



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

LOKALIZACE BIFURKACÍ VE SNÍMCÍCH SÍTNICE
BIFURCATION LOCALIZATION IN RETINA IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN PRES

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LUKÁŠ SEMERÁD

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Pres Martin, Bc.**

Obor: Bioinformatika a biocomputing

Téma: **Lokalizace bifurkací ve snímcích sítnice**
Bifurcation Localization in Retina Images

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s postupem vyhledávání bifurkací ve snímcích sítnic.
2. Navrhněte použití algoritmu pro zvýraznění cév ve snímcích sítnice.
3. Vytvořte aplikaci pro automatizovanou grafickou úpravu snímků sítnice.
4. Do aplikace přidejte funkci vyhledání křížení cév na snímcích a ukládání jejich poloh.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti dalšího vylepšení.

Literatura:

- Dražanský M., Orság F., Doležel M. et al.: *Biometrie*, Brno, CZ, Computer Press, 2011, s. 294, ISBN 978-80-254-8979-6.

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Semerád Lukáš, Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Mezi hlavní prvky sítnice z biometrického hlediska patří slepá skvrna, žlutá skvrna a rozložení cév v sítnici. Větvení cév je pro každého jedince unikátní a právě tato vlastnost se využívá v biometrických systémech pro rozpoznávání osob podle obrazu sítnice. Tento dokument popisuje metodu pro lokalizaci slepé a žluté skvrny, dále se věnuje metodě pro zvýraznění krevních cév, založené na známé metodě *Matched filters*, a nakonec rozebírá lokalizaci bifurkací v extrahovaném krevním řečišti. Hlavním cílem této práce je vytvořit aplikaci pro automatizovanou úpravu snímků sítnice, segmentaci krevního řečiště a lokalizaci markantů. Program je implementován v jazyce Java s využitím knihovny OpenCV.

Abstract

From biometrical point of view, main features of retina are fovea, optic nerve and blood vessel tree. Blood vessel tree is unique for each person and this biological feature is used in biometric systems for person-recognition by retinal images. This document describes methods for optic disc and fovea localization, method for vessel tree segmentation, which is based on well-known *Matched filters* method and also describes method for localization of blood vessel bifurcations. Main goal of this thesis is creation of program which can automatically preprocess input image, segment blood vessels and localize vessel bifurcations. The program is implemented in Java with OpenCV library.

Klíčová slova

Sítnice, snímky sítnice, markant, bifurkace, krevní řečiště, extrakce, matched filters, CLAHE, Java, OpenCV.

Keywords

Retina, fundus images, feature, bifurcation, blood vessels, extraction, matched filters, CLAHE, Java, OpenCV.

Citace

PRES, Martin. *Lokalizace bifurkací ve snímcích sítnice*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Semerád Lukáš.

Lokalizace bifurkací ve snímcích sítnice

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana Ing. Lukáše Semeráda. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Pres
24. května 2016

Poděkování

Rád bych poděkoval Ing. Lukáši Semerádovi za cenné rady, za vstřícnost při konzultacích a za poskytnutí databází a manuálně označených bifurkací pro testovací účely.

© Martin Pres, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Analýza problému	4
2.1 Anatomie sítnice lidského oka	4
2.2 Pořizování snímků sítnice	5
2.3 Hledání bifurkací	5
2.4 Využití v biometrii	6
2.5 Krátký přehled existujících metod	6
3 Stanovení souřadného systému	8
3.1 Lokalizace slepé skvrny	8
3.2 Lokalizace žluté skvrny	10
4 Zvýraznění cév	12
4.1 Předzpracování obrazu	12
4.1.1 Automatické zvýraznění cév a utlumení pozadí	12
4.1.2 Odstranění odlesku	14
4.2 Continuation Algorithm	14
4.2.1 Matched filters	14
4.2.2 Adaptivní ekvalizace histogramu	15
4.2.3 Spojení obou obrazů	16
4.3 Prahování	18
5 Lokalizace markantů	19
5.1 Ztenčení cévního řečiště	19
5.2 Lokalizace bifurkací	20
6 Implementace	24
7 Výsledky testů	25
7.1 Úspěšnost hledání slepé a žluté skvrny	25
7.2 Úspěšnost lokalizace bifurkací	27
7.2.1 Výsledky pro databázi Gold	28
7.3 Úspěšnost extrakce cév	30
8 Závěr	32
Literatura	33

Přílohy	36
Seznam příloh	37
A Obsah CD	38
B Manual	39
B.1 Zprovoznění programu	39
B.2 Použití programu pro detekci bifurkací	39
B.3 Použití programu pro porovnávání manuálního a automatického označení bifurkací	40
B.4 Použití programu pro porovnání manuálně a automaticky segmentovaného krevního řečiště	40
C Výsledky testů	41

Kapitola 1

Úvod

Sítnice lidského oka je ideální prostředek pro rozpoznávání osob, neboť je u každého oka každého jedince zcela unikátní. Liší se u jednovaječných dvojčat a neshodují se ani sítnice v levém a pravém oku stejného jedince. Sítnice je také po celý život jedince stabilní, tedy nemění se v čase (v případě, že je oko zdravé)[10]. Ze sítnice se dají extrahovat rysy (*markanty*) v podobě křížení, resp. větvení cév (tzv. *bifurkací*), které lze pozorovat neinvazivně pomocí speciálních přístrojů. Získané snímky je nutné předzpracovat, obvykle převedením do odstínů šedi a zvýrazněním detailů. Dále následuje vyhledání slepé skvrny a žluté skvrny. Poté se ze snímku extrahuje cévní řečiště, ve kterém se po skeletonizaci lokalizují bifurkace (markanty).

V této práci bude zejména kladen důraz na zvýraznění cév a extrakci bifurkací. Nejdříve se zaměřím na celkové analyzování problému. Pokusím rozebrat postup a úskalí, která mne mohou potkat při řešení a představím některé existující metody, které se tímto problémem zabývají. Z některých představených metod budu vycházet. Ve třetí kapitole pak postupně popíšu, jakým způsobem určíme souřadný systém pro uchování bifurkací a jakým způsobem nalezneme slepou a žlutou skvrnu. Ve čtvrté kapitole se pak zaměřím na předzpracování obrazu tak, aby v něm bylo možné segmentovat krevní řečiště od pozadí snímku a dále se budu věnovat samotné metodě pro segmentování cév ve snímku. Pátá kapitola bude věnována postupu lokalizace bifurkací v segmentovaném krevním řečišti. V této kapitole rozeberu ztenčování cév v černobílém obrazu a dále popíšu jakým způsobem v těchto ztenčených cévách lokalizujeme body větvení. Šestá kapitola se pak bude krátce zabývat implementací a sedmá kapitola bude obsahovat popis testů a jejich výsledky. V závěrečné kapitole pak zhodnotím výsledky a navrhnou postup dalšího zlepšení implementovaných metod.

Kapitola 2

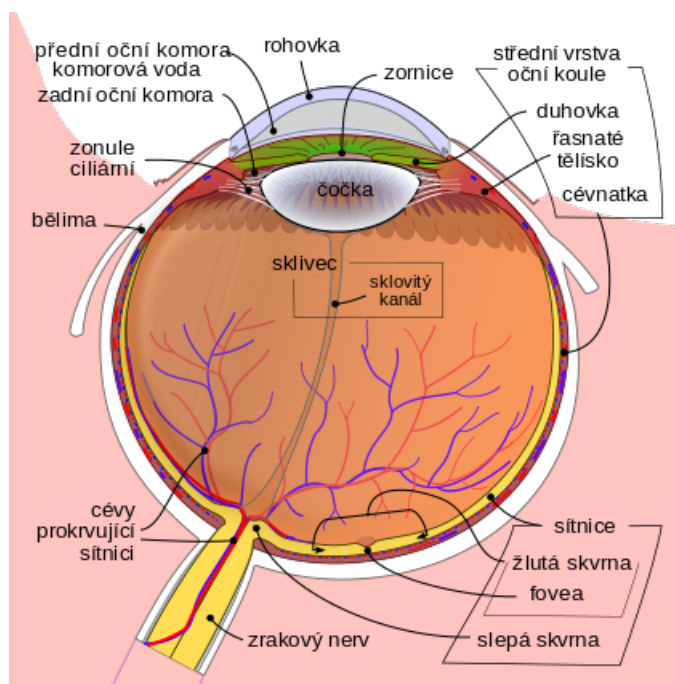
Analýza problému

2.1 Anatomie sítnice lidského oka

Sítnice (lat. retina) je vnitřní tenká světlocitlivá vrstva oka. Její hlavní funkcí je snímání světelných signálů, které na ni dopadají skrze čočku. Působením světelných paprsků vznikají ve světlocitlivých buňkách (fotoreceptory) chemické reakce, které následně vedou ke vzniku nervových impulzů přenášených do centrálního nervového systému.

Nejvýraznější objekty na sítnici jsou *slepá skvrna* a *žlutá skvrna*. Slepá skvrna (též optický disk) je místo, kudy vystupuje zrakový nerv a vstupují krevní cévy, které se dále větví a vyživují sítnici. V tomto místě se nenacházejí žádné fotoreceptory, nevzniká zde tedy žádná obrazová informace (proto slepá skvrna). Žlutá skvrna je místo nejostřejšího vidění a nachází se v tzv. centrální jamce uprostřed oblasti zvané *macula* [3].

Sítnice má vrstvou strukturu, přičemž vrstev je celkem deset a z toho jsou hlavní dvě vrstvy. V první hlavní vrstvě (počítáno zevnitř ven) se nachází buňky s velkým obsahem pigmentu, jejichž úkolem je pohlcovat světlo, které již bylo zaznamenáno fotoreceptory a zabránit jeho zpětnému odrazu čímž zvyšují ostrost vidění. Ve druhé vrstvě se nachází světlocitlivé buňky dvojího typu. *Tyčinky* reagují na světlo (i na slabé), ale nereagují na jeho barvu a neposkytují příliš ostrý obraz. Na sítnici se jich nachází kolem 130 milionů. *Čípky* reagují na barvu a poskytují ostrý obraz, ale jsou aktivní jen při dobrém osvětlení. Na lidské sítnici se nachází přibližně 7 milionů čípků a jejich největší koncentrace je ve žluté skvrně [4].



Obrázek 2.1: Průřez lidským okem (zdroj [1]).

2.2 Pořizování snímků sítnice

Sítnici lze fotografovat přímo pomocí přístroje zvaného *fundus camera*. Tento přístroj se používá v lékařství pro vyšetření oka. Pacient je obvykle usazen před přístroj, bradu a čelo má opřeny o speciální konstrukci a jedno oko má přiloženo k okuláru. Pomocí světelného bodu, který pacient vidí a na který má upřený pohled zkoumaného oka, může doktor nastavit pacientovo oko do požadované pozice. Poté může dojít k pořízení snímku.

Samotné zařízení tvoří specializovaný mikroskop napojený na kameru. Kamery v takovýchto zařízeních se odlišují úhlem, který jsou schopny zaznamenat. Rozsah záběru se obvykle pohybuje kolem 30° . Širokoúhlé kamery pořizují snímky v rozsahu 45° až 140° [2].

2.3 Hledání bifurkací

Celý proces lokalizace bifurkací spočívá v nalezení bodů, ve kterých se krevní cévy v sítnici větví nebo kříží. Tyto body jsou určeny podle souřadného systému (viz kapitola 3). Postup je rozdělen do následujících kroků:

1. Převod snímku do šedotónového obrazu.
2. Lokalizace slepé skvrny (optického disku).
3. Lokalizace žluté skvrny.
4. Extrakce krevního řečiště.
5. Skeletonizace extrahovaných cév.
6. Nalezení bodů křížení, resp. větvení.

Nalezení slepé a žluté skvrny je velmi důležité, jelikož podle jejich vzájemné polohy je určen souřadný systém, do nějž bude zanesena pozice nalezených markantů. Tato pozice bývá určena podle vzdálenosti bodu od počátku souřadného systému a podle úhlu, který tento bod svírá s přímkou, která je dána slepou a žlutou skvrnou. Tento systém bývá invariantní vůči rotacím, posunům a změnám velikosti, ke kterým může dojít při sejmutí vstupního obrazu.

V průběhu extrakce markantů lze očekávat několik problémů. V případě snímků zdravých sítnic se může v obraze vyskytnout nedostatečný kontrast slepé skvrny nebo cév vůči pozadí. Ze snímku sítnice se používá hlavně zelený kanál, protože právě ten má nejlepší kontrast. Obvykle se provádí také úpravy pro zvýraznění detailů v obraze a pro odstranění šumu.

Je nutné počítat i se sítnicemi, které jsou poškozené v důsledku nějaké choroby, jako je např. diabetická retinopatie. V případě takových sítnic se mohou ve snímku vyskytnout fragmenty, které mohou mít barvu podobnou krevním cévám a mohou být v průběhu segmentace mylně označeny jako cévy. Dále může být v některých případech pozadí sítnice velmi nehomogenní a může nastat problém s rozpoznáním toho, který útvar je nebo není céva. Nehomogenní pozadí může ovlivnit i detekci slepé a žluté skvrny. Objevily se snímky, kde jsem měl sám problém určit pozici žluté skvrny. V takových případech očekávám selhání mnou implementovaných metod.

2.4 Využití v biometrii

Jak již bylo naznačeno, lokalizace bifurkací má uplatnění zejména v biometrii. Ze snímku sítnice lze vytvořit biometrické šablony a tyto šablony mezi sebou porovnávat pro identifikaci osoby. Tento přístup má nespornou výhodu v tom, že jedinou pravou sítnicí může disponovat pouze její majitel. Sítnice se nedá odcizit bez vědomí majitele a odcizené oko při nasnímání neprojde detekcí živosti, která nemusí být příliš komplikovaná (např. požádáme uživatele, aby se zaměřil pohledem na jiný bod nebo aby mrknul). Vyrobit falzifikát sítnice podle předlohy tak, aby prošel biometrickým systémem je také velmi obtížné.

Nevýhodou přístupového systému založeného na sítnici je vysoká cena a nízká akceptace mezi uživateli, kteří se často bojí poškození oka. Další nevýhodou je, že sítnice obsahuje mnoho informací o zdravotním stavu uživatele.

Obecně je vhodné používat systémy založené na sítnici lidského oka tam, kde jsou vysoké nároky na bezpečnost. První funkční systém, který pracuje se sítnicí jako s biometrií byl vytvořen v roce 1975 firmou *EyeDentify*.

2.5 Krátký přehled existujících metod

Zejména metod pro segmentaci krevního řečiště je celá řada, protože jeho odhadnutí má význam z medicínského hlediska, zejména kvůli diagnostice různých poruch a nemocí oka. Nutno poznamenat, že všechny metody vyžadují úpravu kontrastu tak, aby cévy lépe vystupovaly nad pozadím a aby se v ideálním případě odstranil šum. Vzhledem k velkému počtu metod se zde budu věnovat pouze některým zajímavým metodám. Většina metod dosahuje přesnosti segmentace kolem 90%, metody prezentované [14] a [5] dokonce kolem 99% a 97%.

I přes to, že pracujeme se zeleným kanálem obrazu, mohou mít cévy na některých snímcích malý kontrast. V těchto případech není příliš efektivní použít běžně používané metody

pro ekvalizaci histogramu, protože pak dojde ke zvýraznění detailů v pozadí a to by zkomplikovalo další postup, neboť v některých případech je obtížné rozeznat fragmenty v pozadí od krevních cév. Zajímavý přístup je aplikace Gausovského rozostření na vstupní obraz a poté vzít minimum z původního a rozostřeného obrazu [14]. Dobrou volbou také může být metoda *(Fast) Gray Level Grouping* [7][8] nebo modifikovaná *Top-hat transformace* [13].

Nejjednodušší metodou segmentace cév je prosté aplikování morfologických operací otevření a uzavření následované prahováním. Další jednoduchý přístup je aplikování detektoru hran. Pokročilejší metodou je metoda *Matched filters* [6]. Tato metoda využívá znalosti rozložení intenzity v příčném řezu cévy. Jejím základem je konvoluce s gausovským filtrem, který se na obraz aplikuje několikrát a vždy o několik stupňů pootočený. Tímto lze získat odezvy z několika směrů, které po sloučení a prahování dají relativně dobře segmentované krevní řečiště. Tato metoda je hojně používaná a několik prací se snaží o její vylepšení, např. [22] využívá pro generování filtru první derivaci Gaussovy funkce (FDOG).

Často je možné se setkat se segmentací pomocí metody *Fuzzy C-Means* (FCM), která se snaží rozdělit pixely do shluků, přičemž každý pixel může patřit do několika shluků s různou mírou příslušnosti ke shluku [20]. Velmi zajímavá metoda je spojení *Support Vector Machine* (SVM), fuzzy clusteringu a neuronové sítě [5], která dokáže rozhodnout zda pixel patří nebo nepatří do cévy podle histogramu z okolí zkoumaného bodu.

Kapitola 3

Stanovení souřadného systému

3.1 Lokalizace slepé skvrny

Nejdříve je nutné lokalizovat místo, kde by se mohla slepá skvrna nacházet. K tomu poslouží jednoduchý výběr nejsvětějšího pixelu v obraze. Předtím je vhodné na obraz aplikovat Gaussovské rozostření kvůli eliminování malých a světlých bodů, které se mohou vyskytnout na některých snímcích. V některých případech však tento krok může způsobit, že okraj snímku bude světlejší než slepá skvrna a to hlavně u nekvalitních snímků. Z toho důvodu je dobré na obraz aplikovat filtr dolní propusti, díky kterému dojde k úpravě intenzity podle okolí a tím i ke zvýraznění pixelů, jejichž okolí je světlé, což je případ právě slepé skvrny. Výšku a šířku oblasti stanovíme jako třetinu výšky a šířky celého obrazu a v případě, že následující metoda nebude úspěšná, dojde ke zmenšení oblasti na čtvrtinu, potom na pětinu, atd.

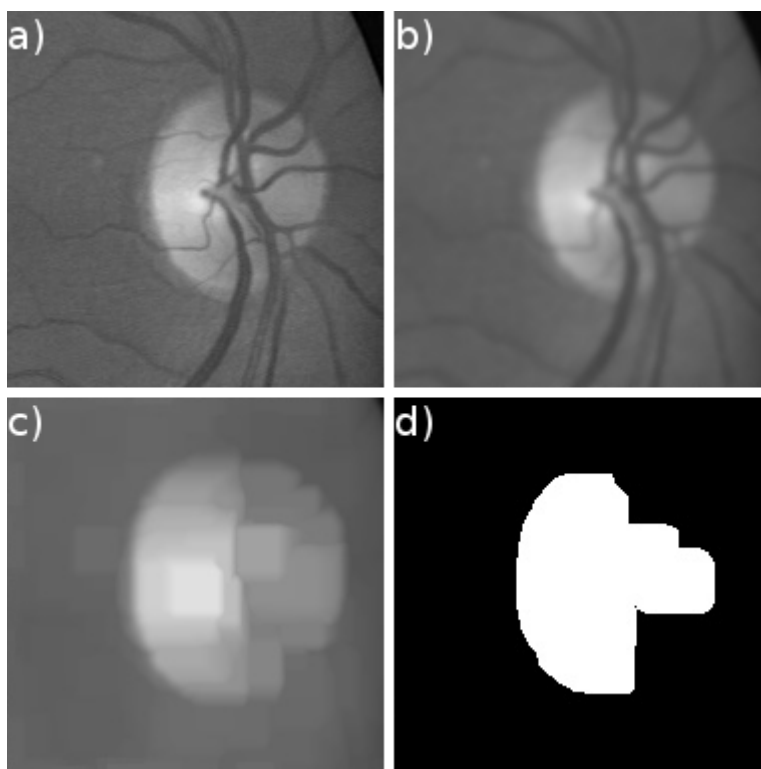
Máme-li určen pixel, který s největší pravděpodobností patří do oblasti se slepou skvrnou, určíme o které oko se jedná podle souřadnice x tohoto bodu. Pokud bude její hodnota menší, než je polovina sloupců obrazové matice, bude se jednat o levé oko. Zda je oko levé nebo pravé zde tedy určíme podle pozice slepé skvrny vzhledem k pozorovateli. Dále můžeme přikročit k určení přesnějšího obrysu slepé skvrny. Obrys bude dán kružnicí, jejíž střed bude jedním ze dvou bodů, které určí souřadný systém. Budeme částečně vycházet z postupu, který je popsán v [11]:

1. Na obraz aplikujeme Gaussovské rozostření nebo mediánový filtr a poté operaci dilatace. Tím z obrazu odstraníme krevní řečiště.
2. Aplikujeme prahování. Tím získáme jednu a více souvislých bílých oblastí na černém pozadí.
3. Ve výřezu vyhledáme aktivní kontury. Tím zjistíme přesný počet bílých oblastí. Odstraníme ty kontury, které jsou menší než zvolená nejmenší velikost a dále odstraníme kontury, které se dotýkají hranic výřezu (pokud zde není jen jediná kontura – v takovém případě ji ponecháme). Zbylé kontury sloučíme do jediné kontury.
4. Určíme nejmenší kružnici, která danou konturu (teď už jen shluk bodů) ohraničuje. Tuto kružnici využijeme jako referenční kružnici, protože udává přibližnou pozici středu a velikosti slepé skvrny.
5. Ve výřezu použijeme Houghovu transformaci pro nalezení kružnice. V případě, že bude nalezeno více kružnic, ponecháme jen ty, které se nacházejí uprostřed výřezu

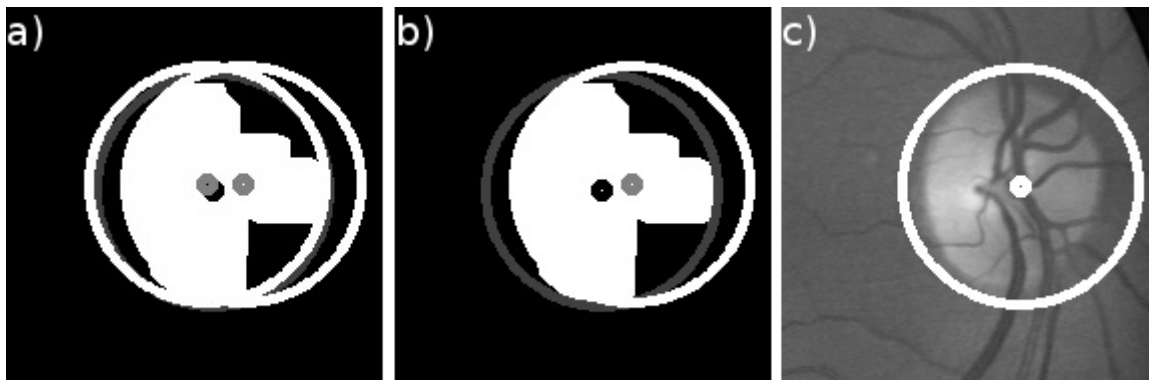
a mají poloměr alespoň stejně velký, jako má referenční kružnice. Dále ponecháme jen ty kružnice, které mají střed nalevo nebo napravo od středu referenční kružnice podle toho, zda jde o levé nebo o pravé oko (levá strana pro levé oko a pravá strana pro pravé oko). Tento krok je důležitý pro zpřesnění lokalizace. Obvykle máme ve výřezu více kružnic z nichž jedna téměř ideálně určuje střed a poloměr. Problém bývá určit, která kružnice to je (viz obr. 3.2). Jisté vodítko je právě její relativní pozice od referenční kružnice.

6. Souřadnice středů a poloměrů kružnic, které nebyly odstraněny, jsou zprůměrovány a výsledná kružnice přibližně ohraničuje slepou skvrnu. V případě, že není v předchozím kroku nalezena (nebo nevyhovuje) žádná kružnice, bere se jako výsledná kružnice referenční kružnice, což není tak přesné, ale dostaneme tak alespoň přibližný výsledek.

Tímto způsobem získáme pozici slepé skvrny, která bude ohraničena kružnicí a její přibližný průměr. Pro naše potřeby postačí jen střed kružnice a průměr d , který využijeme při lokalizaci žluté skvrny.



Obrázek 3.1: Postup lokalizace slepé skvrny. a) Oblast zájmu, b) Gaussovské rozostření, c) dilatace (nyní jsou odstraněny cévy), d) bílá oblast ve které se budou hledat kružnice.



Obrázek 3.2: Zpřesnění lokalizace slepé skvrny. Tmavě šedá kružnice je referenční kružnice a černý bod je její střed. a) Dvě detekované kružnice, b) kružnice nalevo od referenčního bodu odstraněna (jedná se o pravé oko), c) výsledná hranice a střed slepé skvrny.

3.2 Lokalizace žluté skvrny

Pro detekci žluté skvrny se používá její přibližná známá vzdálenost od slepé skvrny (asi $2,5 \cdot d$, kde d je průměr slepé skvrny [17]) a její rozložení intenzity. Je také známo, že její pozice se pohybuje kolem -6° až 3° od středové rovnoběžné osy [11]. S těmito znalostmi můžeme ve snímku určit výřez, kde se žlutá skvrna vyskytuje a hledání její přesné pozice omezit jen na tento výřez.

Výřez bude dán obdélníkem, který definujeme šířkou a výškou. Tyto rozměry určíme pomocí goniometrické funkce kosinus a pravoúhlého trojúhelníka. Rovnoběžná odvěsna a bude mít velikost odvozenou od průměru slepé skvrny a úhel, který svírá s přeponou nastavíme na 6° . Délka přepony c , výška svislé odvěsny b a délka rovnoběžné odvěsny a pak budou dány vztahy

$$a = 2,5 \cdot d, \quad (3.1)$$

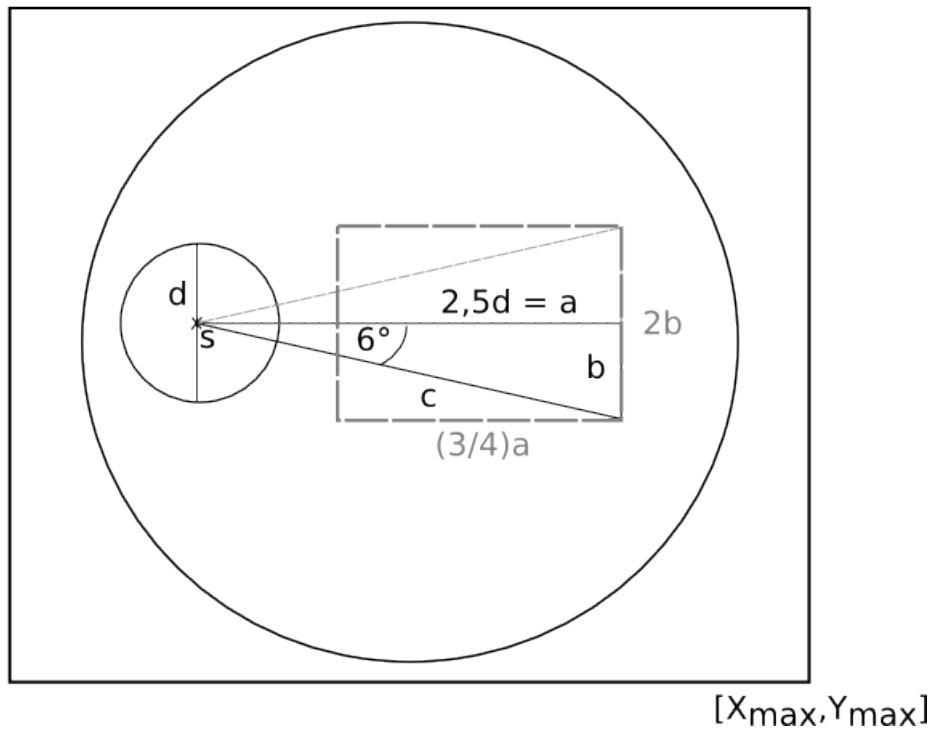
$$c = \frac{a}{\cos 6^\circ}, \quad (3.2)$$

$$b = \sqrt{c^2 - a^2}. \quad (3.3)$$

Velikost svislé odvěsny b vynásobíme dvěma a použijeme ji jako výšku obdélníka, který bude ohraničovat oblast ve které budeme žlutou skvrnu hledat. Šířku oblasti určíme jako $\frac{3}{4}a$. Pro určení pozice oblasti musíme vědět, zda snímek obsahuje sítnici pravého nebo levého oka. To určíme opět podle pozice slepé skvrny. Pokud se slepá skvrna nachází nalevo od svislé osy sítnice, jedná se o levé oko a pravá hrana obdélníka bude mít vzdálenost a od středu slepé skvrny. V opačném případě jde o pravé oko a levá hrana obdélníka bude od středu slepé skvrny vzdálena o a (viz obr. 3.3).

Ve výřezu nejprve odstraníme cévy, které mohou mít nižší intenzitu než žlutá skvrna aplikací gaussovského rozostření a dilatací. Tato úprava odstraní detaily ve výřezu, žlutá skvrna však zůstane zachována. Poté stačí najít pixel s nejnižší intenzitou.

[0,0]



Obrázek 3.3: Určení oblasti žluté skvrny (přerušovanou čarou). Písmeno s označuje střed slepé skvrny a písmeno d její průměr.

Kapitola 4

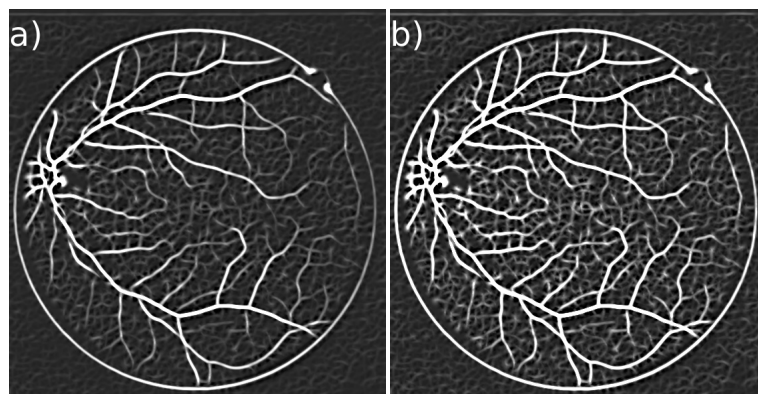
Zvýraznění cév

Extrakce krevního řečiště ze snímku sítnice je klíčová pro správnou lokalizaci bifurkací. Použitá metoda musí počítat s nízkým kontrastem cév, s nekvalitním snímkem a s různými typy poškození, které se na sítnici mohou vyskytnout. Většina metod probíhá v následujících krocích: předzpracování obrazu, segmentace krevních cév a nakonec prahování. Úkolem předzpracování je co možná nejlépe odstranit šum a upravit kontrast. Samotná segmentace je zde největší problém. Metod, které se jí zabývají je mnoho a velká část používá matematickou morfologii a tzv. *Matched Filters* [6]. V této kapitole se budu věnovat metodě *Continuation Algorithm* [14] (využívající právě Matched Filters). Prahování pak převede, obvykle stále šedotónový obraz se zvýrazněnými cévami, na výsledný černobílý obraz.

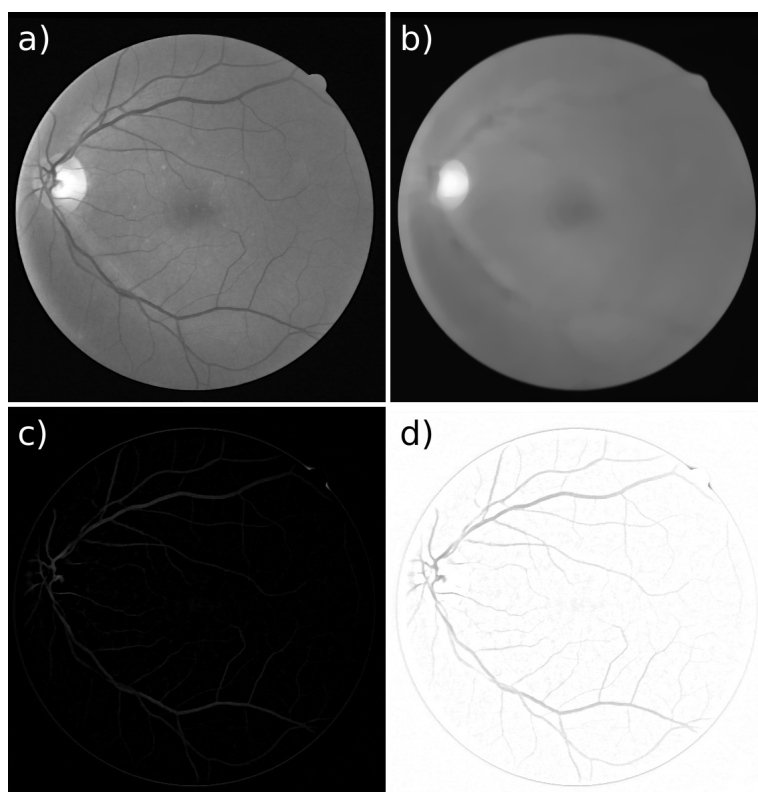
4.1 Předzpracování obrazu

4.1.1 Automatické zvýraznění cév a utlumení pozadí

Začneme homogenizací pozadí snímku. Nejdříve na snímek aplikujeme mediánový filtr o velikosti 3×3 následovaný konvolucí s Gaussovským jádrem o velikosti 9×9 s průměrem $\mu = 0$ a odchylkou $\sigma = 1,8$. Potom opět aplikujeme mediánový filtr o velikosti 25×25 . Tím získáme obrázek s extrahovaným pozadím. Na ten poté aplikujeme operaci otevření a odečteme od něj původní obrázek. Vznikne obrázek se zvýrazněnými cévami, kde bude pozadí tmavé a cévy světlé. Pro další postup budeme potřebovat aby cévy byly tmavé a pozadí světlé, proto obraz invertujeme. Nakonec mírně zlepšíme lokální kontrast metodou CLAHE (Contrast Limited Adaptive Histogram Equalization). Kontrast však nesmíme ekvalizovat příliš, jinak se zvýrazní tmavší fragmenty na pozadí snímku a později dojde k detekci falešných cév (viz obr. 4.1).



Obrázek 4.1: Výrazný nárůst nechtěných detailů v obraze při malé změně limitu u metody CLAHE: a) Limit = 1, b) limit = 3.



Obrázek 4.2: a) Zelený kanál snímku, b) zhomogenizované pozadí, c) rozdíl obrazu pozadí a původního zeleného kanálu snímku, d) invertovaný rozdíl se zvýšeným kontrastem pomocí metody CLAHE.

CLAHE

Tato metoda vychází z adaptivní ekvalizace histogramu, kde je každý pixel transformován na základě svého okolí o dané velikosti. Tento přístup má však nevýhodu v tom, že dochází k velkému zesílení šumu. V případě, že je okolí pixelu homogenní s malými změnami, roz-táhne transformační funkce malý rozsah intenzit do maximálního rozsahu, který je platný

pro celý obraz. Tento problém řeší metoda CLAHE nastavením limitu pro zvyšování kontrastu. Pokud je v histogramu okolí pixelu nalezena větší intenzita než je stanovený limit, sníží se její hodnota na tento limit ještě před výpočtem kumulativní distribuční funkce (CDF). Tím dojde k ovlivnění sklonu CDF a tedy i k ovlivnění transformační funkce. Část histogramu, která překročí limit se nezahazuje, ale rovnoměrně se rozprostře mezi nižší hodnoty histogramu. Tím může dojít k opětovnému navýšení hodnot nad limit. Efektivní limit se pak liší od nastaveného. V případě, že toto není žádoucí, lze ořezání histogramu opakovat vícekrát, dokud není navýšení nad limit zanedbatelné [19].

4.1.2 Odstranění odlesku

V některých případech může mít céva na původním snímku ve svém středu tenký pruh pixelů o vyšší intenzitě, který je způsoben odrazem světla od cévy při pořizování snímku. Tento jev je možné odstranit aplikací operace otevření se strukturním elementem ve tvaru kruhu o průměru tří pixelů [12].

4.2 Continuation Algorithm

Tato technika dokáže spojovat menší fragmenty krevního řečiště a zkompletovat ho. Zároveň ignoruje fragmenty, které nejsou součástí řečiště. Algoritmus se aplikuje na dva binární obrazy, z nichž jeden obsahuje tenké cévy a druhý širší cévy. Oba binární obrazy mohou být generovány způsobem, který dokáže extrahovat tenké nebo široké cévy, přičemž nevadí, že mezi tenkými cévami bude šum. Ten se v průběhu algoritmu eliminuje.

4.2.1 Matched filters

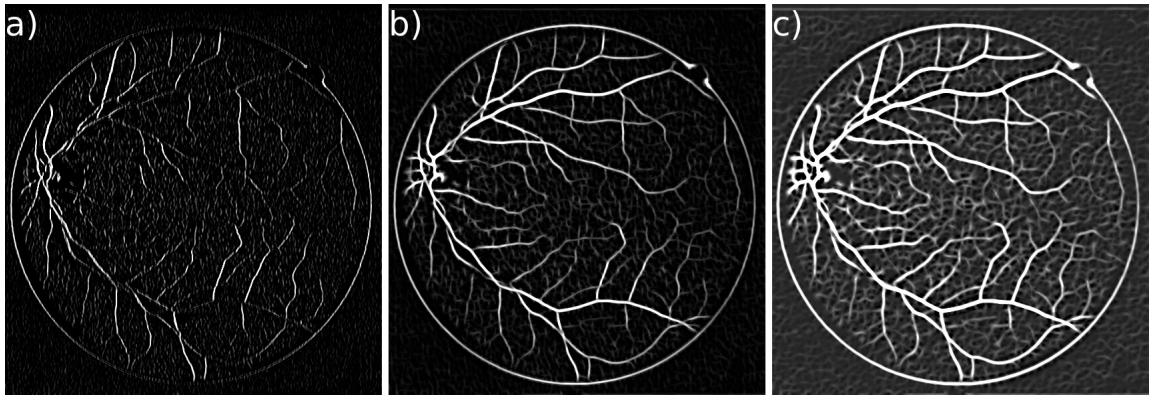
Obraz se širokými cévami vytvoříme metodou Matched Filters. Tato metoda je založená na odezvě vstupního obrazu na 2D filtr, která vzniká konvolucí daného filtru se vstupním obrazem. Metoda využívá znalost rozložení intenzity na příčném řezu cévy. Uprostřed má céva nejnižší intenzitu (kromě případu, kdy má céva ve svém středu světlejší pruh díky odrazu světla), která se postupem na okraj cévy zvyšuje [6]. Filtr pro segment cévy, který leží podél osy y lze vygenerovat pomocí vzorce

$$F(x, y) = -\exp\left(\frac{-x^2}{2\sigma^2}\right), |y| \leq \frac{L}{2} \quad (4.1)$$

kde L je předpokládaná délka segmentu po který má céva stejnou orientaci a σ je směrodatná odchylka. Od všech hodnot filtru se poté odečte střední hodnota prvků filtru a přidá se rámeček tvořený samými nulami, aby filtr správně rotoval. Filtr se poté rotuje podle zvoleného kroku, např. po 9° [14] nebo po 15° [6]. Tím vznikne sada filtrů, které budou ze vstupního obrazu generovat odezvy v různých směrech. Výslednou odezvu získáme kombinací odezev ze všech směrů, kde se pro každý pixel vezme maximální hodnota z těchto odezev.

Šířka filtru má zanedbatelný vliv na tloušťku extrahovaných cév. Čím širší filtr je, tím širší budou i extrahované cévy oproti původnímu obrazu (viz obr. 4.3). Potřebné dva obrazy pro Continuation Algorithm je možné získat pomocí změny ve velikosti konvolučního jádra. Pro obraz, kde potřebujeme široké cévy a málo šumu (dále obraz A) použijeme relativně velké konvoluční jádro. Při testování se nejlépe osvědčilo jádro se šířkou 13. Pro obraz, který bude obsahovat tenké cévy (dále obraz B) lze využít relativně malé konvoluční jádro. Pokud

se v obraze *A* bude vyskytovat šum, lze využít filtrování pomocí délky fragmentů, která by měla vyfiltrovat všechny nechtěné fragmenty. U obraze *B* tento způsob použít nelze, neboť by mohlo dojít k odstranění i částí tenkých cév.



Obrázek 4.3: Důsledky změn v šířce (w) konvolučního jádra, a) $w = 5$ b) $w = 9$ c) $w = 13$

Výsledný obraz této metody se bude stále nacházet ve stupních šedi, takže na něj musíme aplikovat některou z metod prahování. Doporučená metoda pro Matched Filters je podle [14] metoda OTSU (viz sekce 4.3).

Odstranění fragmentů

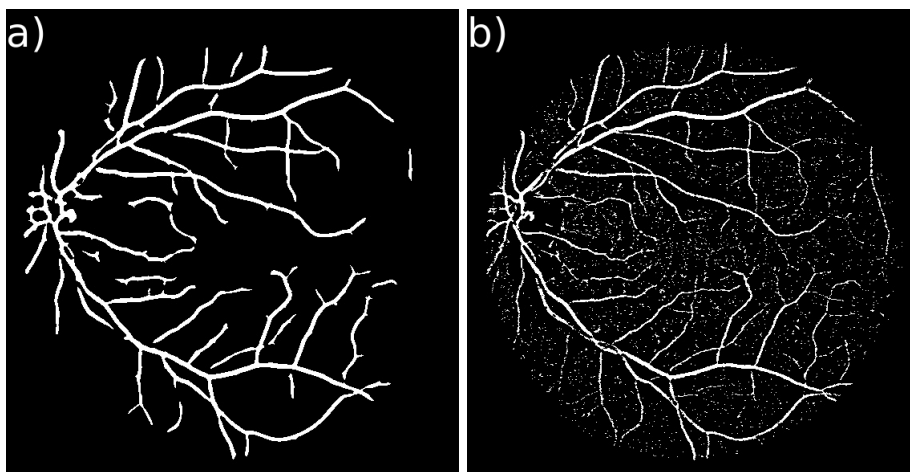
Výsledný obraz může obsahovat fragmenty, které nepatří do krevního řečiště. V obraze, kde potřebujeme široké cévy je musíme odstranit. Velmi dobrou volbou je nalézt všechny kontury v obraze, seřadit je podle velikosti a od určité velikosti je z obrazu odstranit.

4.2.2 Adaptivní ekvalizace histogramu

Obraz s tenkými cévami vytvořenými metodou Matched filters neměl při testování požadovaný efekt, protože extrahované cévy se nelišily tolik pro velká a malá konvoluční jádra. Obraz s tenkými žilami a se zvýrazněnými detaily bude vytvořen pomocí metody CLAHE.

Získáme tedy obraz s upraveným kontrastem. Vytvoříme z něj nový obraz aplikací filtrování průměrem s oknem o velikosti 9×9 a od tohoto obrazu odečteme předchozí obraz. Nakonec aplikujeme prahování pro získání černobílého obrazu.

Výsledný obraz obsahuje lépe zvýrazněné tenké cévy a mnohem víc šumu, což nám nebude vadit. Tenké cévy mohou být lehce fragmentované, to však také nevadí. V následující sekci tento obraz poslouží pro spojování cév v obraze, který vytvořila metoda Matched filters.



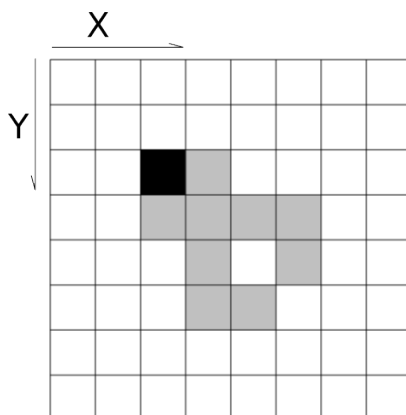
Obrázek 4.4: a) Obraz vytvořený metodou matched filters obsahující široké cévy, b) obraz vytvořený pomocí metody CLAHE obsahující tenké cévy.

4.2.3 Spojení obou obrazů

V tomto okamžiku jsou připraveny dva černobílé obrazy Obraz A s tlustými cévami a žádnými detaily a obraz B s tenkými cévami a spoustou šumu. Continuation Algorithm pracuje ve čtyřech krocích. V prvním kroku se hodnoty pixelů v obraze A , které mají bílou barvu se upraví na hodnotu 100 a v obraze B na hodnotu 1. Dále je pro každý pixel v obraze A , jehož hodnota je nula ($A(i, j) = 0$) a pro ten samý pixel v obraze B jehož hodnota je jedna ($B(i, j) = 1$) vypočítána suma hodnot pixelů podle vzorce

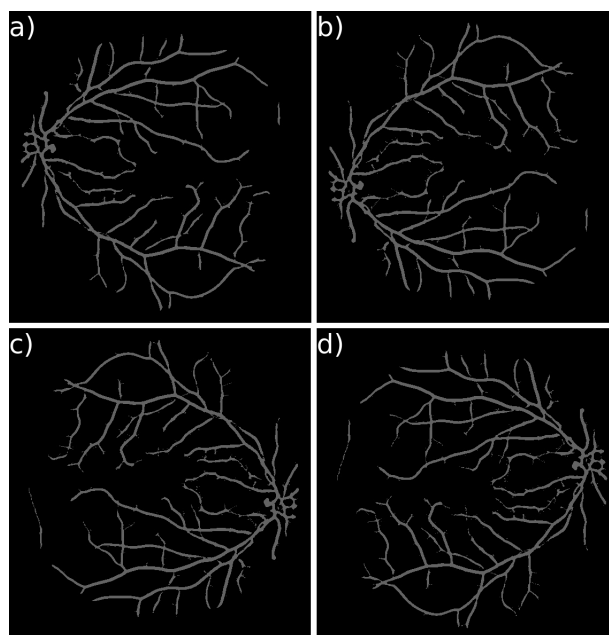
$$\begin{aligned}
 sum = & A(i, j + 1) + A(i + 1, j) + A(i + 1, j + 1) \\
 & + A(i + 1, j + 2) + A(i + 1, j + 3) + A(i + 2, j + 1) \\
 & + A(i + 2, j + 3) + A(i + 3, j + 1) + A(i + 3, j + 2)
 \end{aligned} \tag{4.2}$$

Ze kterých pixelů se suma počítá je dále vidět na obrázku 4.5. Pokud je $sum \geq 100$, pak je pixel v obraze A nastaven na hodnotu 100. Proces se opakuje tak dlouho, dokud platí podmínka $A(i, j) = 0 \wedge B(i, j) = 1$.

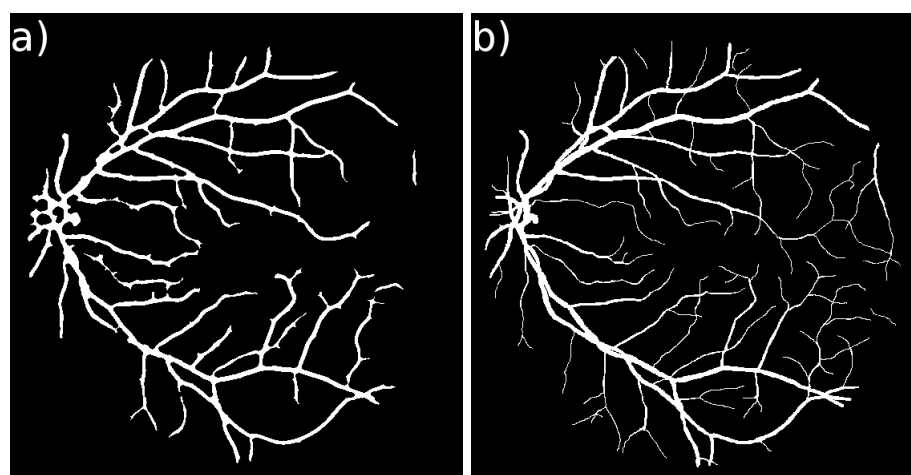


Obrázek 4.5: Suma pro Continuation algorithm. Černě označený pixel je aktuálně zpracováván pixel a šedě označené pixely jsou pixely, z nichž se počítá suma.

V druhém kroku je modifikovaný obraz A vertikálně převrácen a celý postup se opakuje. Ve třetím kroku je obraz A modifikovaný druhým krokem převrácen horizontálně a celý proces se znovu opakuje. Ve čtvrtém kroku je obraz A modifikovaný třetím krokem znovu vertikálně převrácen a celý proces prvního kroku se opakuje naposledy. Výsledný obraz je ještě převrácen horizontálně, aby jeho orientace odpovídala vstupnímu obrazu. Na závěr je na obraz aplikována operace uzavření s kruhovým strukturním elementem, aby došlo k vyplnění drobných děr a spojení fragmentů, které dosud spojeny nebyly.



Obrázek 4.6: a) Obraz A po prvním kroku, b) obraz A získaný po druhém kroku (po vertikálním převrácení), c) obraz A získaný po třetím kroku (po horizontálním převrácení), d) obraz A získaný po čtvrtém kroku (po vertikálním převrácení).



Obrázek 4.7: Výsledná segmentace cév z Continuation algorithm na snímku z databáze DRIVE. a) Automatická segmentace, b) manuální segmentace

4.3 Prahování

Výstupem metod, které zvýrazňují krevní řečiště bývá obvykle šedotónový obraz, který se až posléze prahuje, aby měl výsledný obraz pouze dvě barvy – bílou v místech, kde je céva (popředí) a černou pro pozadí. Prahování však může být použito i v průběhu metody segmentace, jako je tomu u Continuation Algorithm. Kvůli variabilitě histogramů mezi různými snímky však nemůže být použita metoda prahování s pevným prahem a práh musí být vypočten automaticky. Mnoho metod pracuje při výpočtu prahu právě s histogramem. V ideálním případě je histogram bimodální, kde jeden vrchol reprezentuje pozadí a druhý popředí a je mezi nimi velké rozpětí. V praxi se toho však jen obtížně dosahuje.

Vhodný algoritmus pro automatický výpočet prahu je metoda OTSU [15]. Tato metoda předpokládá, že obraz tvoří dvě skupiny (třídy) pixelů, jedna pro popředí, druhá pro pozadí. Vhodný práh počítá tím, že se snaží od sebe tyto dvě třídy co nejvíce oddělit tak, aby byla jejich mezitřídní variabilita co největší. Mezitřídní variabilitu definujeme jako

$$\sigma_b^2(T) = w_0(T)w_1(T) [\mu_0(T) + \mu_1(T)]^2 \quad (4.3)$$

Váhy w_0 a w_1 jsou pravděpodobnosti, že obě skupiny budou separovatelné prahem T , μ_0 a μ_1 jsou střední hodnoty těchto dvou skupin. Pravděpodobnosti w_0 a w_1 a střední hodnoty μ_0 a μ_1 jsou počítány podle:

$$w_0(T) = \sum_{i=0}^{T-1} p(i) \quad (4.4)$$

$$w_1(T) = \sum_{i=0}^{N-1} p(i) \quad (4.5)$$

$$\mu_0(T) = \sum_{i=0}^{T-1} \frac{ip(i)}{w_0(T)} \quad (4.6)$$

$$\mu_1(T) = \sum_{i=0}^{N-1} \frac{ip(i)}{w_1(T)} \quad (4.7)$$

kde N je rozsah intenzit šedotónového obrazu a $p(i)$ jsou pravděpodobnosti výskytu pixelu o intenzitě n_i . Pravděpodobnosti $p(i)$ jsou počítány jednoduše jako

$$p(i) = \frac{n_i}{N} \quad (4.8)$$

Samotný algoritmus pak pracuje následovně:

1. Spočítej histogram a pravděpodobnosti $p(i)$ pro každou úroveň intenzity.
2. Nastav počáteční $w_i(0)$ a $\mu_i(0)$
3. Iteruj přes všechny možné prahy $T = 1 \dots n_{max}$
 - (a) Aktualizuj w_i a μ_i
 - (b) Vypočítej $\sigma_b^2(T)$
4. Výsledný práh je maximální hodnota $\sigma_b^2(T)$

Výstupem je černobílý obraz, kde jsou bílou barvou označeny pixely, na kterých se nachází céva a černou barvou pixely, které v původním obraze odpovídají pozadí. Tento obraz pak slouží jako vstup pro krok ztenčování, který je blíže popsán v sekci 5.1.

Kapitola 5

Lokalizace markantů

5.1 Ztenčení cévního řečiště

Aby mohly být v krevním řečišti jednoduše vyhledány markanty, je potřeba detekované cévy ztenčit na tloušťku jednoho pixelu. Ztenčená céva musí ležet přesně ve středu původní cévy. Většina algoritmů pro ztenčování funguje na principu opakovaného odstraňování vrstev pixelů až do okamžiku, kdy není možné odstranit další vrstvu. Rychlý a jednoduchý algoritmus je Zhang-Suen [16]. V tomto algoritmu je pixel odstraněn v případě, že pro něj a jeho osmi-okolí platí následující podmínky:

1. Konektivita (počet přechodů černá \rightarrow bílá v osmi-okolí pixelu, které je seřazeno do pole podle hodinových ručiček) je rovna jedné.
2. Má alespoň dva sousedy, ale ne více než šest.
3. Soused nahoře, nalevo nebo napravo má hodnotu pozadí (alespoň jeden z nich).
4. Soused nahoře, dole a nalevo má hodnotu pozadí (alespoň jeden z nich).

Aby nedošlo k deformaci ztenčené cévy, je potřeba rovnoměrně střídat směry, ve kterých budou vrstvy odstraňovány. Změny směru dosáhneme výměnou bodů 3. a 4. za následující dva:

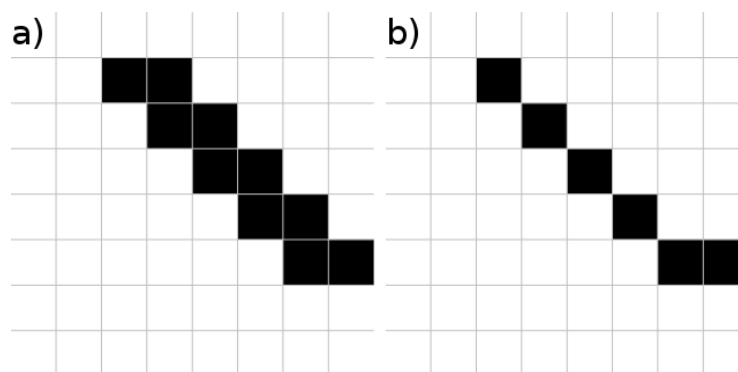
1. Soused nahoře, dole nebo napravo má hodnotu pozadí (alespoň jeden z nich).
2. Soused dole, napravo a nalevo má hodnotu pozadí (alespoň jeden z nich).

Základní varianta algoritmu ponechává v obraze schodovité struktury (viz obr. 5.1). Tyto struktury by způsobily detekování spousty falešných markantů a proto se musí odstranit. Na bílé pixely obrazu a jejich osmi-okolí se aplikují dva logické výrazy

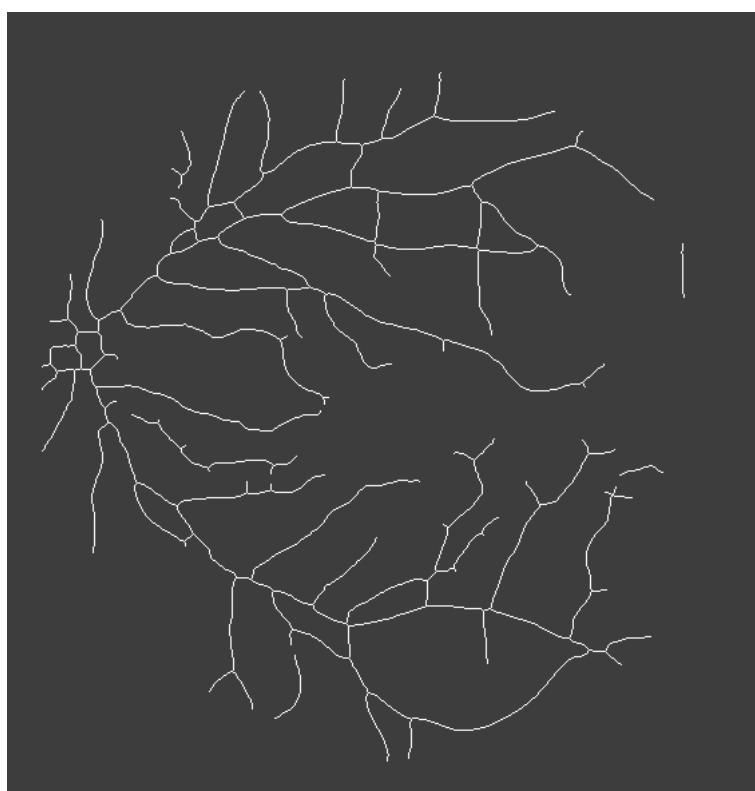
$$v(N) \wedge ((v(E) \wedge \neg v(NE) \wedge \neg v(SW) \wedge (\neg v(W) \vee \neg v(S)) \vee (v(W) \wedge \neg v(NW) \wedge \neg v(SE) \wedge (\neg v(E) \vee \neg v(S))), \quad (5.1)$$

$$v(S) \wedge ((v(E) \wedge \neg v(SE) \wedge \neg v(NW) \wedge (\neg v(W) \vee \neg v(N)) \vee (v(W) \wedge \neg v(SW) \wedge \neg v(NE) \wedge (\neg v(E) \vee \neg v(N))), \quad (5.2)$$

kde velkými písmeny jsou označeny pixely z osmi-okolí podle světových stran (N = horní pixel, S = spodní pixel, apod.) a funkce $v(x)$ vrací **true**, pokud má pixel x bílou barvu. Pokud je alespoň jeden z těchto výrazů **true**, pixel je odstraněn.



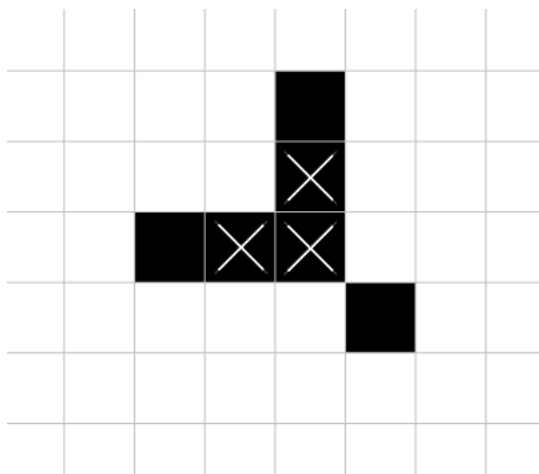
Obrázek 5.1: Ukázka schodových struktur: a) obraz se schodovými strukturami, b) obraz bez schodových struktur.



Obrázek 5.2: Ztenčené cévy na tloušťku 1px u snímku z databáze DRIVE.

5.2 Lokalizace bifurkací

Jakmile získáme ztenčené cévy, postupně projdeme všechny jejich pixely a kontrolujeme osmi-okolí těchto pixelů. Nejjednodušší varianta pouze spočítá počet pixelů v osmi-okolí a pokud je počet tři a více, znamená to bifurkaci. Problém je v tom, že při ztenčení mohou vzniknout krátké výběžky, které na původní cévě nejsou, ale zde by byly detekovány jako bifurkace. Dalším problémem jsou vzory, kdy dojde k detekci bifurkací na třech sousedních pixelech. Příklad takového vzoru je na obrázku 5.3.



Obrázek 5.3: Problémový vzor ve kterém budou tři sousední pixely (označené křížkem) označeny jako bifurkace, protože v osmi-okolí každého z nich se nacházejí tři další pixely.

Krátké výběžky lze z detekování vřadit kontrolou délky cév, které vycházejí ze zadaného bodu. Vždy když jsou v osmi-okolí bodu tři pixely bílé barvy, postupuje se po cévách, které těmito pixely započaly a počítá se jejich délka. To probíhá tak, že se při postupu na cévu, která započala konkrétním pixelem, uloží na zásobník první pixel cévy a směr ze kterého se na daný pixel přišlo. Dokud jsou pak na zásobníku nějaké pixely a dokud nebyla dosažena požadovaná délka vybere se z vrcholu zásobníku pixel a z jeho osmi-okolí se všechny pixely uloží na zásobník kromě pixelu, ze kterého jsme přišli na současný pixel a inkrementuje se počítadlo délky. Pokud je požadovaná délka dosažena u všech cév vycházejících z bodu, je v tomto bodě lokalizována bifurkace. Je pak jedno, kolik je v prohledávaném segmentu bifurkací a jak jsou dlouhé. Počítá se pouze počet pixelů od počátečního pixelu dále.

Větší počet bifurkací v sousedních pixelech bude vyřešen sloučením bifurkací, které jsou umístěny blízko u sebe. Pozice bifurkace bude dána vzdáleností d od středu slepé skvrny a úhlem φ , který svírá s přímkou mezi slepou a žlutou skvrnou. Při hledání bifurkací je budeme vkládat do vyhledávacího stromu, kde klíčem bude vzdálenost od slepé skvrny. Před samotným vložením konkrétní bifurkace do stromu se v něm vyhledají bifurkace s nejpodobnější vzdáleností. Tyto se pak projdou a vypočítá se průměrná vzdálenost a úhel těch, které budou mít velmi blízký úhel ke vkládané bifurkaci. Ta se pak vloží se zprůměrovanou vzdáleností i úhlem a původní bifurkace se odstraní. Pro případ, že se objeví více bifurkací se stejným klíčem (tedy s přesně stejnou vzdáleností), obsahuje uzel stromu seznam bifurkací.

Zda bude klíčem vyhledávacího stromu vzdálenost nebo úhel není podstatná, protože počet bifurkací na úhlu nebo vzdálenosti se příliš neliší. Záleží samozřejmě na zvolené toleranci, kdy bifurkace patří nebo nepatří do daného úhlu/vzdálenosti.

Bifurkace se nejprve uchovávají v kartézském systému souřadnic podle toho, kde se v obraze nacházejí a až před jejich uložením do vyhledávacího stromu se provede přepočítání na polární souřadnice podle

$$d = \sqrt{x^2 + y^2}, \quad (5.3)$$

$$\varphi = \arctan 2(y, x), \quad (5.4)$$

kde x a y označují vzdálenosti od středu slepé skvrny v osách X a Y (viz obr. 5.4) a funkce

$\arctan 2$ je definována takto:

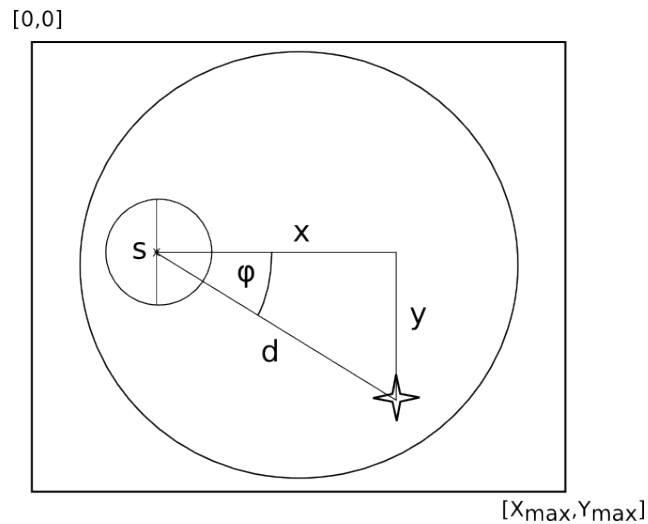
$$\arctan 2(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{je-li } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{je-li } x < 0 \wedge y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{je-li } x < 0 \wedge y < 0 \\ \frac{\pi}{2} & \text{je-li } x = 0 \wedge y > 0 \\ -\frac{\pi}{2} & \text{je-li } x = 0 \wedge y < 0 \\ \text{nedefinováno} & \text{je-li } x = 0 \wedge y = 0 \end{cases} \quad (5.5)$$

Využijeme také převod z polárních souřadnic zpět na kartézské souřadnice podle

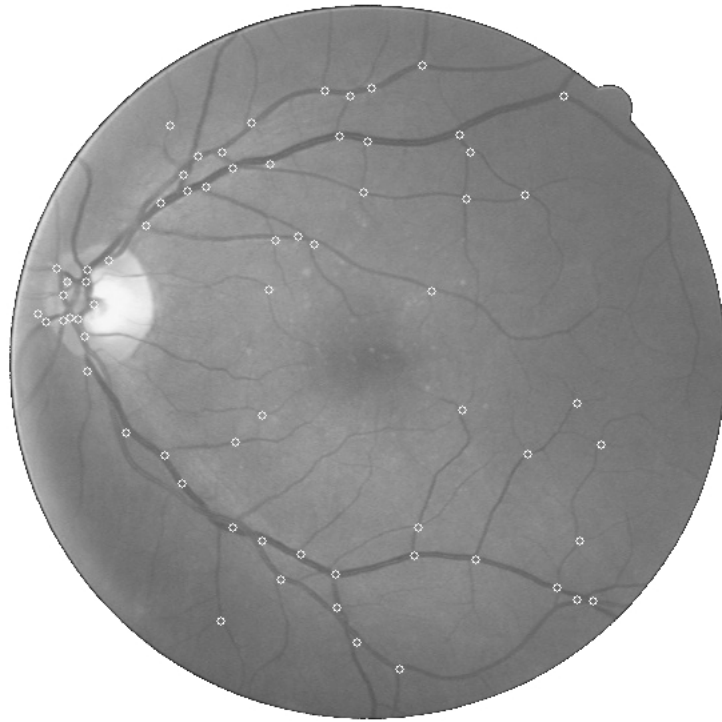
$$x = d \cos(\varphi), \quad (5.6)$$

$$y = d \sin(\varphi). \quad (5.7)$$

Získané souřadnice budou relativní vzhledem ke středu slepé skvrny. Abychom dostali přesné umístění bifurkace v obraze, musíme hodnoty x a y přičíst k souřadnicím středu slepé skvrny.



Obrázek 5.4: Převod mezi systémy souřadnic. Bifurkace je znázorněna čtyřcípou hvězdičkou. Písmeno s označuje střed slepé skvrny.



Obrázek 5.5: Lokalizované bifurkace na snímku z databáze DRIVE.

Kapitola 6

Implementace

Program pro lokalizaci bifurkací je napsán v programovacím jazyce Java verze 1.8. Pro práci s obrazem byla využita knihovna *OpenCV* ve verzi 3.0. Program je logicky strukturován do několika tříd, které všechny dědí od abstraktní třídy *Detector*. Tato třída obsahuje základní rozhraní pro ostatní třídy, metody pro extrakci kanálů z obrazu a především metodu pro kruhové oříznutí obrazu po obvodu sítnice.

Pro detekci slepé skvrny slouží třída *OpticDiscDetector*, která na vstupu očekává barevný obraz sítnice a na výstup dá bod, který udává střed slepé skvrny, její průměr a zda se jedná o snímek levého nebo pravého oka. Tento výstup slouží jako vstup třídy *FoveaDetector*, která vybere v obrazu oblast, kde se pravděpodobně nachází žlutá skvrna a v této oblasti se jí pokusí vyhledat. Jejím výstupem je bod, který udává střed žluté skvrny.

Pro extrakci krevních cév je zde třída *CAVesselDetector*, která na vstupu očekává snímek. Třída pak provede automaticky úpravy vedoucí ke zvýraznění cév v obraze a jejich extrakci pomocí algoritmu popsaném v kapitole 4.2. Výstupem je černobílý obraz, kde jsou bílou barvou označeny cévy a černou pozadí.

Tento obraz slouží jako vstup třídě *Thinning*, ve které je implementován ztenčovací algoritmus podle sekce 5.1. Výstupem je opět černobílý obraz s cévami ztenčenými na tloušťku jednoho pixelu.

Tento obraz se vkládá na vstup třídě *Bifurkations*, která ve ztenčených cévách hledá bifurkace. Výstupem této třídy je seznam objektů typu *Bifurkation*, které obsahují všechny potřebné informace o bifurkacích. Polární souřadnice je možno převést opět na kartézské souřadnice, aby je bylo možné zakreslit do obrazu.

Kapitola 7

Výsledky testů

7.1 Úspěšnost hledání slepé a žluté skvrny

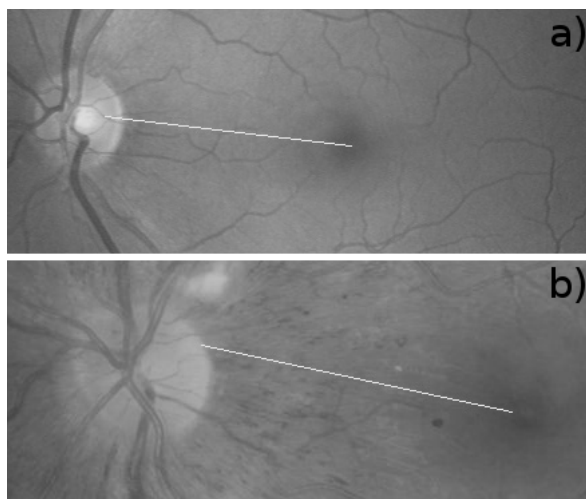
Kontrola správnosti určení slepé a žluté skvrny bude pouze vizuální. Testy byly provedeny na databázích DRIVE [18], části databáze Messidor (Base21) [9] a Gold. Zejména u slepé skvrny jsem se zaměřil na to, jestli je detekována na úplně jiném místě, nebo zda je její střed opravdu výrazně mimo skutečný střed. Obrázek 7.1 ukazuje příklad toho, co je bráno jako chyba a co je tolerováno. U žluté skvrny je situace jednodušší, protože tam jsme buď v toleranci nebo úplně mimo skvrnu. Výsledky testu všech tří databází pro slepou skvrnu jsou uvedeny v tabulce 7.1 a pro žlutou skvrnu v tabulce 7.2.

Databáze	Správně lokalizovány	Chybně lokalizovány	Úspěšnost [%]
Base21	90	10	90%
DRIVE	40	0	100%
Gold	42	3	93,33%
Celkem	172	13	92,97%

Tabulka 7.1: Úspěšnost lokalizace slepé skvrny.

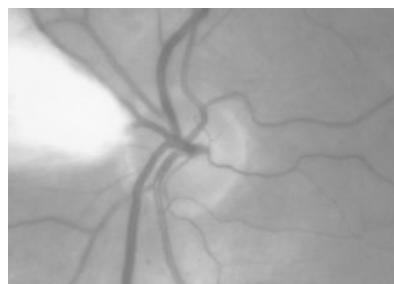
Databáze	Správně lokalizovány	Chybně lokalizovány	Úspěšnost [%]
Base21	95	5	95%
DRIVE	35	5	87,50%
Gold	44	1	97,78%
Celkem	174	11	94,05%

Tabulka 7.2: Úspěšnost lokalizace žluté skvrny.

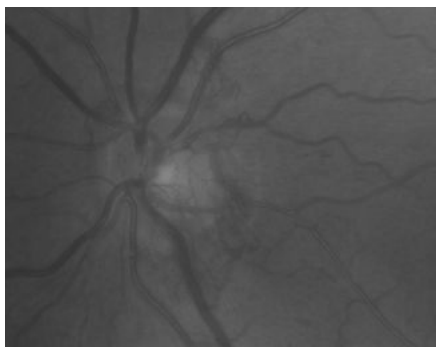


Obrázek 7.1: Příklad a) bude akceptován jako správně lokalizovaná slepá skvrna, zatímco b) bude brán jako chyba (koncové body úsečky určují střed slepé a žluté skvrny).

Chybnou lokalizaci slepé skvrny obvykle způsobuje nějaká oblast, která je ve snímku světlejší než sama slepá skvrna (viz obr. 7.2). V takovém případě se slepá skvrna vyhledává ve výřezu této oblasti. Slepá skvrna může také mít malý kontrast vůči pozadí, což rovněž způsobí výskyt světlejší oblasti na jiném místě v obraze (viz obr 7.3).



Obrázek 7.2: Výrazná světlá oblast hned vedle slepé skvrny.



Obrázek 7.3: Slepá skvrna s nízkým kontrastem vůči pozadí sítnice.

V případě chybné lokalizace žluté skvrny bývá naopak na vině oblast s tmavší barvou, než má skutečná žlutá skvrna. K chybné detekci obvykle dochází v případě, že je na sítnici

nemoc. Někdy je však žlutá skvrna lokalizována špatně jen kvůli tomu, že byla špatně lokalizována slepá skvrna, protože pozice žluté skvrny se odvíjí od pozice a velikosti slepé skvrny. V takových případech je běžně za žlutou skvrnu brána výrazná tmavá žíla.

7.2 Úspěšnost lokalizace bifurkací

Pro otestování správné pozice detekovaných bifurkací mám k dispozici manuálně označené bifurkace ve snímcích databáze *Gold*, které mi laskavě poskytl pan Ing. Lukáš Semerád¹. Každému snímku databáze *Gold* odpovídá jeden textový soubor s manuálním označením. Na prvním řádku souboru jsou souřadnice a rozměry slepé skvrny, na druhém řádku je pozice žluté skvrny a dále každý řádek odpovídá jedné bifurkaci. Pozice bifurkací je dána dvojicí (vzdálenost, úhel). Vzhledem k tomu, že zde neporovnávám mezi sebou pozice bifurkací více snímků stejné sítnice, ale pouze bifurkace v jednom snímku ke každé sítnici, mohu pozici převést do kartézského systému souřadnic. Pozice bifurkace pak bude dána souřadnicemi bodu x a y v obraze.

Bifurkace budeme porovnávat následujícím způsobem. Vytvoříme pomocnou matici, která bude mít stejnou velikost (počet řádků a sloupců) jako matice snímku sítnice a bude obsahovat samé nuly. Poté projdeme všechny manuálně označené bifurkace a na příslušných souřadnicích v pomocné matici zakreslíme kruh s nějakým rozumně zvoleným poloměrem. Jakmile zakreslíme všechny manuálně označené bifurkace, projdeme automaticky označené bifurkace a budeme na jejich pozicích kontrolovat pomocnou matici. V případě, že se pro danou bifurkaci trefíme dovnitř nějakého kruhu, máme *true positive* (TP). V případě, že se do žádného kruhu v pomocné matici netrefíme, máme *false positive* (FP). Nakonec spočítáme kruhy, do kterých jsme se ani jednou netrefili (*false rejection* (FR) nebo též *false negative* (FN)).

Postupně budeme počítat výskyty hodnot *true positive*, *false positive* a *false rejection* a pro každý snímek vypočítáme hodnoty *True Positive Rate* (TPR), *False Positive Rate* (FPR nebo též FAR) a *False Rejection Rate* (FRR) podle vzorců:

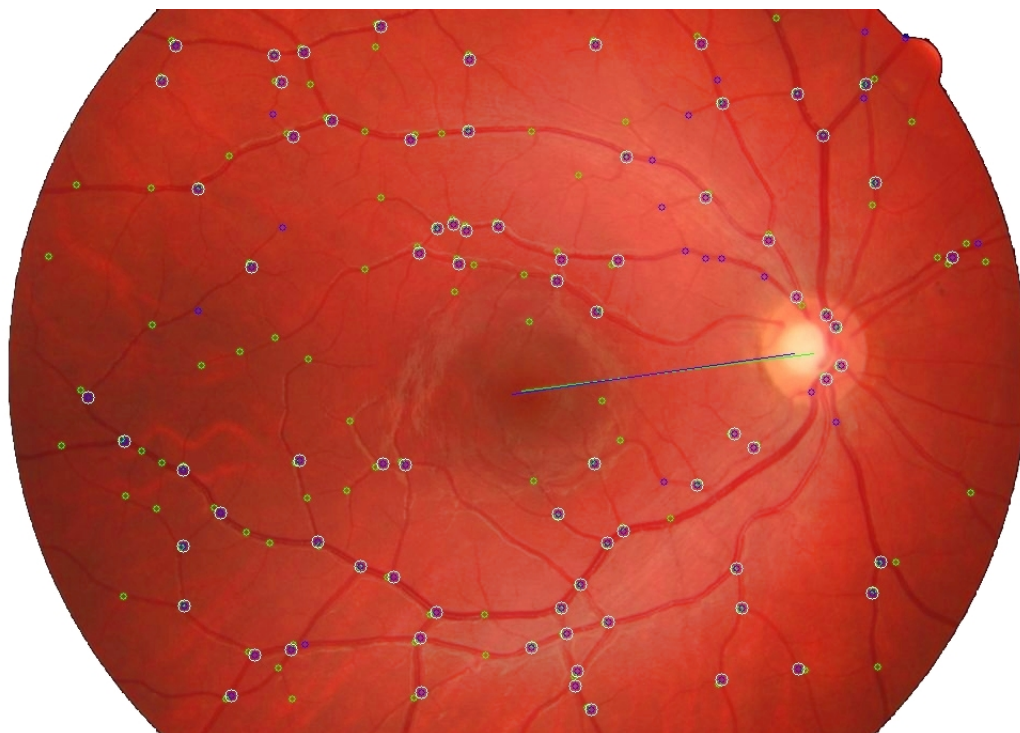
$$\text{TPR} = \frac{\text{TP}}{N}, \quad (7.1)$$

$$\text{FPR} = \frac{\text{FP}}{N}, \quad (7.2)$$

$$\text{FRR} = \frac{\text{FR}}{M}, \quad (7.3)$$

kde N je počet automaticky označených bifurkací a M je počet manuálně označených bifurkací. K automatickému porovnání je vytvořen program v jazyce Java, který načte textové soubory se souřadnicemi bifurkací a příslušný snímek sítnice. Bifurkace přepočítá do kartézského systému souřadnic a zakreslí je do snímku. Poté označí automaticky nalezené bifurkace, které byly lokalizovány na správných místech (viz obr. 7.4).

¹<http://www.stud.fit.vutbr.cz/~isemerad/>



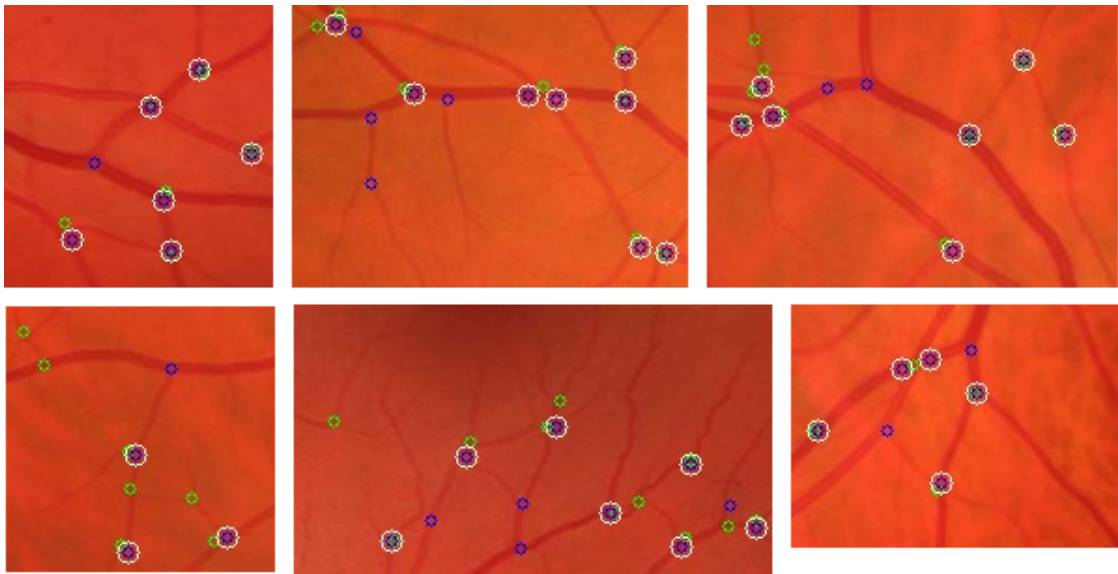
Obrázek 7.4: Ukázka porovnání manuálně a automaticky lokalizovaných bifurkací. Zelenou barvou mají bifurkace označené manuálně, modrou barvou pak ty, které byly lokalizovány automaticky. Bílou barvou jsou označeny správně lokalizované bifurkace. Zelená úsečka spojuje manuálně a modrá automaticky označenou slepou s žlutou skvrnu.

7.2.1 Výsledky pro databázi Gold

Zprůměrované výsledky testů jsou uvedeny v tabulce 7.3 a výsledky pro jednotlivé snímky jsou v tabulce C.1. Při testování se ukázalo, že některá manuální označení nejsou dokonalá a můj program našel bifurkaci v místě, kde se skutečně nacházela a nebyla označena. Při kontrole jsem takových bifurkací našel více a je tedy jasné, že tím byla mírně ovlivněna hodnota FPR.

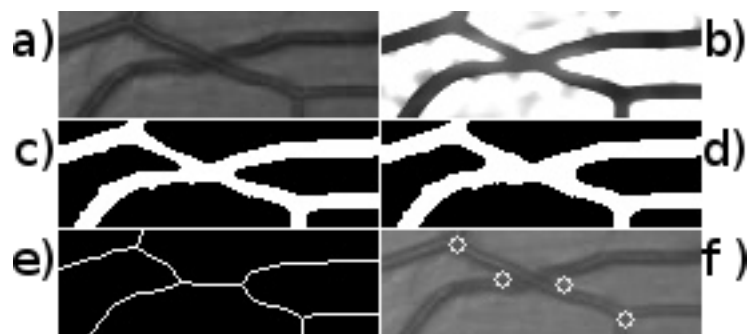
TP	58,4222	TPR	0,6723
FP	28,8889	FPR	0,3277
FR	33,4000	FRR	0,3659
N	87,3111	M	91,8000

Tabulka 7.3: Zprůměrované výsledky testů pro všechny snímky v databázi Gold.



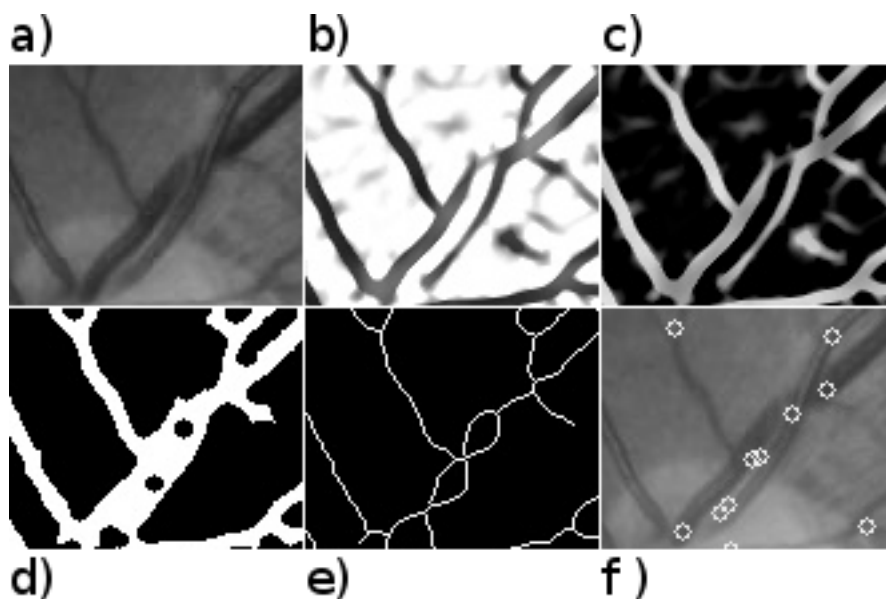
Obrázek 7.5: Příklady, kdy byly některé viditelné bifurkace označeny pouze automaticky (modrá = automaticky, zelená = manuálně).

Nutno zmínit, že některé bifurkace byly detekovány zbytečně nebo nepřesně v místech, kde dochází ke křížení nebo větvení cév pod ostrým úhlem. Kvůli ztenčování cév zde dochází k obrazcům, které připomínají písmeno H a chybně jsou zde detekovány dvě bifurkace (viz obr. 7.6). Dilatace, která je použita pro spojení některých fragmentů cév problém ještě zhoršuje. V některých případech je toto vyřešeno zprůměrováním bifurkací, které jsou příliš blízko u sebe. Práh minimální vzdálenosti však nelze nastavit na příliš vysokou hodnotu, protože by mohly být zprůměrovány i bifurkace v místech, kde by se měly nalézat dvě bifurkace blízko u sebe. Tento problém se vyskytuje téměř u všech snímků, nejen u databáze Gold.



Obrázek 7.6: Příklad problémového křížení cév. Na c) a d) je vidět, jak dilatace vyplní malé mezery a tím přispěje ke zhoršení problému.

Další problém, který stojí za zmínku je falešná detekce bifurkací v místech, kde běží dvě cévy těsně vedle sebe. Opět je na vině dilatace, která často způsobuje, že obě cévy splynou do jedné a zase se rozdělí. To při ztenčení způsobí útvary, které jsou vidět na obr. 7.7.



Obrázek 7.7: Ukázka falešných bifurkací u dvou paralelně běžících cév.

Poslední problém, který se při testování objevil, je detekování falešných bifurkací na okrajích snímku, kde je mimo sítnici černá nebo velmi tmavá barva. Pokud nedojde k ořezání po kroku, kdy se na obraz aplikuje technika Matched filters, můžou ve snímku zůstat tyto okraje zvýrazněny a pokud nezmizí při prahování, jsou dále brány jako cévy. U kruhových snímků není ořezání problém, ale u snímků, které mají tvar oválu (nahore a dole rovný, po stranách kruhový) je přesné automatické ořezání obtížné. Při testování se u snímků z databáze Gold, které mají právě oválný tvar, vyskytlo jen malé množství falešných bifurkací a proto jsem je ponechal neořezané.

7.3 Úspěšnost extrakce cév

Databáze DRIVE obsahuje celkem 40 snímků a ke každému snímku je k dispozici manuálně extrahované krevní řečiště od lidského experta. Test provedeme tak, že pro každý snímek změříme parametry *sensitivity*, *specificity* a *accuracy*, které jsou definovány následovně:

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \quad (7.4)$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \quad (7.5)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (7.6)$$

kde hodnota TP (*true positive*) udává počet pixelů, kde byla správně určena céva, TN (*true negative*) udává počet pixelů, kde bylo správně určeno pozadí, FP (*false positive*) je počet pixelů, kde byla nesprávně určena céva a FN (*false negative*) je počet pixelů ve kterých bylo chybně určeno pozadí snímku.

Pro automatické porovnání jsem vytvořil program v jazyce Java, který načte dva snímky, které spolu po jednotlivých pixelech porovná a spočítá hodnoty *sensitivity*, *specificity* a *accuracy*. Zprůměrované výsledky je možné nahlédnout v tabulce 7.4 a výsledky pro jednotlivé snímky jsou uvedeny v tabulce C.2.

Sensitivity	Specificity	Accuracy
0,7123	0,9770	0,9537

Tabulka 7.4: Zprůměrované výsledky porovnání segmentace cév pro všechny snímky v databázi DRIVE.

Kapitola 8

Závěr

V rámci této diplomové práce byl vytvořen program v jazyce Java pro automatizovanou úpravu snímků sítnice a vyhledávání bifurkací v krevním řečišti s využitím knihovny OpenCV pro práci s obrazem. Program umí exportovat obraz s upraveným kontrastem tak, aby byly zvýrazněny cévy a utlumeno pozadí sítnice a umí exportovat extrahované krevní řečiště v černobílém obraze. Nalezené bifurkace jsou vypsané v textové formě na standardní výstup. Program se používá z příkazové řádky.

Program dokáže s poměrně dobrou úspěšností detekovat slepou a žlutou skvrnu ve snímcích zdravých sítnic. Je-li však snímek nekvalitní nebo je na sítnici nemoc, může lokalizace vykazovat velkou chybovost. Zejména detekci slepé skvrny lze výrazně vylepšit například metodou prezentovanou v [21].

Detekce bifurkací v krevním řečišti sítnice má celkem dobré výsledky na širokých a dobře viditelných cévách. V případě některých tenkých cév nejsou bifurkace nalezeny vůbec, protože algoritmus extrakce cév tyto cévy vůbec nerozeznal. Tato oblast má velký prostor pro zlepšení, protože pokročilých metod segmentace cév s prezentovanou dobrou úspěšností je mnoho. Dobrou volbou mohou být např. neuronové sítě.

Literatura

- [1] File:Schematic diagram of the human eye cs.svg.
URL https://commons.wikimedia.org/wiki/File:Schematic_diagram_of_the_human_eye_cs.svg
- [2] Fundus Photography Overview.
URL <http://www.opsweb.org/?page=fundusphotography>
- [3] Simple Anatomy of the Retina by Helga Kolb.
URL <http://webvision.med.utah.edu/book/part-i-foundations/simple-anatomy-of-the-retina/>
- [4] Structure of the retina.
URL <http://en.excimerclinic.ru/retina/structure/>
- [5] Akhavan, R.; Faez, K.: Two Novel Retinal Blood Vessel Segmentation Algorithms. *International Journal of Electrical and Computer Engineering (IJECE)*, ročník 4, č. 3, 2014: s. 398–410, ISSN 2088-8708.
URL <http://iaesjournal.com/online/index.php/IJECE/article/view/5829>
- [6] Chaudhuri, S.; Chatterjee, S.; Katz, N.; aj.: Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging*, ročník 8, č. 3, 1989: s. 263–269.
URL <http://repository.ias.ac.in/7783/>
- [7] Chen, Z.; Abidi, B.; Page, D.; aj.: Gray-level grouping (GLG). *IEEE Transactions on Image Processing*, ročník vol. 15, č. issue 8, 2006: s. 2290–2302, ISSN 1057-7149, doi:10.1109/TIP.2006.875204.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1658093>
- [8] Chen, Z.; Abidi, B.; Page, D.; aj.: Gray-level grouping (GLG). *IEEE Transactions on Image Processing*, ročník vol. 15, č. issue 8, 2006: s. 2303–2314, ISSN 1057-7149, doi:10.1109/TIP.2006.875201.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1658094>
- [9] Decencière, E.; Zhang, X.; Cazuguel, G.; aj.: FEEDBACK ON A PUBLICLY DISTRIBUTED IMAGE DATABASE. *Image Analysis*, ročník vol. 0, č. issue 0, 2014-07-04: s. –, ISSN 1854-5165, doi:10.5566/ias.1155.
URL <http://www.ias-iss.org/ojs/IAS/article/view/1155>

- [10] Drahanický, M.; Orság, F.: *Biometrie*. [Brno: M. Drahanický], první vydání, 2011, ISBN 978-80-254-8979-6.
- [11] Drozd, R.; Hájek, J.; Drahanický, M.: An Algorithm for Retina Features Extraction Based on Position of the Blood Vessel Bifurcation: str. 308.
doi:10.1007/978-3-642-35136-5_37.
URL http://link.springer.com/10.1007/978-3-642-35136-5_37
- [12] Kumar, P.; Atlas, L. G.: An Efficient Method for Retinal Hemorrhages Detection in Fundus Images Using Anfis and Cross Section Profile Analysis. *Middle-East Journal of Scientific Research*, ročník 23, č. 10, 2015: s. 2484–2492, ISSN 1990-9233, doi:10.5829/idosi.mejsr.2015.23.10.95132.
URL <http://www.idosi.org/mejsr/mejsr23%2810%2915/19.pdf>
- [13] Mendonca, A.; Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Transactions on Medical Imaging*, ročník vol. 25, č. issue 9, 2006: s. 1200–1213, ISSN 0278-0062, doi:10.1109/TMI.2006.879955.
URL
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1677726>
- [14] Mudassar, A. A.; Butt, S.: Extraction of Blood Vessels in Retinal Images Using Four Different Techniques. *Journal of Medical Engineering*, ročník vol. 2013, 2013: s. 1–21, ISSN 2314-5129, doi:10.1155/2013/408120.
URL <http://www.hindawi.com/journals/jme/2013/408120/>
- [15] Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, ročník vol. 9, č. issue 1, 1979: s. 62–66, ISSN 0018-9472, doi:10.1109/TSMC.1979.4310076.
URL
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4310076>
- [16] Parker, J.: *Algorithms for image processing and computer vision*. New York: Wiley Computer Pub., 1997, ISBN 978-0-471-14056-6.
- [17] Raj, V.; Devi, V.: Retinal Image Analysis Using Fovea Detection using Unsymmetrical Trimmed Median Filter. *International Journal of Engineering and Innovative Technology (IJEIT)*, ročník 3, č. 3, 2013: s. 263–269, ISSN 2277-3754.
URL http://www.ijeit.com/Vol%203/Issue%203/IJEIT1412201309_41.pdf
- [18] Staal, J.; Abramoff, M.; Niemeijer, M.; et al.: Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, ročník 23, č. 4, 2004: s. 501–509.
- [19] Wikipedia: Adaptive histogram equalization — Wikipedia, The Free Encyclopedia. 2016, [Online].
URL https://en.wikipedia.org/w/index.php?title=Adaptive_histogram_equalization&oldid=720123250
- [20] Yang, Y.; Huang, S.; Rao, N.: An Automatic Hybrid Method for Retinal Blood Vessel Extraction. *International Journal of Applied Mathematics and Computer Science*, ročník vol. 18, č. issue 3, 2008-01-1: s. –, ISSN 1641-876x,

doi:10.2478/v10006-008-0036-5.

URL <http://www.degruyter.com/view/j/amcs.2008.18.issue-3/v10006-008-0036-5/v10006-008-0036-5.xml>

- [21] Youssif, A. A.-H. A.-R.; Ghalwash, A. Z.; Ghoneim, A. A. S. A.-R.: Optic Disc Detection From Normalized Digital Fundus Images by Means of a Vessels' Direction Matched Filter. *IEEE Transactions on Medical Imaging*, ročník vol. 27, č. issue 1, 2008: s. 11–18, ISSN 0278-0062, doi:10.1109/TMI.2007.900326.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4408713>
- [22] Zhang, B.; Zhang, L.; Zhang, L.; aj.: Retinal vessel extraction by matched filter with first-order derivative of Gaussian. *Computers in Biology and Medicine*, ročník vol. 40, č. issue 4, 2010: s. 438–445, ISSN 00104825, doi:10.1016/j.combiomed.2010.02.008.
URL <http://linkinghub.elsevier.com/retrieve/pii/S0010482510000302>

Přílohy

Seznam příloh

A Obsah CD	38
B Manual	39
B.1 Zprovoznění programu	39
B.2 Použití programu pro detekci bifurkací	39
B.3 Použití programu pro porovnávání manuálního a automatického označení bifurkací	40
B.4 Použití programu pro porovnání manuálně a automaticky segmentovaného krevního řečiště	40
C Výsledky testů	41

Příloha A

Obsah CD

- **bin** – Adresář se spustitelnými programy:
 - **CompareTests** – Program pro porovnání manuálně a automaticky označených bifurkací.
 - **CompareVessels** – Program pro porovnání manuálně a automaticky segmentovaného krevního řečiště.
 - **RetinaFeatureExtractor** – Program pro vyhledávání bifurkací ve snímcích sítnice.
- **doc** – Adresář s pdf souborem obsahujícím text této diplomové práce a její zdrojové texty pro software \LaTeX .
- **src** – Zdrojové kódy programů a projektové adresáře prostředí NetBeans. Obsahuje podadresáře:
 - **CompareTests** – Zdrojové kódy a projektové soubory programu pro porovnání manuálně a automaticky označených bifurkací.
 - **CompareVessels** – Zdrojové kódy a projektové soubory programu pro porovnání manuálně a automaticky segmentovaného krevního řečiště.
 - **RetinaFeatureExtractor** – Zdrojové kódy a projektové soubory programu pro vyhledávání bifurkací ve snímcích sítnice.
- **test** – Skripty pro automatické testování a adresáře s výstupy testů. Obsahuje podadresáře:
 - **automatically_marked** – Snímky z různých databází s automaticky označenými slepými a žlutými skvrnami a s automaticky lokalizovanými bifurkacemi spolu s textovými soubory obsahující jejich souřadnice v příslušném snímku.
 - **compare_results** – Výstupní snímky programu pro porovnávání manuálně a automaticky lokalizovaných bifurkací.
 - **databases** – Databáze snímků sítnic, na které jsou nastaveny testovací skripty.
 - **manually_marked** – Textové soubory se souřadnicemi slepých a žlutých skvrn a bifurkací, které byly označeny manuálně.

Příloha B

Manual

B.1 Zprovoznění programu

Program ke svému chodu vyžaduje knihovnu OpenCV 3.0 se zkompilevaným Java modulem. Zkompilevaná statická knihovna spolu s `.jar` souborem pro 64-bitový linuxový systém se nachází na CD. V případě 32-bitového systému nebo v případě, že přiložená knihovna nebude na Vašem systému fungovat, je nutné přikročit k ruční kompilaci¹. Program je také možné importovat do vývojového prostředí NetBeans verze 8.0.2 a vyšší².

B.2 Použití programu pro detekci bifurkací

Program se spouští z příkazové řádky pomocí připraveného bash skriptu a očekává minimálně jeden parametr, který určuje cestu ke snímku sítnice. Na výstup vypíše pozici slepé a žluté skvrny a pozice jednotlivých nalezených bifurkací v polárních souřadnicích. Platné parametry programu jsou následující:

- `-h` nebo `--help` – Vypíše nápovědu a ukončí program.
- `--input` – Cesta ke vstupnímu snímku sítnice.
- `--output` – Soubor, kde bude uložen výstupní obraz se zakreslenými bifurkacemi. Tento parametr je nepovinný.
- `--output-vessel-tree` – Soubor, kde bude uložen černobílý obraz s extrahovanými cévami. Tento parametr je nepovinný.
- `--output-contrast-enhanced` – Soubor, kde bude uložen obraz s upraveným kontrastem, aby byly zvýrazněny krevní cévy a utlumeno pozadí sítnice. Tento parametr je nepovinný.
- `--debug-dir` – Adresář, kde se budou postupně ukládat výsledky jednotlivých kroků vyhledávání bifurkací. Tento parametr je nepovinný.

¹Návod na kompilaci: http://docs.opencv.org/3.0-beta/doc/tutorials/introduction/desktop_java/java_dev_intro.html

²Návod pro nastavení knihovny OpenCV v prostředí NetBeans: <http://www.codeproject.com/Tips/717283/How-to-Use-OpenCV-with-Java-under-NetBeans-IDE>

- `--disable-roi` – Pokud je zadán tento parametr, nedojde k ořezání hranic snímku sítnice, což může způsobit detekci falešných bifurkací. Tento parametr je doporučeno použít v případě, že snímek sítnice má jiný než kruhový tvar. Tento parametr je nepovinný.

Program vrací výsledky v textové podobě, která má následující formát:

```
sl_x,sl_y,d
zl_x;zl_y
vzdalenost;uhel
...
```

kde `sl_x`, `sl_y`, `zl_x` a `zl_y` jsou souřadnice slepé a žluté skvrny a `d` je průměr slepé skvrny. Tento formát vychází z formátu manuálního označení sítnic, který generuje aplikace od pana Ing. Lukáše Semeráda.

B.3 Použití programu pro porovnávání manuálního a automatického označení bifurkací

Tento jednoduchý program v jazyce Java očekává při spuštění čtyři parametry v tomto pořadí:

1. Cesta k souboru s manuálním označením skvrn a bifurkací.
2. Cesta k souboru s automatickým označením skvrn a bifurkací.
3. Cesta k příslušnému snímku na kterém bylo označení provedeno.
4. Výstupní soubor s obrazem, kde bude zakresleno oboje označení a kde budou označeny bifurkace, které se shodují.

Soubory s automatickým označením musí mít stejný formát, jako je výstupní formát programu pro detekci bifurkací. Je možné vynechat průměr slepé skvrny, protože s ním se zde nepracuje. Soubory s manuálním označením se liší v prvním řádku, kde je místo průměru uvedena šířka a výška elipsy, která slepou skvrnu ohraničuje. Tuto informaci je zde také možno vynechat. Důležitá je jen pozice.

Program vrací hodnoty TP, FP, FR, N, M, TPR, FPR a FRR v tomto pořadí oddělené středníkem. Jejich význam je vysvětlen v sekci 7.2.

B.4 Použití programu pro porovnání manuálně a automaticky segmentovaného krevního řečiště

Tento jednoduchý program v jazyce Java očekává při spuštění dva parametry. Oba udávají cestu k obrazovým souborům, kde je manuální nebo automatické označení pixelů, které jsou nebo nejsou céva. Oba obrazy musí být striktně černobílé. Program odmítne porovnat obrazy v nichž se vyskytuje jiná barva nebo odstín šedi. Na standardní výstup pak program vypíše získané hodnoty *sensitivity*, *specificity* a *accuracy* v tomto pořadí oddělené středníkem.

Příloha C

Výsledky testů

Snímek	TP	FP	FR	N	M	TPR	FPR	FRR
01_dr	56	19	13	75	69	0,7467	0,2533	0,1884
01_g	48	18	46	66	94	0,7273	0,2727	0,4894
01_h	71	34	37	105	108	0,6762	0,3238	0,3426
02_dr	62	26	28	88	90	0,7045	0,2955	0,3111
02_g	49	29	47	78	96	0,6282	0,3718	0,4896
02_h	102	49	28	151	130	0,6755	0,3245	0,2154
03_dr	37	41	25	78	62	0,4744	0,5256	0,4032
03_g	32	11	33	43	65	0,7442	0,2558	0,5077
03_h	77	25	43	102	120	0,7549	0,2451	0,3583
04_dr	37	27	27	64	64	0,5781	0,4219	0,4219
04_g	45	16	30	61	75	0,7377	0,2623	0,4000
04_h	65	42	29	107	93	0,6075	0,3925	0,3118
05_dr	62	16	23	78	85	0,7949	0,2051	0,2706
05_g	34	11	31	45	65	0,7556	0,2444	0,4769
05_h	49	24	27	73	76	0,6712	0,3288	0,3553
06_dr	35	25	36	60	71	0,5833	0,4167	0,5070
06_g	46	18	39	64	85	0,7188	0,2813	0,4588
06_h	74	36	44	110	118	0,6727	0,3273	0,3729
07_dr	66	29	26	95	92	0,6947	0,3053	0,2826
07_g	55	31	28	86	83	0,6395	0,3605	0,3373
07_h	50	19	27	69	77	0,7246	0,2754	0,3506
08_dr	58	59	20	117	78	0,4957	0,5043	0,2564
08_g	75	49	26	124	101	0,6048	0,3952	0,2574
08_h	68	26	36	94	104	0,7234	0,2766	0,3462
09_dr	33	23	16	56	49	0,5893	0,4107	0,3265
09_g	64	18	44	82	108	0,7805	0,2195	0,4074
09_h	46	29	22	75	68	0,6133	0,3867	0,3235
10_dr	64	31	34	95	98	0,6737	0,3263	0,3469
10_g	72	30	32	102	104	0,7059	0,2941	0,3077
10_h	48	23	50	71	98	0,6761	0,3239	0,5102
11_dr	69	28	51	97	120	0,7113	0,2887	0,4250
11_g	72	24	31	96	103	0,7500	0,2500	0,3010

11_h	75	27	19	102	94	0,7353	0,2647	0,2021
12_dr	42	32	28	74	70	0,5676	0,4324	0,4000
12_g	84	40	31	124	115	0,6774	0,3226	0,2696
12_h	69	28	43	97	112	0,7113	0,2887	0,3839
13_dr	42	30	36	72	78	0,5833	0,4167	0,4615
13_g	55	43	24	98	79	0,5612	0,4388	0,3038
13_h	78	19	50	97	128	0,8041	0,1959	0,3906
14_dr	67	50	41	117	108	0,5726	0,4274	0,3796
14_g	54	28	34	82	88	0,6585	0,3415	0,3864
14_h	72	25	53	97	125	0,7423	0,2577	0,4240
15_dr	54	52	35	106	89	0,5094	0,4906	0,3933
15_g	59	25	32	84	91	0,7024	0,2976	0,3516
15_h	57	15	48	72	105	0,7917	0,2083	0,4571

Tabulka C.1: Výsledky porovnání manuálně a automaticky označených bifurkací pro jednotlivé snímky v databázi Gold.

Snímek	Sensitivity	Specificity	Accuracy
01_test	0,7545	0,9712	0,9518
02_test	0,7878	0,9747	0,9556
03_test	0,6345	0,9857	0,9507
04_test	0,6993	0,9865	0,9600
05_test	0,6791	0,9870	0,9581
06_test	0,6375	0,9867	0,9527
07_test	0,7026	0,9796	0,9543
08_test	0,6514	0,9814	0,9530
09_test	0,6793	0,9836	0,9589
10_test	0,6888	0,9840	0,9597
11_test	0,7181	0,9707	0,9481
12_test	0,7164	0,9776	0,9550
13_test	0,6840	0,9806	0,9516
14_test	0,7511	0,9724	0,9545
15_test	0,7832	0,9665	0,9534
16_test	0,7195	0,9765	0,9533
17_test	0,6719	0,9823	0,9561
18_test	0,7469	0,9756	0,9575
19_test	0,7777	0,9786	0,9619
20_test	0,7487	0,9754	0,9588
21_training	0,7468	0,9780	0,9607
22_training	0,6864	0,9817	0,9550
23_training	0,7931	0,9317	0,9226
24_training	0,6773	0,9831	0,9477
25_training	0,5904	0,9910	0,9526
26_training	0,6666	0,9886	0,9617
27_training	0,7377	0,9768	0,9557

28_training	0,6937	0,9804	0,9524
29_training	0,6964	0,9809	0,9570
30_training	0,6563	0,9826	0,9570
31_training	0,7893	0,9733	0,9622
32_training	0,6990	0,9821	0,9589
33_training	0,7504	0,9792	0,9607
34_training	0,6968	0,9383	0,9147
35_training	0,7647	0,9746	0,9564
36_training	0,6803	0,9771	0,9448
37_training	0,7543	0,9755	0,9562
38_training	0,7310	0,9766	0,9554
39_training	0,7172	0,9735	0,9515
40_training	0,7302	0,9790	0,9601

Tabulka C.2: Výsledky porovnání manuálně a automaticky extrahovaného krevního řečiště v databázi DRIVE.