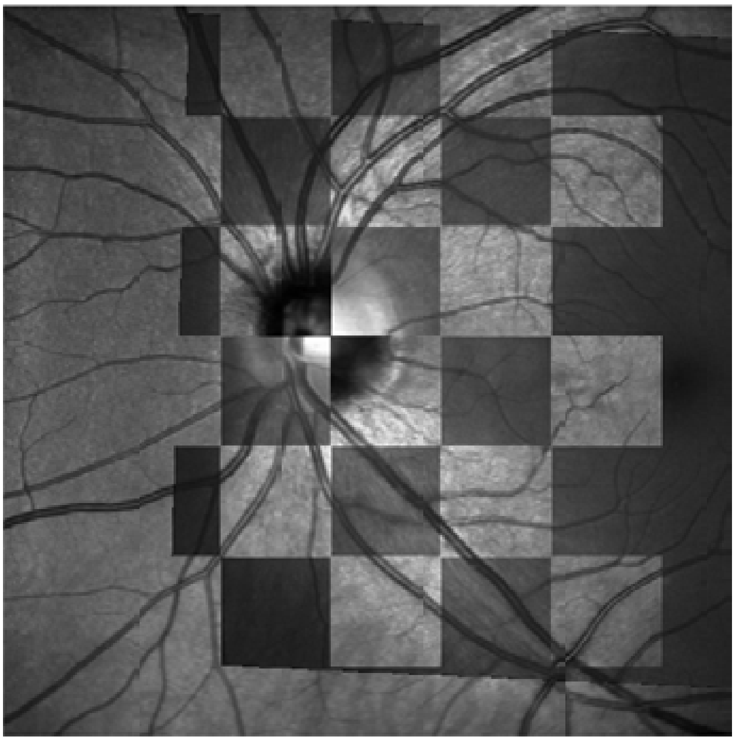

Fig. 2.18 - The “Chessboard” representation of the semi-automatic registration result

2.4.2 Semi-automatic registration block diagram

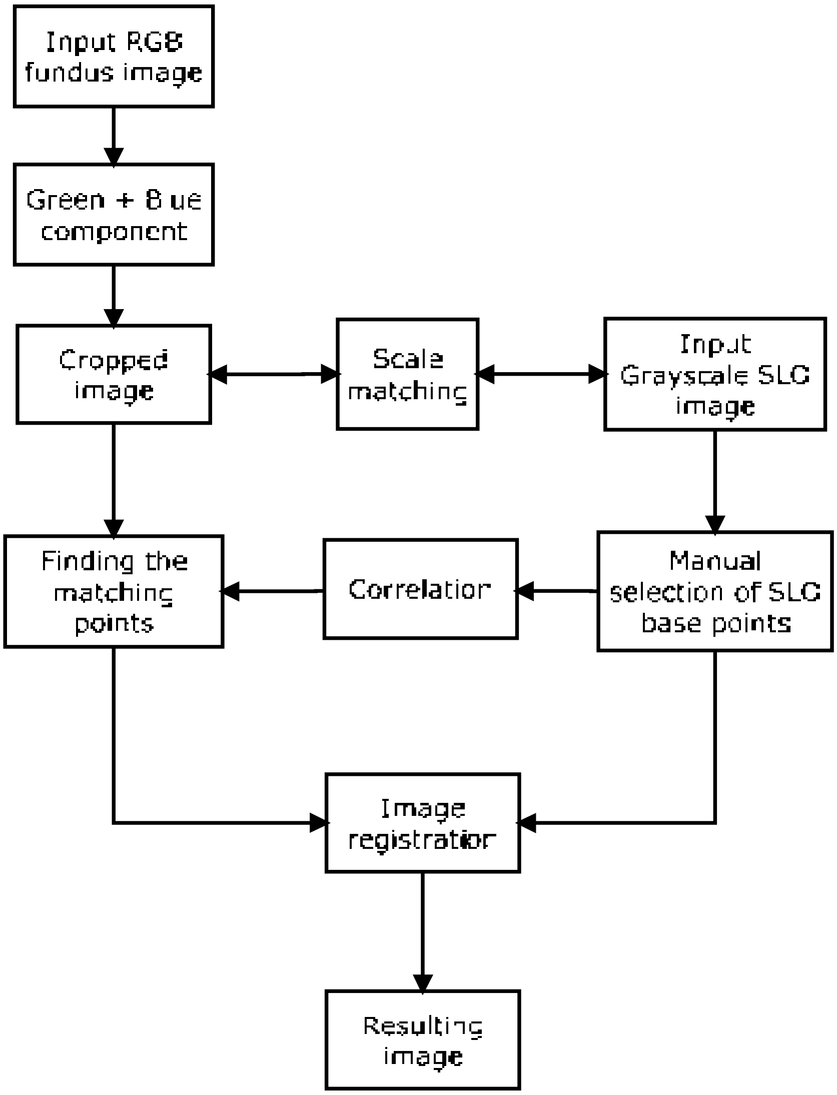

Fig. 2.19 – Semi – automatic registration block diagram

2.5 The fundus image and OCT B-scans matching

The final task was to match the fundus image with OCT B-scans. Each B-scan represents approximately one line of fundus image. B-scans are in sectional view (see figure 1.6). From these B-scans is possible to reconstruct image which is similar to fundus image and thus it is possible to register these images.

2.5.1 Pre-processing of OCT and fundus images

For successful matching is the pre-processing crucial. First of all the B-scans have to be transformed to another view. For this transformation, the function *vyrovnani_upravene.m,* is used. It was made by Ing. Gazárek of Department of biomedical engineering of Faculty of electrical engineering and communication of Brno University of Technology. In this function the lower edge of the retina is detected (see figure 2.21). Then about 10 pixels below each one pixel of the edge are picked and average is made. This average represents one line of the reconstructed image, which is similar to fundus or SLO image (see figure 2.22).

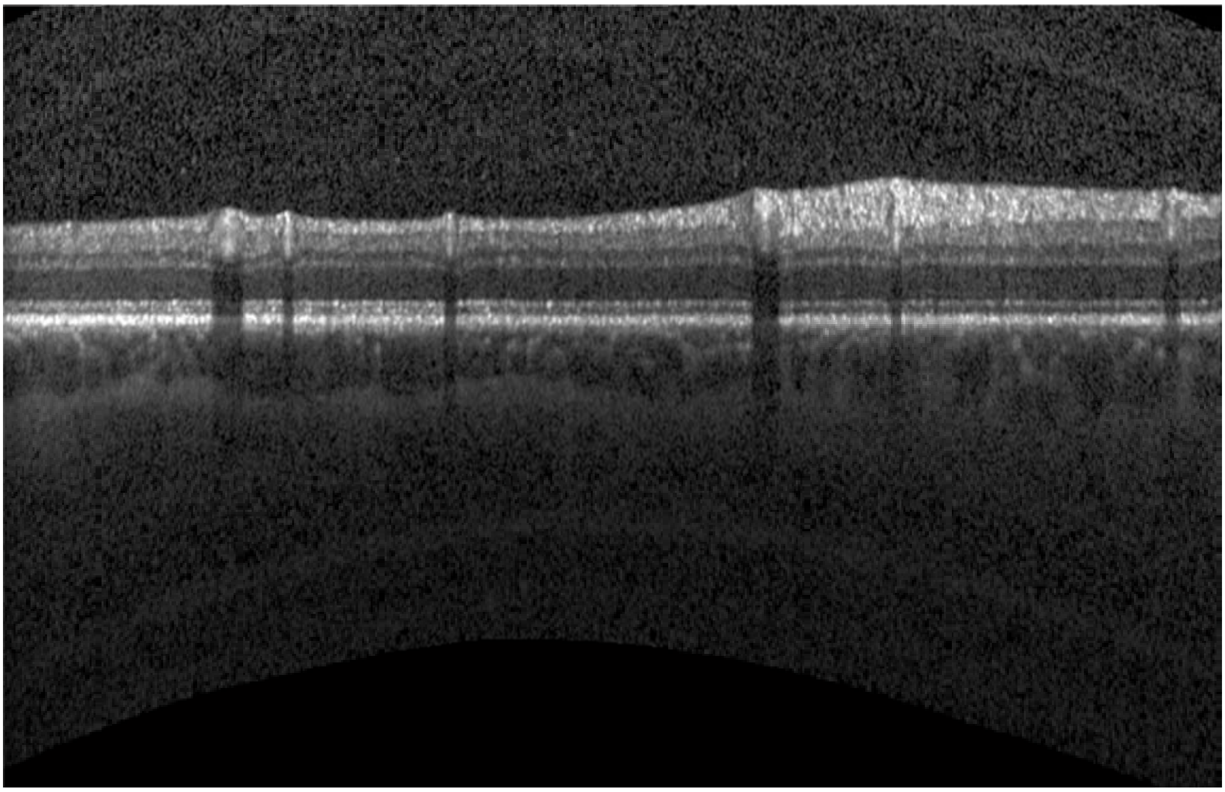

Fig. 2.20 – OCT B-scan

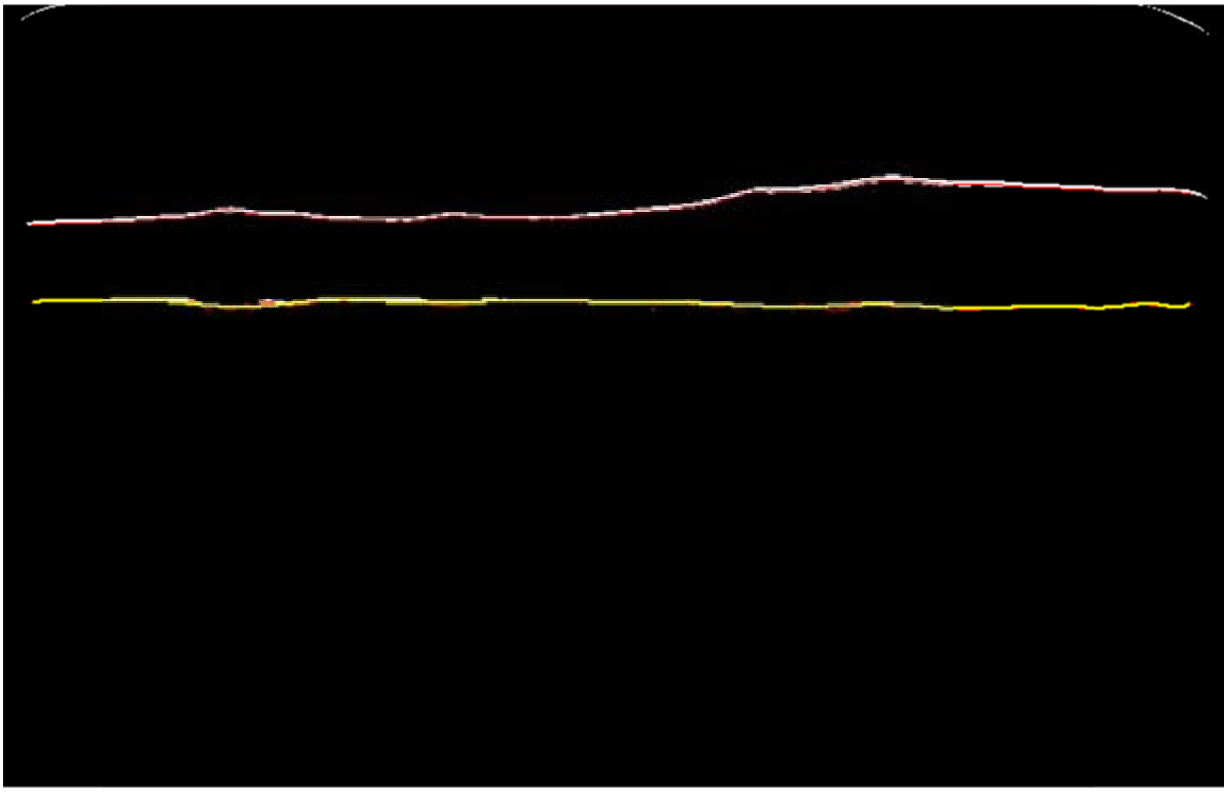

Fig. 2.21 – Detected lower edge on B-scan

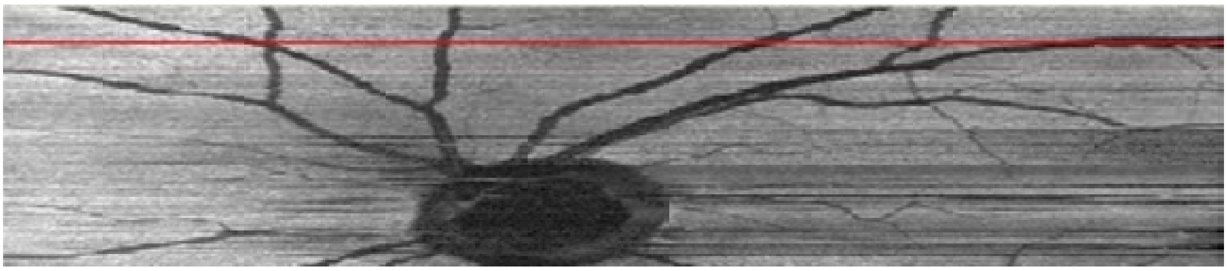

Fig. 2.22 – Reconstructed image from B-scans with marked the exact line as is the edge on the figure 2.21

The fundus image is pre-processed by function *enhacement2_normovany.m* which was created by Doc. Kolář [16] of Department of biomedical engineering of Faculty of electrical engineering and communication of Brno University of Technology. In this function, the contrast and illumination is adjusted. Then the image is cropped to represent exact area as the image with B-scans. The final step of fundus pre-processing is to undersample the fundus image.

2.5.2 B-scans and fundus matching

The registration is provided by correlation (see chapter 1.3.5). Each line of B-scans image is chosen and then is correlated with range of lines in fundus image (see figure 2.23). The range of searched area of lines can be manually set. The default setting is 15 lines, which corresponds with number of lines (rows) in the B-scans image divided by ten, thus the searched area represents one tenth of the image. After all lines are successfully detected, the fundus image is resampled to its original size.

Fig. 2.23 – Correlation of lines in B-scans image and undersampled fundus image

2.5.3 Matching result

The result is shown on the figure 2.24. Each B-scan is matched to its position on fundus image. Because the images not represent the exact area, there is a small error. The result on the whole fundus image can be seen in figure 2.25. The whole process in block diagram is figured in figure 2.26.

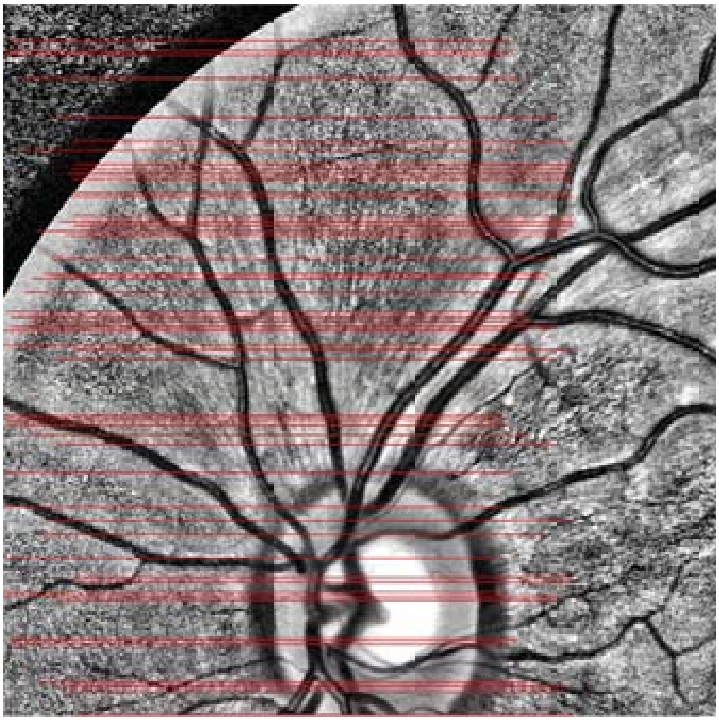

Fig. 2.24 – Registration result – detected B-scans in the fundus image

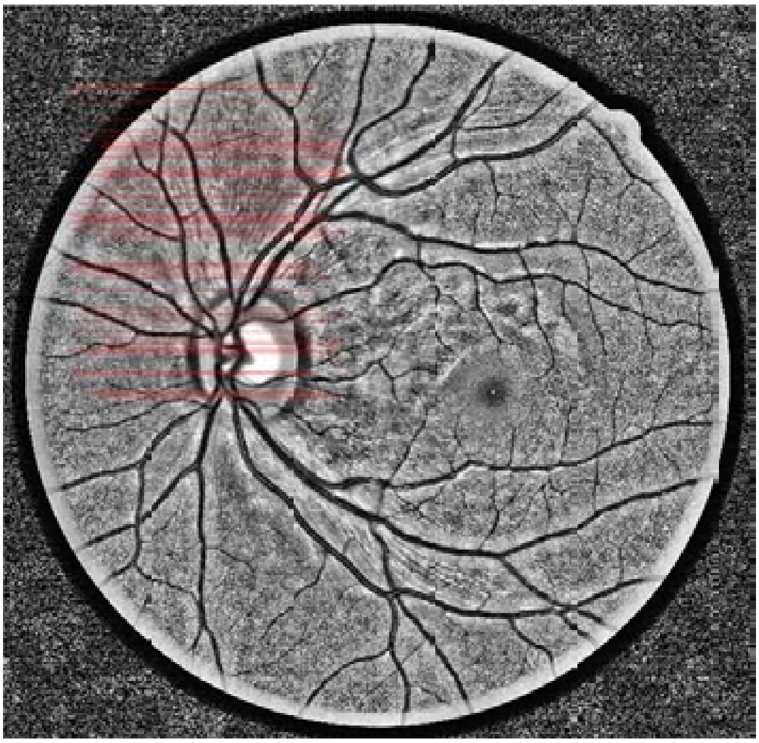

Fig. 2.25 – Registration result - detected B-scans in the whole fundus image

2.5.4 Block diagram of B-scans matching with fundus image

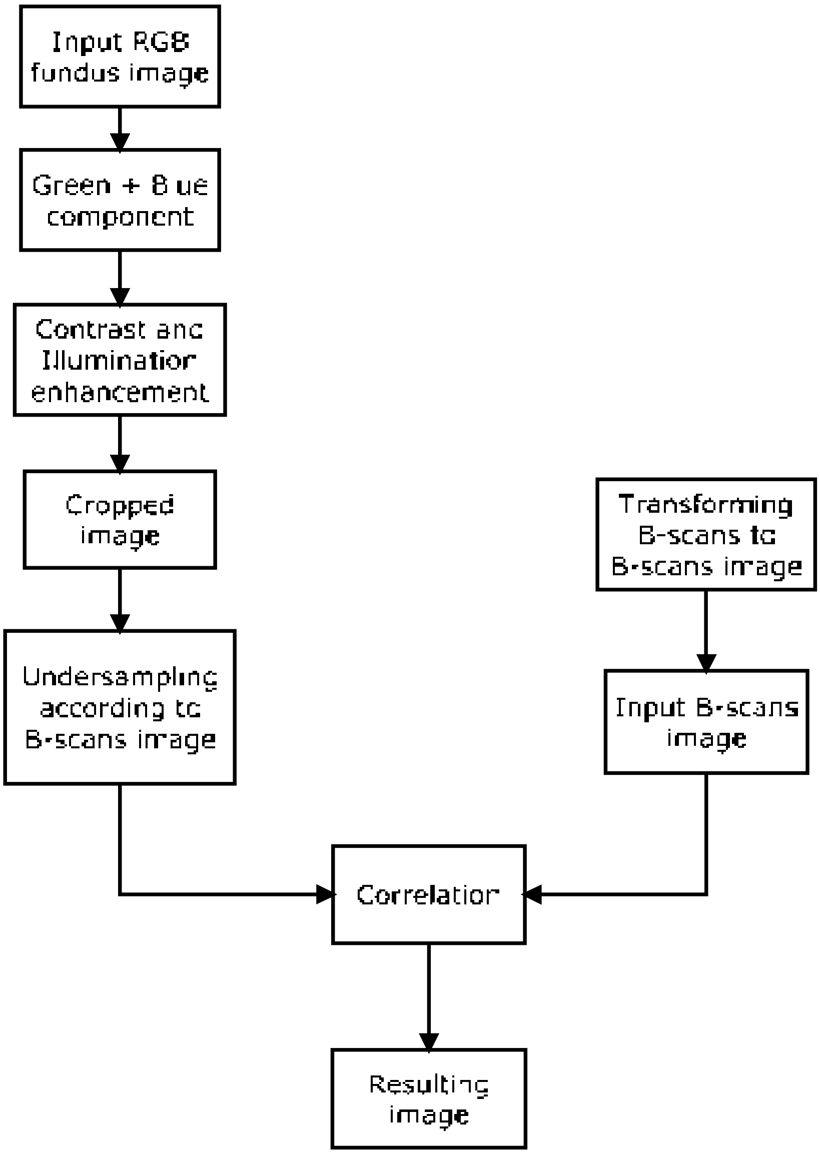
Input RGB fundus image
Green + Blue component
Contrast and Illumination enhancement
Cropped image
Undersampling according to B-scans image
Transforming B-scans to B-scans image
Input B-scans image
Correlation
Resulting image

Fig. 2.26 – Block diagram of matching B-scans to fundus image

CONCLUSION

This thesis is dealing with multimodal registration of retinal devices from different scanning devices. The most important task in this thesis was to create different types of algorithms which can be used for image registration of retina images. First type of registration – manual registration, is the simplest type of registration. It is also very accurate. But it requires the skilled user to control it, and it is not very fast. In manual registration the user has to choose exact same points in both images to be registered. The source code can be found in supplement C. Second type of registration is automatic registration. Automatic registration has big advantage, because it needs no user. It can work fully automatically. The base points are selected automatically then the correlation algorithm finds the exact matches. But automatic registration is not very good in very different images. If the images look very different than the previous images, the automatic registration probably will not work properly. So the user is forced to slightly adjust the thresholds, or areas for searching the base points. The possible procedure how to adjust the automatic registration algorithm if it is not working properly in the image is described in chapter 2.3.4. The default settings in the automatic registration algorithm was tested in all available data (images) and its evaluation is in table 1, supplement A. The source code can be found in supplement D. The third type of registration is combination of both previous method and it is called semi-automatic registration. The difference between automatic and semi-automatic registration is in selecting the base points. In semi-automatic registration the user has to select base points manually, and the input points in another image are found automatically by correlation algorithm. The source code can be found in supplement E. The final task was to create algorithm which can match B-Scans from OCT to image from fundus camera. The source code can be found in supplement F and the results of the matching are plotted in figures 2.24 and 2.25.

REFERENCES

[1] JAN, Jiří. *Medical Image Processing, Reconstruction and Restoration - Concepts and Methods*. CRC Tylor and Francis, New York 2005. ISBN 0-8247-5849-8.

[2] BRONZINO, Joseph. *The Biomedical Engineering Handbook.* 2nd edition, 2000. CRC Press LLC, Boca Raton 2000. ISBN 0-8493-0461-X.

[3] HONZÍKOVÁ, N., HONZÍK P. *Biologie člověka*. Skripta v elektronické podobě. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2003.

[4] DUBOVÝ, P. *Základy neuroanatomie a nervových drah.* Skripta v elektronické podobě. Brno: Masarykova univerzita, Lékařská fakulta, Anatomický ústav, 2007.

[5] SCHMITT, M. J. *Optical Coherence Tomography (OCT): A Review*. In *IEEE Journal of selected topics in quantum electronics July/August 1999.* Pleasanton. IEEE, 1999. Volume 5. s. 1205 – 1215.

[6] ZÍTOVÁ, B., FLUSSER, J. *Image registration methods: a survey.* In *Image and Vision Computing June 2003.* Prague. Academy of Sciences of the Czech Republic, 2003. s. 977 – 995.

[7] KOLÁŘ, R. *Zobrazovací systémy pro diagnostiku zadního segmentu oka* [počítačová prezentace]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2010 [citováno 2010-04-04]. Prezentace k předmětu Diagnostika bio- a ekosystémů.

[8] MathWorks Inc. MATLAB. *Matlab online help* [online]. c2008, last revision April 2011, [quoted 2011-04-12]. < http://www.mathworks.com/help/techdoc/index.html >.

[9] KORN, M. T., KORN G. A. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover publications, Mineola, NY 2000. ISBN 0-486-41147-8.

[10] GONZALO R. A. *Nonlinear Signal Processing: A Statistical Approach.* John Wiley & Sons, Hoboken, New Jersey, 2005. ISBN 0-471-67624-1.

[11] LATECKI J. L., BAI X., LIU W. Y. *Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution.* In *IEEE Transactions on pattern analysis and machine intelligence March 2007.* IEEE, 2007. Volume 29. s. 449 – 462.

[12] KOLÁŘ R. *Methods for image analysis and pattern recognition – Application to early glaucoma diagnosis.* Habilitation thesis. Brno, Brno University of Technology. Faculty of Electrical Engineering and Communication, 2008.

[13] WOJTKOWSKI M., SRINIVASAN V., KO T., FUJIMOTO J. *High Speed, Ultrahigh Resolution Retinal Imaging using Spectral/Fourier Domain OCT.* In *Conference on Lasers & Electro-Optics (CLEO) May 2005.* IEEE, 2005. s. 2058 – 2060.

[14] AITKEN V. C. *Retinal Motion and Microvascular Pulsation Measurements from SLO Image Sequences.* In *IMTC 2005 – Instrumentation and Measurement Technology Conference May 2005.* Ottawa. IEEE, 2005. s. 111 – 116.

[15] SHARP P. F., MANIVANNAN A. *The scanning laser ophthalmoscope.* In *Phys. Med. Biol. 42 1997.* University of Aberdeen, 1995. Volume 42. s. 951 – 966.

[16] KOLÁŘ R., TAŠEVSKÝ P. *Registration of 3D Retinal Optical Coherence Tomography Data and 2D Fundus Image.* Brno, Brno University of Technology. Faculty of Electrical Engineering and Communication.

SUPPLEMENTS

Content of supplements

A Evaluation of the automatic registration algorithm

Table 1 - Evaluation of automatic registration

Number of the image	Average deviation [%]	Maximal deviation [%]
31600	0,48	50,06
31604	1,56	57,58
31608	0,79	52,69
59042	12,67	77,11
59047	4,95	55,57
59048	1,31	70,38
59051	5,63	71,72
70872	3,73	98,67
87930	3,60	65,30
90790	0,76	53,58
91930	1,65	74,60
91931	1,57	59,64
92422	5,66	97,19
92423	2,76	63,99
97873	3,60	72,19
98410	0,90	56,53
98411	0,86	47,54

B Source code of function fce_cteni_fundus.m

```
function [obr_novy] = fce_cteni_fundus (path)

I = imread(path);

zmenseni = 4;

R=I(:,:,1); % converting image to R,G,B components
G=I(:,:,2);
B=I(:,:,3);
obr = G+B;
obr = double(obr);
obr = obr/255;
obr = imresize(obr, 1/zmenseni);

stred_x = length(obr(:,1))/2;
stred_y = length(obr(1,:))/2;

bod_y = 0;
bod_x = 0;
sw = 1; % switch for searching the middle of the eye

 for j = 1:length(obr(1,:))-1 % searching for the edge of the eye (horizontal)
     if ((obr(stred_x,j+1) - obr(stred_x,j)) > 0.05) && (sw == 1) % comparing
differences between pixels
          bod_y(1) = j;
          sw = 0;
     end

     if (obr(stred_x,j+1) - obr(stred_x,j)) < -0.09 % searching for the second
edge of the eye (horizontal)
          bod_y(2) = j;
     end
 end

 sw = 1;

 for i = 1:length(obr(:,1))-1 % searching for the edge of the eye (vertical)
     if ((obr(i+1,stred_y) - obr(i,stred_y)) > 0.05) && (sw == 1)
          bod_x(1) = i;
          sw = 0;
     end

     if (obr(i+1,stred_y) - obr(i,stred_y)) < -0.05
          bod_x(2) = i;
     end
 end


souradnice_hran(1,1:2) = [stred_x bod_y(1)]; % saving the coordinances
souradnice_hran(2,1:2) = [stred_x bod_y(2)];
souradnice_hran(3,1:2) = [bod_x(1) stred_y];
souradnice_hran(4,1:2) = [bod_x(2) stred_y];
```

```
obr_novy =
obr(min(souradnice_hran(:,1)):max(souradnice_hran(:,1)),min(souradnice_hran(:,
2)):max(souradnice_hran(:,2)));
obr_novy = obr_novy * 255;
lim1 = round(length(obr_novy(:,1))/6); % limitation of image
lim2 = round(length(obr_novy(:,1))/1.2);
obr_novy = obr_novy(lim1:lim2,lim1:lim2);
```

C Source code of manual registration man_reg.m

```
close all;
clear all;
clc;

pocet_ref_bodu = 4; % number of reference points
poz_vel = 1000; % desired size of the image

nereg = fce_cteni_fundus_man('data\98411.JPG');
zmenseni = length(nereg(1,:))/poz_vel;
nereg = imresize(nereg, 1/zmenseni);

vzor = imread('data\98411_slo.png');
vzor = double(vzor);
vzor = imresize(vzor, [poz_vel poz_vel]);

x1 = 0; y1 = 0;
x2 = 0; y2 = 0;

for i = 1:pocet_ref_bodu
figure(1)
subplot(1,2,1)
imshow(nereg,[]);

if i > 1 % plotting of chosen points
    plot(x1, y1, 'r*');
end

hold on;
title('Image to register - mark the input points');
[x1(i) y1(i)] = ginput(1); % getting of reference points
plot(x1(i), y1(i), 'r*'); % marking the reference points by "+"

subplot(1,2,2)
imshow(vzor,[]);

if i > 1
    plot(x2, y2, 'r*');
end

hold on;
title('Template image - mark the reference (base) points');
[x2(i) y2(i)] = ginput(1);
plot(x2(i), y2(i), 'r*');

end
hold off;

input_points(1:pocet_ref_bodu,1) = x1; % loading of chosen coordinates
input_points(1:pocet_ref_bodu,2) = y1;
base_points(1:pocet_ref_bodu,1) = x2;
base_points(1:pocet_ref_bodu,2) = y2;
```

```
transf = cp2tform(input_points,base_points,'affine'); % registration using
cp2tform function
reg = imtransform(nereg,transf,'XData',[1 poz_vel], 'YData',[1 poz_vel]);

[k l] = size(reg);
reg2 = zeros(poz_vel);

for i = 1:k
    for j = 1:l
        if reg(i,j) ~= 0
        reg2(i,j) = reg(i,j);  % cuttting the image
        end
    end
end

reg2 = imresize(reg2, [k l]);
souc = zeros(poz_vel);

for i = 1:length(souc(:,1))
    for j = 1:length(souc(1,:))
        if reg2(i,j) ~= 0   % connecting the template image to registered
image
           souc(i,j) = reg2(i,j);
        else
            souc(i,j) = reg2(i,j) + vzor(i,j);
        end
    end
end

figure(2);
subplot(1,2,1);
imshow(souc,[]);
title('Registration result - vessel connection');
subplot(1,2,2);
imshow(reg2+vzor/2,[]);
title('Registration result - vessel coverage');

sach = MakeChessBoard(souc, vzor, 150);
figure(3); imshow(sach,[]);
```

D Source code of automatic registration aut_reg.m

```
clear all;
close all;
clc;

des_size = 1000; % desired size of Fundus and OCT image
win_size = 100; % size of serching window in the image

%----------- reading ------------
fundus = fce_cteni_fundus('data\98411.JPG');
zmenseni = length(fundus(1,:))/des_size;
fundus = imresize(fundus, 1/zmenseni);
oct = imread('data\98411_slo.png');
oct = double(oct);
oct = imresize(oct,[des_size des_size]);

[X Y] = size(fundus);
step = 10;

figure(1)
subplot(1,2,1);
imshow(fundus,[]);
subplot(1,2,2);
imshow(oct,[]);
impixelinfo;
hold on;

sw2 = 1;
sw3 = 1;
sw4 = 1;

max_p = max(max(oct));

num_points = 4;
oct_points = zeros(num_points,2);

oct_grad = zeros(des_size);
oct_thresh = zeros(des_size);
oct_skelet = zeros(des_size);
hold off;

for j = 1:Y
    oct_grad(:,j) = gradient(oct(:,j));
end

prah = mean(min(oct_grad));

%----------setting the threshold according to max intensity  ------------
for i = 1:X
    for j = 1:Y

        if max_p > 254 && max_p < 255
            if oct_grad(i,j) <= -3
```

```
                oct_thresh(i,j) = oct_grad(i,j);
                oct_skelet(i,j) = 1;
            end
        end

        if max_p > 255 && max_p < 256
            if oct_grad(i,j) <= -1.5
               oct_thresh(i,j) = oct_grad(i,j);
               oct_skelet(i,j) = 1;
            end
        end

        if max_p > 256
            if oct_grad(i,j) <= -1
               oct_thresh(i,j) = oct_grad(i,j);
               oct_skelet(i,j) = 1;
            end
        end
    end
end

oct_skelet = bwmorph(oct_skelet,'skel',2);
oct_skelet = medfilt2(oct_skelet,[3 3]);

figure(102);
imshow(oct_skelet,[]);
hold on;

cnt = 0;
cnt1 = 0;
cnt2 = 0;
cnt3 = 0;
cnt4 = 0;

%-----------counting the points of interest in set range of coordinates----

for i = 100:200
    for j = 300:400
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+2,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1)))
            cnt1 = cnt1 + 1;
        end
    end
end

for i = 100:200
    for j = 450:550
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+1,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1)))
            cnt2 = cnt2 + 1;
        end
    end
end

for j = 400:500
```

```
    for i = 600:700
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+2,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1)))
            cnt3 = cnt3 + 1;
        end
    end
end

sw = 1;
for j = 600:700
    for i = 650:750
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+1,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1)))
            cnt4 = cnt4 + 1;
        end
    end
end

%----------selecting and plotting the middle POI in the searched area---------
------

for i = 100:200
    for j = 300:400
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+2,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1))) && sw == 1;
            cnt = cnt + 1;
            if cnt == round(cnt1/2)
                sw = 0;
                plot(j,i,'g+');
            end
            oct_points(1,2) = j-win_size/2;
            oct_points(1,1) = i-win_size/2;
        end
    end
end

cnt = 0;
sw = 1;
for i = 100:200
    for j = 450:550
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+1,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1))) && sw == 1;
            cnt = cnt + 1;
            if cnt == round(cnt2/2)
                sw = 0;
                plot(j,i,'g+');
            end
            oct_points(2,2) = j-win_size/2;
            oct_points(2,1) = i-win_size/2;
        end
    end
end

cnt = 0;
sw = 1;
for j = 400:500
```

```
    for i = 600:700
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+2,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1))) && sw == 1;
            cnt = cnt + 1;
            if cnt == round(cnt3/2)
                sw = 0;
                plot(j,i,'g+');
            end
            oct_points(3,2) = j-win_size/2;
            oct_points(3,1) = i-win_size/2;
        end
    end
end

cnt = 0;
sw = 1;
for j = 600:700
    for i = 650:750
        if (((oct_skelet(i,j) == 1) && (oct_skelet(i+1,j) == 1)) ||
((oct_skelet(i,j) == 1) && (oct_skelet(i,j+1) == 1))) && sw == 1;
            cnt = cnt + 1;
            if cnt == round(cnt4/2)
                sw = 0;
                plot(j,i,'g+');
            end
            oct_points(4,2) = j-win_size/2;
            oct_points(4,1) = i-win_size/2;
        end
    end
end

hold off;

sw = 1;
fundus_points = zeros(num_points,2);
max_cor = zeros(1,num_points);
win_size_mini = 20;
cor_coef_mini = zeros(win_size-win_size_mini,win_size-win_size_mini);
oct_points_mini = zeros(num_points,2);
fundus_points_mini = zeros(num_points,2);

for cyk = 1:num_points % correlation for finding the exact match

cor_coef = zeros(1,Y/10); % coeficients of correlation between Fundus and OCT
image
counter2 = 0;
counter = 1;

for i = 1:step:X-win_size
    for j = 1:step:Y-win_size
        if counter2 == Y/step;
            counter2 = 0;
            counter = counter+1;
        end
        counter2 = counter2 + 1;
        cor_coef(counter,counter2) =
corr2(oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1),fundus(i:i+win_size-
1,j:j+win_size-1));  % picking up the correlation coefficients
    end
end

counter2 = 0;
counter = 1;
max_cor(cyk) = max(max(cor_coef));

figure(2)
subplot(1,2,1);
imshow(fundus,[]);
title('Fundus');
hold on;

for i = 1:step:X-win_size
    for j = 1:step:Y-win_size
        if counter2 == Y/step;
            counter2 = 0;
            counter = counter+1;
        end
        counter2 = counter2 + 1;
        if cor_coef(counter,counter2) > max_cor(cyk) - 0.0001 % plotting of
most correlated image
        plot(j:j+win_size-1,i,'r-');
        plot(j,i:i+win_size-1,'r-');
        plot(j:j+win_size-1,i+win_size-1,'r-');
        plot(j+win_size,i:i+win_size-1,'r-');
        fundus_points(cyk,1) = i;
        fundus_points(cyk,2) = j;
        end
    end
end

hold off;
subplot(1,2,2);
imshow(oct,[]);
title('SLO');
hold on;

%------------------plotting of oct mini window--------------------------
plot(oct_points(cyk,2):oct_points(cyk,2)+win_size-1,oct_points(cyk,1),'r-');
plot(oct_points(cyk,2),oct_points(cyk,1):oct_points(cyk,1)+win_size-1,'r-');
plot(oct_points(cyk,2):oct_points(cyk,2)+win_size-
1,oct_points(cyk,1)+win_size-1,'r-');
plot(oct_points(cyk,2)+win_size,oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,'r-');

figure(3);
subplot(1,2,1);
imshow(fundus(fundus_points(cyk,1):fundus_points(cyk,1)+win_size,fundus_points
(cyk,2):fundus_points(cyk,2)+win_size),[]);
title('Fundus');
subplot(1,2,2);
```

```
imshow(oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1),[]);
title('SLO');
fundus_mini = fundus(fundus_points(cyk,1):fundus_points(cyk,1)+win_size-
1,fundus_points(cyk,2):fundus_points(cyk,2)+win_size-1);
oct_mini = oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1);

figure(4);
imshow(oct_mini,[]);
hold on;
min_oct1 = min(min(oct_mini(1:win_size-win_size_mini,1:win_size-
win_size_mini)));

%------------------searching of the minimum in oct mini window------------
for i = 1:win_size-win_size_mini
    for j = 1:win_size-win_size_mini
        if oct_mini(i,j) < min_oct1+0.001
           oct_points_mini(cyk,2) = j;
           oct_points_mini(cyk,1) = i;
           plot(j,i,'r+');
        end
    end
end

hold off;

%-----------------searching for correlation coefficients-------------------
for i = 1:win_size-win_size_mini
    for j = 1:win_size-win_size_mini
          cor_coef_mini(i,j) =
corr2(oct_mini(oct_points_mini(cyk,1):oct_points_mini(cyk,1)+win_size_mini-
1,oct_points_mini(cyk,2):oct_points_mini(cyk,2)+win_size_mini-
1),fundus_mini(i:i+win_size_mini-1,j:j+win_size_mini-1));
    end
end

max_cor2 = max(max(cor_coef_mini));


figure(5);
subplot(1,2,1);
imshow(fundus_mini,[]);
title('Fundus mini 20x20');
hold on;

for i = 1:win_size-win_size_mini
    for j = 1:win_size-win_size_mini
        if cor_coef_mini(i,j) > max_cor2 - 0.0001
        plot(j:j+win_size_mini-1,i,'r-');
        plot(j,i:i+win_size_mini-1,'r-');
        plot(j:j+win_size_mini-1,i+win_size_mini-1,'r-');
        plot(j+win_size_mini,i:i+win_size_mini-1,'r-');
        fundus_points_mini(cyk,1) = i;
        fundus_points_mini(cyk,2) = j;
        end
    end
end


hold off;

subplot(1,2,2);
imshow(oct_mini,[]);
title('SLO mini 20x20');
hold on;
%------------------plotting of oct mini mini window--------------------------
plot(oct_points_mini(cyk,2):oct_points_mini(cyk,2)+win_size_mini-
1,oct_points_mini(cyk,1),'r-');
plot(oct_points_mini(cyk,2),oct_points_mini(cyk,1):oct_points_mini(cyk,1)+win_
size_mini-1,'r-');
plot(oct_points_mini(cyk,2):oct_points_mini(cyk,2)+win_size_mini-
1,oct_points_mini(cyk,1)+win_size_mini-1,'r-');
plot(oct_points_mini(cyk,2)+win_size_mini,oct_points_mini(cyk,1):oct_points_mi
ni(cyk,1)+win_size_mini-1,'r-');
hold off;

end

input_points(:,1) = fundus_points(:,2) + fundus_points_mini(:,2);
input_points(:,2) = fundus_points(:,1)+ fundus_points_mini(:,1);
base_points(:,1) = oct_points(:,2) + oct_points_mini(:,2);
base_points(:,2) = oct_points(:,1) + oct_points_mini(:,1);


transf = cp2tform(input_points,base_points,'affine'); % image spatial
transform using cp2tform function
reg = imtransform(fundus,transf,'XData',[1 des_size], 'YData',[1 des_size]);


[k l] = size(reg);
reg2 = zeros(des_size);

for i = 1:k
    for j = 1:l
        if reg(i,j) ~= 0
        reg2(i,j) = reg(i,j);  % cuttting the image
        end
    end
end

reg2 = imresize(reg2, [k l]);
souc = zeros(des_size);

for i = 1:length(souc(:,1))
    for j = 1:length(souc(1,:))
        if reg2(i,j) ~= 0   % connecting the template image to registered
image
           souc(i,j) = reg2(i,j);
        else
            souc(i,j) = reg2(i,j) + oct(i,j);
        end
    end
end

figure(6);
subplot(1,2,1);
imshow(souc,[]);
title('Registration result - vessel connection');
subplot(1,2,2);
imshow(reg2+oct/2,[]);
title('Registration result - vessel coverage');

sach = MakeChessBoard(souc, oct, 150);
figure(7); imshow(sach,[]);
```

E Source code of semi-automatic registration semi_reg.m

```
clear all;
close all;
clc;

des_size = 1000; % desired size of Fundus and OCT image
win_size = 100; % size of serching window in the image

%----------- reading ------------
fundus = fce_cteni_fundus('data\31600.JPG');
zmenseni = length(fundus(1,:))/des_size;
fundus = imresize(fundus, 1/zmenseni);
oct = imread('data\31600_slo.png');
oct = double(oct);
oct = imresize(oct,[des_size des_size]);

[X Y] = size(fundus);
step = 10;

sw2 = 1;
sw3 = 1;
sw4 = 1;

max_p = max(max(oct));
num_points = 4;
oct_points = zeros(num_points,2);

figure(1)
subplot(1,2,1)
imshow(fundus,[]);
title('Image to register - do not mark points in this image')

for i = 1:num_points
subplot(1,2,2)
imshow(oct,[]);
hold on;

if i > 1 % plotting of chosen points
    plot(x1, y1, 'r*');
end

hold on;
title('Template image - mark the reference (base) points');
[x1(i) y1(i)] = ginput(1); % getting of reference points
plot(x1(i), y1(i), 'r*'); % marking the reference points by "+"

oct_points(i,1) = y1(i)-win_size/2;
oct_points(i,2) = x1(i)-win_size/2;

end
hold off;
```

```
sw = 1;
fundus_points = zeros(num_points,2);
max_cor = zeros(1,num_points);
win_size_mini = 20;
cor_coef_mini = zeros(win_size-win_size_mini,win_size-win_size_mini);
oct_points_mini = zeros(num_points,2);
fundus_points_mini = zeros(num_points,2);

for cyk = 1:num_points

cor_coef = zeros(1,Y/10); % coeficients of correlation between Fundus and OCT
image
counter2 = 0;
counter = 1;

for i = 1:step:X-win_size
    for j = 1:step:Y-win_size
        if counter2 == Y/step;
            counter2 = 0;
            counter = counter+1;
        end
        counter2 = counter2 + 1;
        cor_coef(counter,counter2) =
corr2(oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1),fundus(i:i+win_size-
1,j:j+win_size-1));  % picking up the correlation coefficients
    end
end

counter2 = 0;
counter = 1;
max_cor(cyk) = max(max(cor_coef));

figure(2)
subplot(1,2,1);
imshow(fundus,[]);
title('Fundus');
hold on;

for i = 1:step:X-win_size
    for j = 1:step:Y-win_size
        if counter2 == Y/step;
            counter2 = 0;
            counter = counter+1;
        end
        counter2 = counter2 + 1;
        if cor_coef(counter,counter2) > max_cor(cyk) - 0.0001 % plotting of
most correlated image
        plot(j:j+win_size-1,i,'r-');
        plot(j,i:i+win_size-1,'r-');
        plot(j:j+win_size-1,i+win_size-1,'r-');
        plot(j+win_size,i:i+win_size-1,'r-');
        fundus_points(cyk,1) = i;
        fundus_points(cyk,2) = j;
        end
    end
```

```
end

hold off;
subplot(1,2,2);
imshow(oct,[]);
title('SLO');
hold on;

%------------------plotting of oct mini window--------------------------
plot(oct_points(cyk,2):oct_points(cyk,2)+win_size-1,oct_points(cyk,1),'r-');
plot(oct_points(cyk,2),oct_points(cyk,1):oct_points(cyk,1)+win_size-1,'r-');
plot(oct_points(cyk,2):oct_points(cyk,2)+win_size-
1,oct_points(cyk,1)+win_size-1,'r-');
plot(oct_points(cyk,2)+win_size,oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,'r-');

figure(3);
subplot(1,2,1);
imshow(fundus(fundus_points(cyk,1):fundus_points(cyk,1)+win_size,fundus_points
(cyk,2):fundus_points(cyk,2)+win_size),[]);
title('Fundus');
subplot(1,2,2);

imshow(oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1),[]);
title('SLO');

fundus_mini = fundus(fundus_points(cyk,1):fundus_points(cyk,1)+win_size-
1,fundus_points(cyk,2):fundus_points(cyk,2)+win_size-1);
oct_mini = oct(oct_points(cyk,1):oct_points(cyk,1)+win_size-
1,oct_points(cyk,2):oct_points(cyk,2)+win_size-1);

figure(4);
imshow(oct_mini,[]);
hold on;
min_oct1 = min(min(oct_mini(1:win_size-win_size_mini,1:win_size-
win_size_mini)));

%------------------searching of the minimum in oct mini window------------
for i = 1:win_size-win_size_mini
    for j = 1:win_size-win_size_mini
        if oct_mini(i,j) < min_oct1+0.001
           oct_points_mini(cyk,2) = j;
           oct_points_mini(cyk,1) = i;
           plot(j,i,'r+');
        end
    end
end

hold off;

%-----------------searching for correlation coefficients-------------------
for i = 1:win_size-win_size_mini
    for j = 1:win_size-win_size_mini
```

```
          cor_coef_mini(i,j) = 
corr2(oct_mini(oct_points_mini(cyk,1):oct_points_mini(cyk,1)+win_size_mini-
1,oct_points_mini(cyk,2):oct_points_mini(cyk,2)+win_size_mini-
1),fundus_mini(i:i+win_size_mini-1,j:j+win_size_mini-1));
    end
end

max_cor2 = max(max(cor_coef_mini));



figure(5);
subplot(1,2,1);
imshow(fundus_mini,[]);
title('Fundus mini 20x20');
hold on;

for i = 1:win_size-win_size_mini
    for j = 1:win_size-win_size_mini
        if cor_coef_mini(i,j) > max_cor2 - 0.0001
        plot(j:j+win_size_mini-1,i,'r-');
        plot(j,i:i+win_size_mini-1,'r-');
        plot(j:j+win_size_mini-1,i+win_size_mini-1,'r-');
        plot(j+win_size_mini,i:i+win_size_mini-1,'r-');
        fundus_points_mini(cyk,1) = i;
        fundus_points_mini(cyk,2) = j;
        end
    end
end



hold off;

subplot(1,2,2);
imshow(oct_mini,[]);
title('SLO mini 20x20');
hold on;
%------------------plotting of oct mini mini window--------------------------
plot(oct_points_mini(cyk,2):oct_points_mini(cyk,2)+win_size_mini-
1,oct_points_mini(cyk,1),'r-');
plot(oct_points_mini(cyk,2),oct_points_mini(cyk,1):oct_points_mini(cyk,1)+win_
size_mini-1,'r-');
plot(oct_points_mini(cyk,2):oct_points_mini(cyk,2)+win_size_mini-
1,oct_points_mini(cyk,1)+win_size_mini-1,'r-');
plot(oct_points_mini(cyk,2)+win_size_mini,oct_points_mini(cyk,1):oct_points_mi
ni(cyk,1)+win_size_mini-1,'r-');
hold off;

end

input_points(:,1) = fundus_points(:,2) + fundus_points_mini(:,2);
input_points(:,2) = fundus_points(:,1)+ fundus_points_mini(:,1);
base_points(:,1) = oct_points(:,2) + oct_points_mini(:,2);
base_points(:,2) = oct_points(:,1) + oct_points_mini(:,1);
```

```
transf = cp2tform(input_points,base_points,'affine'); % image registration
using cp2tform function
reg = imtransform(fundus,transf,'XData',[1 des_size], 'YData',[1 des_size]);

[k l] = size(reg);
reg2 = zeros(des_size);

for i = 1:k
    for j = 1:l
        if reg(i,j) ~= 0
        reg2(i,j) = reg(i,j);  % cuttting the image
        end
    end
end

reg2 = imresize(reg2, [k l]);
souc = zeros(des_size);

for i = 1:length(souc(:,1))
    for j = 1:length(souc(1,:))
        if reg2(i,j) ~= 0   % connecting the template image to registered
image
           souc(i,j) = reg2(i,j);
        else
            souc(i,j) = reg2(i,j) + oct(i,j);
        end
    end
end

figure(6);
subplot(1,2,1);
imshow(souc,[]);
title('Registration result - vessel connection');
subplot(1,2,2);
imshow(reg2+oct/2,[]);
title('Registration result - vessel coverage');

sach = MakeChessBoard(souc, oct, 150);
figure(7); imshow(sach,[]);
```

F Source code of matching the B-Scans to fundus image

```
clc
close all;
clear all;

load data_bscan
bscan = final(:,28:727);
l_b = length(bscan(1,:));
r_b = length(bscan(:,1));
k = 1000/r_b;
bscan_win = [1 l_b];
fundus = imread('3org_Obr50.jpeg');
fundus = fundus(118:717,428:1327);
fundus_velky = imresize(fundus,[1000 1000]);
fundus = imresize(fundus,[1000/k 1000]);
cor_coef_bscan = zeros(1,100);

fundus = medfilt2(fundus, [5 5]);
[X Y] = size(bscan);
[X_f Y_f] = size(fundus);
fundus_rows = zeros(X,2);

c = 0;

for int = 6:r_b-10
    counter = 1;
    counter2 = 0;
for i = int-5:int+10
    for j = 1:100
        if counter2 == 100
            counter = counter + 1;
            counter2 = 0;
            c = c+1;
        end
        counter2 = counter2+1;
        cor_coef_bscan(counter,counter2) = corr2(bscan(int,:),fundus(i,j:j+Y-
1));
    end
end

max_bscan = max(max(cor_coef_bscan));


figure(2);
subplot(1,2,1);
imshow(bscan,[]);

hold on;

plot(1:1+bscan_win(2)-1,int,'r-');
plot(1,int:int+bscan_win(1)-1,'r-');
plot(1:1+bscan_win(2)-1,int+bscan_win(1)-1,'r-');
plot(1+bscan_win(2),int:int+bscan_win(1)-1,'r-');
```

```
hold off;

subplot(1,2,2);
imshow(fundus,[]);
hold on;

if int == 1
    fun_reg_points = zeros(1,2);
end


for i = 1:length(cor_coef_bscan(:,1))
    for j = 1:length(cor_coef_bscan(1,:))
        if cor_coef_bscan(i,j) == max_bscan
            plot(j:j+bscan_win(2)-1,i+int-1,'r-');
            plot(j,i+int-1:i+int-1+bscan_win(1)-1,'r-');
            plot(j:j+bscan_win(2)-1,i+int-1+bscan_win(1)-1,'r-');
            plot(j+bscan_win(2),i+int-1:i+int-1+bscan_win(1)-1,'r-');
            fundus_rows(int,1) = i+int;
            fundus_rows(int,2) = j;
        end
    end
end

fundus_rows(int,1) = fundus_rows(int,1).*k;
hold off;
figure(3);
imshow(fundus_velky,[]);
hold on;
plot(fundus_rows(int,2):fundus_rows(int,2)+bscan_win(2),fundus_rows(int,1),'r-
');
hold off;



end

figure(4);
imshow(fundus_velky,[]);
hold on;
for i = 1:X
    plot(fundus_rows(i,2):fundus_rows(i,2)+bscan_win(2),fundus_rows(i,1),'r-
');
end

orig = imread('3org_Obr50.jpeg');
figure(5);
imshow(orig,[]);
```