



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**SKLÁDÁNÍ SNÍMKŮ SÍTNICE OKA**

STITCHING OF RETINAL IMAGES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**VADYM HLADYUK**

**VEDOUCÍ PRÁCE**

**DRAHANSKÝ, Ph.D.**

SUPERVISOR

**Prof. Ing., Dipl.-Ing. MARTIN**

BRNO 2020

## Zadání bakalářské práce



Student: **Hladyuk Vadym**  
Program: Informační technologie  
Název: **Skládání snímků sítnice oka**  
**Stitching of Retinal Images**  
Kategorie: Umělá inteligence

### Zadání:

1. Prostudujte dostupnou literaturu zpracování snímků sítnice oka, zejména se zaměřte na jejich skládání.
2. Navrhňte algoritmus skládání více snímků sítnice oka (typicky 5 až 7).
3. Navržený algoritmus z předchozího bodu implementujte.
4. Proveďte experimentální ověření výsledku ve spolupráci s očními lékaři.
5. Sumarizujte a zhodnoťte dosažené výsledky.

### Literatura:

- AKRAM, M. Usman, et al. Seamless Fundus image stitching using WLD to improve field of view. In: *2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. IEEE, 2015. p. 106-110.
- FORUTANPOUR, Bijan; LE NGUYEN, Phi Hung; BI, Ning. *Oriented image stitching for spherical image content*. U.S. Patent Application No 10/325,391, 2019.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Drahanský Martin, prof. Ing., Dipl.-Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 31. října 2019

## Abstrakt

Cílem této práce je sestavení kompletního snímku sítnice, z několika dílčích snímků. Do této doby neexistuje způsob, jakým by bylo možné zachytit celou sítnici v jednom snímku, proto je důležité, se touto problematikou zabývat. Text představí a detailně popíše metody, algoritmy a knihovny, které byly použity k dosažení požadovaných výsledků. Na závěr textu jsou poté demonstrovány dosažené výsledky. Problém skládání snímků sítnice byl vyřešen extrakcí cév ve snímcích sítnice, vyhledáním klíčových bodů ve snímcích, nalezením společných klíčových bodů, vypočítáním transformační matice a transformací jednoho snímku na druhý. Výsledky celé práce byly také konzultovány s očním lékařem a na základě jeho doporučení byly navrženy další možné kroky, kterými by se práce mohla dále ubírat, které jsou rozebrány v textu. Text poskytne čtenáři znalosti o očním aparátu a také představí oblast barevných modelů, formátů snímků, algoritmů pro hledání klíčových bodů, transformace snímků, dále poskytne možný způsob jak poskládat snímky sítnice a navrhne možné vylepšení.

## Abstract

The purpose of this work is to create a complete picture of the retina, by stitching together a number of partial photos. Since there is no working solution which would be able to capture the entire retina in one picture, this is an important problem to cover. The results will be demonstrated at the end of the text. The problem of stitching partial pictures together was solved by extracting vessels in retinal images, finding key points in images, finding common key points, calculating a transformation matrix and transforming one image into another. After a consultation with an ophthalmologist I was able to define steps which will allow me to further improve the work, which are analyzed in texts. The thesis will provide the reader knowledge about the eye apparatus. It will also introduce field of color models, image formats, algorithms for searching for key points, the transformation of the images themselves and it will also provide a possible way to compose retinal images and also suggest possible improvements.

## Klíčová slova

medicínské snímky, počítačové vidění, skládání snímků, OpenCV, klíčové body, SIFT, KAZE, homografie, ransac, transformační matice

## Keywords

medical images, computer vision, image stitching, OpenCV, keypoints, SIFT, KAZE, homography, ransac, transformation matrix

## Citace

HLADYUK, Vadym. *Skládání snímků sítnice oka*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Ing., Dipl.-Ing. Martin Draňanský, Ph.D.

# Skládání snímků sítnice oka

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Prof. Ing., Dipl.-Ing. Martina Drahanského Ph.D. Další informace mi poskytli Ing. Lukáš Semerád, Ing. Tomáš Vičar a MUDr. Tomáš Mňuk. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Vadym Hladyuk

28. května 2020

## Poděkování

Chtěl bych poděkovat všem, kteří se na této práci podíleli. Vedoucímu Prof. Ing., Dipl.-Ing. Martinu Drahanskému Ph.D. za výborné zadání, nápady a kontakt na očního lékaře. Dále bych chtěl poděkovat Ing. Lukáši Semerádovi, který mi pohodl se vším, s čím jsem si nevěděl rady a také poskytoval výbornou zpětnou vazbu. Ing. Tomáši Vičarovi za program na extrakci cév a vysvětlení fungování programu. Panu MUDr. Tomášovi Mňukovi za skvělou zpětnou vazbu na výsledky práce, Karolíně Motílové za pomoc s kapitolou Anatomie, fyziologie a nemoci oka a dalším.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Anatomie, fyziologie a nemoci oka</b>	<b>4</b>
2.1	Anatomie oka . . . . .	4
2.2	Fyziologie zraku . . . . .	4
2.3	Nejčastější choroby zraku . . . . .	5
<b>3</b>	<b>Metody pro skládání snímků</b>	<b>7</b>
3.1	Počítačové vidění . . . . .	7
3.2	Snímek, jeho formát a práce s pozadím . . . . .	8
3.2.1	DICOM . . . . .	8
3.2.2	RAW . . . . .	8
3.2.3	JPEG . . . . .	9
3.2.4	PNG . . . . .	9
3.2.5	Barevné modely . . . . .	10
3.2.6	Míchání barev . . . . .	10
3.2.7	Model RGB . . . . .	11
3.2.8	Práce s pozadím . . . . .	12
3.3	Detekce klíčových bodů, jejich popis, hledání shodných bodů a příkladů algoritmů . . . . .	13
3.3.1	Definování a detekce klíčových bodů . . . . .	13
3.3.2	Popis klíčových bodů . . . . .	14
3.3.3	Hledání shodných bodů . . . . .	14
3.4	Harrisův detektor rohů . . . . .	15
3.5	SIFT . . . . .	16
3.5.1	Scale-space detekce extrémů . . . . .	17
3.5.2	Lokalizace klíčových bodů, v našem případě extrémů . . . . .	18
3.5.3	Přidělení orientace . . . . .	18
3.5.4	Popis klíčového bodu . . . . .	19
3.5.5	Výstup SIFT algoritmu a jeho použití pro komerční účely . . . . .	21
3.6	KAZE . . . . .	22
3.6.1	Nelineární rozptylové filtrování . . . . .	22
3.6.2	AOS schéma . . . . .	23
3.6.3	Postup algoritmu KAZE . . . . .	23
3.7	Homografie . . . . .	27
3.8	RANSAC . . . . .	28
<b>4</b>	<b>Návrh implementace</b>	<b>30</b>

4.1	Snímky a jejich formát . . . . .	30
4.2	Skládání snímků . . . . .	30
4.3	Cyklus skládání snímků . . . . .	31
<b>5</b>	<b>Implementace</b>	<b>32</b>
5.1	Knihovna OpenCV . . . . .	32
5.2	Knihovna NumPy . . . . .	32
5.3	Pillow . . . . .	32
5.4	Příprava dat . . . . .	32
5.5	Skládání snímků . . . . .	33
<b>6</b>	<b>Sumarizace a konzultace výsledků s očním lékařem</b>	<b>39</b>
<b>7</b>	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>43</b>

# Kapitola 1

## Úvod

Skládání snímků sítnice slouží především lékařům a jejich pacientům. Lékaři pro přehled o celé sítnici ovšem vždy, když zpozorují nějakou nemoc či artefakt, vždy provedou vyfocení konkrétního místa přímo snímacím aparátem, aby měli jeho co nejpřesnější snímek. Pacientům to slouží jako dobrá prezentace a je mnohem snadnější, aby lékař vysvětlil pacientovi, o jaký problém se jedná, v kontextu celé sítnice než v případě jednoho snímku. Toto skládání musí být automatické, jelikož lékař takových snímků může denně vyfotit i desítky a bylo by nepraktické, lékař pracoval na skládání snímků sítnice.

Jako dobré prvnotní zdroje informací jsou doporučené literatury od vedoucího práce [2] a [9]. Existují postupy, které se zabývají skládáním snímků do jedné větší mozaiky. Program musí být co nejobecnější, jelikož se neví, kolik snímků sítnice se bude skládat a také se neví nic o konkrétních snímcích. Skládání snímků už existuje například v kameře ZEISS VISUCAM 224/524, ovšem ta je doplněna o funkcionality, které jí pomáhají pro mnohem lepší skládání snímků, o kterém se podrobněji píše v části 6.

V této práci se budeme zabývat samotným okem 2, které je předmětem všech snímků. Dále zde v nejobsáhlejší části 3 vypíšeme a představíme všechny potřebné metody a algoritmy, které můžeme použít u skládání snímků. V dalších kapitolách probereme návrh implementace 4 a také samotnou implementaci 5 a v poslední řadě sumarizujeme výsledky obohacené o názor očního lékaře. Také zde uvedeme možné zlepšení a také řešení špičového programu v kameře ZEISS VISUCAM 224/524 a jak se takovému řešení přiblížit 6.

## Kapitola 2

# Anatomie, fyziologie a nemoci oka

Před začátkem skládání snímků sítnice by bylo podle mě vhodné probrat oko a jeho části a časté nemoci, který se bude na našich snímcích nacházet. Je důležité vědět popsat oko a jeho nejčastějších nemoci, abychom věděli co se na snímcích nachází.

### 2.1 Anatomie oka

Oko, *bulbus oculi*, má tvar koule. V průměru má u dospělého člověka přibližně 25mm. Pro orientaci lze na oční kouli popsat přední a zadní pól. Mediálně od zadního pólu oka leží výstup zrakového nervu. Oko je uloženo v očníci a skládá se z oční koule, obsahující dioptrický aparát, a z přídatných orgánů oka s funkcí ochrannou (víčka, spojivka a slzný aparát) a hybnou (okohybné svaly). Stěna oční koule je tvořena třemi základními vrstvami: povrchovou, střední a vnitřní vrstvou.

- Povrchová vrstva je vazivová vnější vrstva, kterou v zadních 5/6 oka tvoří bělima a v přední 1/6 průhledná rohovka.
- Střední, cévnatá vrstva oční koule má odzadu dopředu tyto části - cévnatka, řasnaté těleso a duhovka. Cévnatka vyživuje hluboké vrstvy sítnice a úponem na řasnaté těleso jej táhne dozadu a tím napomáhá udržet zaostření oka na dálku.
- Vnitřní vrstva oční koule je tvořena sítnicí. Sítnice ve své nejrozsáhlejší části, pars optica, obsahuje světločivné elementy tyčinky a čípky a další neurony. Směrem k řasnatému tělesu ubývá v sítnici počet tyčinek a čípků. Část sítnice, která přechází na zadní plochu duhovky, již neobsahuje tyčinky, čípky ani další neurony. Na sítnici, mediálně od zadního pólu oka, se nachází tzv. slepá skvrna - v této oblasti chybí v sítnici tyčinky a čípky. Laterálně od osy oka leží na sítnici žlutá skvrna, ve které jsou přítomny převážně čípky a jedná se proto o místo nejostřejšího vidění.

V oku se dále nachází čočka, orgán zavěšený na řasnatém tělísku, a sklivec, což je čirá, průhledná a světlolomná hmota s vysokým obsahem vody (až 98%), která vyplňuje oční komory, prostor za čočkou a řasnatým tělískem. Mezi pomocné struktury oka patří okohybné svaly, oční víčka, spojivka a slzný aparát. Tato podkapitola byla převzata z [27] a [13]

### 2.2 Fyziologie zraku

Zrak je definován jako schopnost přijmat a zpracovávat informace z okolí prostřednictvím viditelného světla dopadajícího na sítnici oka. Pro člověka je zrak nejdůležitějším smyslem



a dodává asi 70% všech informací. Jakékoliv poškození zraku je proto přijímáno mnohem tíže, než poškození či ztráta kteréhokoliv jiného smyslu.

Lidské oko vnímá elektromagnetické záření o vlnových délkách od 400nm (fialová) do 750 nm (červená) a ve velkém rozsahu intenzit (jasu). Dále je uzpůsobeno na rozlišování kontrastu, tj. rozdílů jasu nebo barevného odstínu sousedních ploch, a tím rozlišování obrysů. Roli při tom hrají nejen optické systémy oka, ale i vlastnosti sítnice a mechanismy dalšího zpracování obrazové informace.

Základem pro vznik zrakového vjemu je podráždění receptorů sítnice. Aby toto podráždění co nejpřesněji odráželo skutečnost, musí se optickým aparátem oka vytvořit na sítnici obraz pozorovaných předmětů. Paprsky vycházející z předmětu, nebo se od něj odrážející se na optických rozhraních oka, kterými jsou přední plocha rohovky a přední a zadní plocha čočky, lámou a vytvářejí na sítnici zmenšený a převrácený obraz. Největší podíl na optické lomivosti oka má přední plocha rohovky.

Světlo z blízkých předmětů dopadá do oka pod větším úhlem než ze vzdálených předmětů. Aby došlo k vytvoření obrazu předmětu na sítnici, musí být tyto paprsky více lámány. Úprava lomivosti optického prostředí oka podle vzdálenosti objektu spočívá ve změně tvaru čočky (akomodace). Tyto změny jsou ovládány svaly řasnatého tělesa a jejich vazy, které jsou zakotveny v čočce. Při pohledu do blízka se tahem za jeden ze svalů řasnatého tělesa (*musculus ciliaris*) uvolní závěsný aparát čočky a ta svojí pružností zaujme sférickější tvar, tedy více vyklenutý. Vzroste její optická mohutnost a paprsky se více lámou. Naopak při pohledu do dálky je sval relaxován a čočka je oploštěná.

Světločivými elementy jsou tyčinky a čípky sítnice. Čípků je přibližně 6 milionů. Umožňují vnímat odstín a intenzitu barvy. Tyčinek je zhruba 120 milionů a umožňují vnímat kontrast, což je důležité pro adaptaci na tmou. Tyčinky a čípky jsou synapticky spojeny s několika typy neuronů sítnice. Přes axony nervových buněk sítnice, které dohromady tvoří zrakový nerv, se informace ze sítnice dostávají do vyšších oddílů mozku. Axony nervových buněk z vnitřních polovin sítnice se na spodině mozku kříží (*chiasma opticum*). Do zrakové oblasti jedné hemisféry tak přichází informace ze stejné poloviny sítnice (nasální X temporální), tedy přibližně dva stejné obrazy. Zraková dráha je přepojena v talamu a končí v týlní oblasti mozkové kůry (primární zraková kůra).

Tato podkapitola byla převzata z [18]

## 2.3 Nejčastější choroby zraku

Kromě krátko zrakosti nebo dlouho zrakosti, fyzického poškození nebo méně závažných onemocnění, pokud se včas podchytí jako zánět spojivek nebo ječné zrno existují i další nemoci.

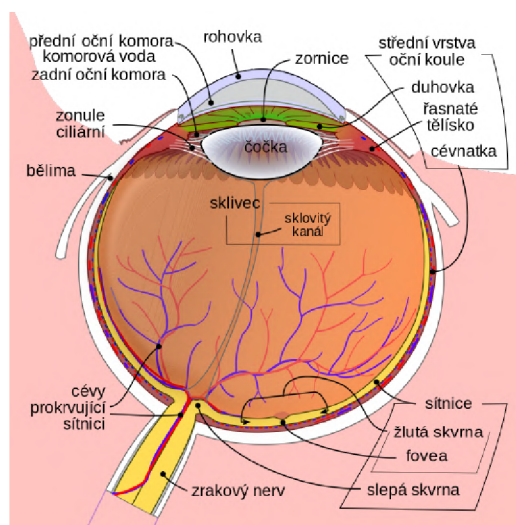
Šedý zákal (katarakta) je onemocnění, při kterém dochází ke snížení průhlednosti čočky. Prevalence je asi 50% u osob nad 50 let a asi u 75% osob nad 75 let. Terapie šedého zákalu je výlučně chirurgická a hlavní indikací k provedení operace je pokles zrakové ostrosti. Operace šedého zákalu je dnes prováděna ambulantně, trvá asi 20 minut a pacient může ihned po operaci sám odejít. Podstatou výkonu je odstranění zkalené čočky a ve většině případů její nahrazení čočkou umělou.

Šedý zákal může být i vrozený - buď dědičný nebo vzniklý působením toxických vlivů při vývoji plodu (virové záněty matky v prvním trimestru, radiační záření, jedy). Dále může být perinatální (vzniklý při porodu, v důsledku vrozených metabolických poruch), již zmiňovaný senilní (nejčastější), traumatický (při poranění oka, kdy je poškozeno pouz-

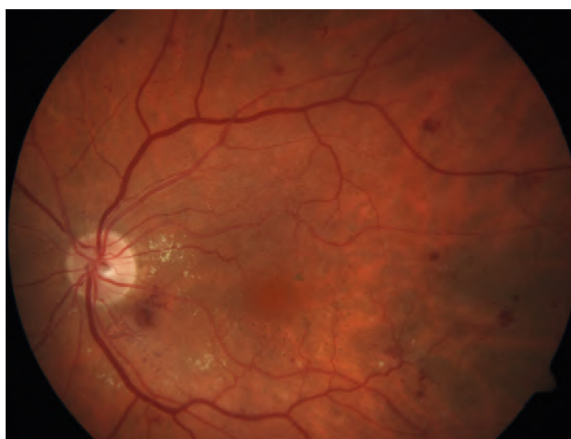
dro čočky), iatrogenní pooperační, vzniklý chemickým působením, radiací nebo v důsledku metabolických onemocnění, jako například diabetu.

Zelený zákal (glaukom) je onemocnění, které se nejdříve projevuje drobnými výpadky v zorném poli, a může skončit až úplnou ztrátou vidění. Podstatou vzniku tohoto onemocnění je zvýšení tlaku nitrooční tekutiny, vedoucí k útlaku vláken zrakového nervu, která postupně odumírají. Dále dochází k narušení prokrvení sítnice a zrakového nervu. Základem farmakologické a chirurgické léčby je zlepšení odtoku nitrooční tekutiny z očních komor. Rizikovým faktorem pro vznik glaukomu, stejně jako katarakty, je vyšší věk. Glaukom však může být i vrozený, vzniklý v důsledku nevyvinutí odtokových cest komorového moku. Může vznikat i v důsledku nitroočních zánětů a tumorů, terapie glukokortikoidy, či traumatu oka.

Tato podkapitola byla převzata z [14]



Obrázek 2.1: Na snímku můžeme vidět průřez oka s popisem jeho jednotlivých částí.<sup>1</sup>



Obrázek 2.2: Na snímku můžeme vidět příklad snímků sítnice, které budeme skládat.<sup>2</sup>

<sup>3</sup>Snímek je z databáze v rámci projektu Viderize.

<sup>1</sup>Snímek byl převzat z [https://upload.wikimedia.org/wikipedia/commons/thumb/a/a3/Schematic\\_diagram\\_of\\_the\\_human\\_eye\\_cs.svg/600px-Schematic\\_diagram\\_of\\_the\\_human\\_eye\\_cs.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/a/a3/Schematic_diagram_of_the_human_eye_cs.svg/600px-Schematic_diagram_of_the_human_eye_cs.svg.png)

## Kapitola 3

# Metody pro skládání snímků

Nejdříve jsem si musel vyhledat vůbec co všechno je potřeba pro skládání snímků. Tento článek [1] jsem si nastudoval jako první abych ses vůbec zorientoval v problému skládání snímků. Je zde zmíněno popsán základní princip skládání snímků a i některé algoritmy a metody použité pro skládání. Článek popisuje skládání snímků pro panoramata, což v našem případě se jedná spíše o mozaiku, ovšem základní principy jsou shodné. Rozebereme si metody, které využíváme ve skládání snímků a také potřebnou teorii.

### 3.1 Počítačové vidění

Zrak je naším nejdůležitějším smyslem. Podle odhadů získáváme 80% všech informací zrakem. Lidé jsou svým zrakem schopni rozpoznat velikost objektů, jejich barvy a strukturu, zrak nám pomáhá při orientaci v prostoru. To je dáno tím, že máme dvě oči, na které dopadají různé obrazy a z nich je náš mozek schopen všechny tyto informace získat.

Dalším smyslem, který je díky množství informací, které jeho prostřednictvím získáváme, druhým v pořadí důležitosti, je sluch. Dokážeme určit sílu zvuku, směr odkud přichází, pokud však nejsme oklamáni například ozvěnou.

Když porovnáme, o co dříve jsme byli schopni nahrávat, ukládat a přehrávat zvuk ve srovnání s vyfocením, uložením a zobrazením digitálního snímku, zjistíme, že rozdíl činí přes 80 let. Přenést to, co vidíme, do počítače bylo mnohem náročnější než přenést to, co slyšíme. Dnes jsme schopni zaznamenávat širší zvuková pásma, než jsme schopni slyšet, a perfektně rekonstruovat zvuk v prostoru. Při prezentaci toho, co vidíme, pomocí počítače jsme ochuzeni o mnoho informací. Fotoaparáty vyfotí snímek, na kterém ovšem není přesně stejně tak zbarvený stůl, jak ho vidíme my. Fotoaparáty vyfotí snímek, na kterém nepoznáme jakou přesně strukturu má stolní deska. Námi natočený sprinter je na videu rozmazaný, ovšem my jsme ho viděli perfektně. Dat, která musíme uložit a zobrazit, abychom dobře viděli, je mnohem více, než dat, která potřebujeme uložit a přehrát, abychom slyšeli. Jejich kvalita je také rozdílná. Zvuk nahraný na studiový mikrofon a přehrávaný do kvalitních sluchátek je vynikající. U fotografie je problémem už i zaznamenat přesnou barvu objektů, natož velikost objektu bez kontextu, vzdálenost objektu nebo jeho strukturu.

Počítačové vidění je obor zabývající se transformací dat ze zařízení, které zaznamenává obraz nebo video do digitální podoby, a jejich další interpretací, jedná se například o převedení snímku do černobílé podoby, identifikování objektů na snímku, počet objektů na snímku, otočení snímku nebo vyříznutí osob na snímku. Lidský mozek všechny tyto operace automatizované a ani si neuvědomuje jak je provádí.

Bohužel počítače takto nepracují. Všechno, co počítače vidí, je pouze mřížka čísel. Víc toho počítač „nevidí“. Z mřížky čísel je bez určitých algoritmů nemožné říci, co počítač vidí. Re prezentaci této mřížky počítač zvládne a zobrazí, co se nachází na snímku. Interpretace je ovšem opět na nás a také počítač netuší, co zobrazil. Dále se potýkáme s problémem při pořizování těchto snímků. Šumy, rozmazání, odrazy a jiné kvalitu snižující defekty, které vznikají při zaznamenávání světa kolem nás, znamenají problém, který formálně nejde vyřešit.

*"Ve skutečnosti je problém, který jsme výše představili, ještě těžší; je formálně neřešitelný. Mějme 2D zobrazení 3D světa. Neexistuje žádný unikátní postup, který by umožnil zrekonstruovat původní 3D signál z takového zobrazení. Formálně řečeno, takto neúplně zadaný problém nemá žádné unikátní definitivní řešení. Totéž 2D zobrazení by mohlo reprezentovat kteroukoliv z nekonečných kombinací 3D scén, a to i za předpokladu dokonalých vstupních dat."*[6]

Naším úkolem je „sešít“ jednotlivé snímky sítnice dohromady, vytvořit z nich mozaiku. Abychom to dokázali, budeme muset aplikovat určité algoritmy, které se v oblasti počítačového vidění využívají a o kterých pojednáme dále.

## 3.2 Snímek, jeho formát a práce s pozadím

Snímek ze záznam reality v určitém minulém čase, která se nachází v digitální podobě na paměťovém zařízení v počítači. To, v jaké podobě je uložen, je určeno formátem snímku. Nyní uvedeme, jaké možnosti existují a jaký formát jsme se rozhodli použít v této práci.

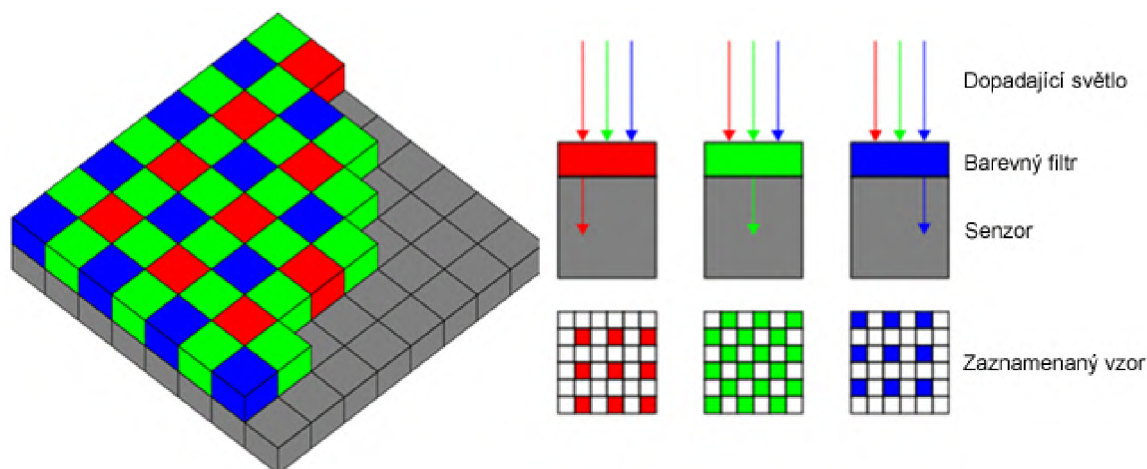
### 3.2.1 DICOM

DICOM neboli digitální zobrazování a komunikace v medicíně je způsobem sdílení a ukládání medicínských snímků. Scany z CT, MRI, ultrazvuku nebo v našem případě snímky sítnice jsou uloženy v tomto formátu. Jedná se o standard, který je velmi rozsáhlý. Konkrétně se jedná o ISO standard 12052:2006. Důvodem k využití tohoto standardu je, že soubor neobsahuje pouze snímky, ale také další dodatečné informace v hlavičce a také udává, jak budeme s těmito daty pracovat, jak se budou ukládat a sdílet, aby byla nejen nemocničním zařízením, ale i při jejich spolupráci i na mezinárodní úrovni ulehčena práce. Hlavička obsahuje meta-data, například jméno pacienta, typ snímku, typ vyšetření a podobné informace, které se mohou hodit lékařům. Konkrétní snímek je dále uložen ve formátu bez komprese takzvaný RAW neboli surový, což znamená, že je neupravený nebo JPEG, JPEG lossless a další.[5]

### 3.2.2 RAW

Pokud něco vyfotografujeme, všechno odražené světlo putuje na senzor, který má určitý počet buněk, což je udáváno počtem pixelů u každé kamery. Tyto pixely jsou však barvoslepé, určují pouze intenzitu světla, které dále A/D převodník převede na bity. Vzniká otázka, jak je to možné, že kamery vlastně fotí barevné fotografie. To nám umožňují filtry, které se nacházejí nad každou buňkou senzoru. Nejrozšířenější filtry, které se nad buňkami senzoru nacházejí, jsou RGB. To znamená, že například 36 MPix senzor obsahuje filtry v poměru 1:2:1. Červených filtrů je 6 milionů, zelených je 12 milionů a modrých 6 milionů. Tomuto uspořádání barevných filtrů se říká Bayerova maska. Poté dojde Bayerovou interpolací k vypočtení barvy pro každý pixel z okolních pixelů.[17] Formát RAW je formát, který neprojde žádnou kompresí ani jinou digitální úpravou. Všechny tyto operace můžeme provést

následně my při úpravě snímků pro naše potřeby. Samozřejmě je zde nevýhoda velikosti snímků. [21]



Obrázek 3.1: Bayerova maska a znázornění způsobu zpracování dopadajícího světla.<sup>1</sup>

### 3.2.3 JPEG

JPEG je dalším formátem ukládání snímků. Mezi hlavní výhody JPEG formátu patří komprese, která se ovšem snaží co nejvíc zachovat původní kvalitu. Jelikož výsledek této práce by využívali především lékaři, kteří budou spíše dbát na kvalitu snímků oproti velikosti nebo rychlosti přenosu, není pro naši práci tento formát vhodný. [11]

### 3.2.4 PNG

PNG je formát, který jsem se rozhodl používat v této práci, a proto všechny snímky převádím do tohoto formátu. Důvody, proč jsem se rozhodl používat tento formát, jsou především dva. Prvním je bezztrátová komprese. PNG vznikl pro rastrovou grafiku, aby bylo možné ji ukládat bezztrátově. Jelikož potřebujeme co největší kvalitu snímků, ovšem pokud možno s menší velikostí souborů než je formát RAW, je formát PNG ideálním řešením. Kvalita snímků je pro nás, respektive pro lékaře, kteří kontrolují snímek, důležitá. Seběmenší skvrny mohou znamenat onemocnění, a čím dříve jsme je schopni zachytit, tím lépe. Nечceme, aby kvůli kompresi snímků lékař něco takového přehlédl a potenciálně došlo k zanedbání léčby.

Dalším důvodem použití PNG formátu je alfa kanál, který je součástí RGBA. Alfa kanál je další složkou pixelu, která udává průhlednost. Ta funguje stejně jako zbylé složky RGB, neboť jsou různé stupně průhlednosti. Tuto vlastnost potřebujeme, abychom byli schopni snímky přes sebe skládat a provádět jejich sešití. Jediný formát, ve kterém se to dařilo, byl PNG. [25]

<sup>1</sup>Snímek byl převzat z [https://geoinformatika-1.vsb.cz/mms/images/cv02/bayer\\_mask.png](https://geoinformatika-1.vsb.cz/mms/images/cv02/bayer_mask.png)

<sup>3</sup>Snímek byl převzat z [https://www.pinpng.com/pngs/m/5-50635\\_hue-alpha-rainbow-gradient-png-transparent-png.png](https://www.pinpng.com/pngs/m/5-50635_hue-alpha-rainbow-gradient-png-transparent-png.png)



Obrázek 3.2: Na snímku můžeme vidět postupně klesající hodnotu alpha kanálu od shora dolů, kdy je snímek průhlednější a průhlednější a můžeme jasněji vidět pozadí.<sup>3</sup>

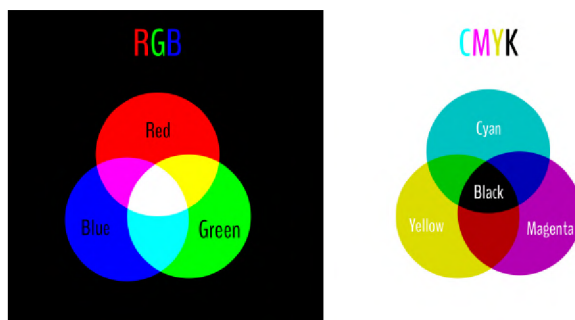
### 3.2.5 Barevné modely

Barevné modely je způsob reprezentace barev v počítači, kdy je barva popsána podílem jednotlivých složek. Ty tvoří dohromady barevný prostor. Pro tisk barev na hmotných věcech se používá model CMY nebo CMYK. V digitální podobě se používá model RGB nebo méně používaný HSV.

### 3.2.6 Míchání barev

Existuje dvojí způsob míchání barev. Subtraktivní míchání barev, které funguje na principu míchání barev, kdy platí, čím více barev přidáme, tím více světla pohltí. To znamená, abychom vytvořili černou barvu, potřebujeme použít všechny barvy, aby bylo světlo pohlceno a žádné se od této plochy neodrazilo. Model CMY nebo CMYK se skládá z barev azurové, purpurové a žluté případně černé. Tato metoda subtraktivního míchání barev se využívá v tisku.

Naopak aditivní míchání barev funguje přesně naopak. Čím více přidáváme barevných složek, tím více barev zobrazujeme, a tím je i barva světlejší. Pokud přidáme všechny barvy, zobrazíme bílou barvu. Tak fungují monitory a jiná zobrazovací zařízení, jako je například projektor, které využívají světlo pro vytváření barev. V digitální fotografii využíváme aditivního míchání barev, tudíž se subtraktivnímu míchání barev už nebudeme podrobněji věnovat.[8]



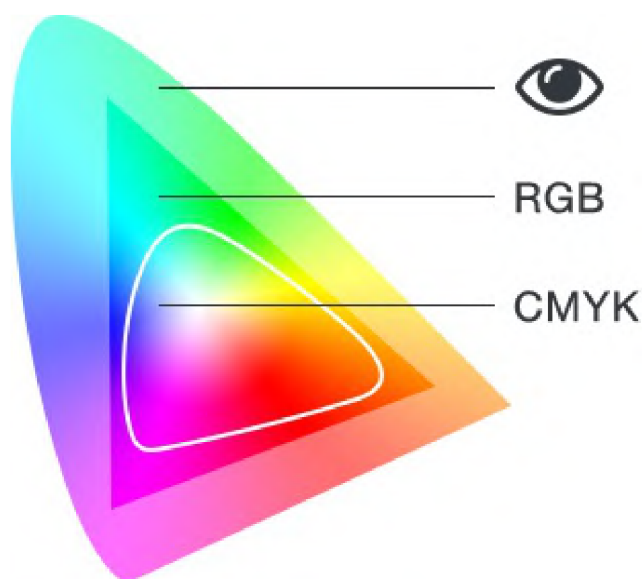
Obrázek 3.3: Na snímku je zobrazeno aditivní a subtraktivní míchání barev.<sup>4</sup>

### 3.2.7 Model RGB

Model RGB se skládá ze 3 základních barev, a to červené, zelené a modré. Přesné vlnové délky těchto barev jsou definovány mezinárodně. Jedná se o aditivní míchání barev. To znamená, že kombinací všech těchto barev získáme bílou barvu, oproti subtraktivnímu míchání barev, například CMY, které se používá například při tisku mícháním barev azurové, purpurové a žluté. Smícháním těchto 3 barev získáme černou, což je opakem aditivního míchání barev. My se soustředíme na aditivní míchání barev, neboť to se používá v zobrazování například monitorem.

Model RGB je tedy kombinací 3 barev, o kterých jsme se zmínili výše. Dále zde potřebujeme určit intenzitu těchto barev. Mohutnost se udává barevnou hloubkou, která je definována pevným počtem bitů pro každou barvu stejně. Nejpoužívanější bitová délka je tedy 8 bitů pro každou barvu. Tím pádem máme 24bitovou barevnou hloubku. Na 8 bitech jsme schopni zobrazit 256 odstínů jedné barvy (od 0 včetně po 255 včetně). 24bitová barevná hloubka poskytuje možnost zobrazit 256 stupňů šedi. Jedná se o nejpoužívanější barevný model pro digitální reprezentaci barev.

Ovšem jak bylo zmíněno výše, existuje ještě varianta RGB modelu, která se nazývá RGBA model. Tento model má další složku, kterou však není barva, ale průhlednost, takzvaný alfa kanál. Každý pixel má svoji vlastní průhlednost. V této práci využíváme RGBA model, protože průhlednost určitých pixelů (pixel je nejmenší jednotkou digitálního snímku, z kterých je složený celý snímek, a tento jeden bod je reprezentovaný modely, které jsou zmíněny dříve například RGB nebo HSB) pomáhá při skládání snímků jeden přes druhý. Potřebujeme, aby pozadí jednoho snímku bylo průhledné a mohli jsme zobrazit další snímek, který se nachází tzv. „pod ním“. [8]



Obrázek 3.4: Na snímku je zobrazeno celé spektrum barev, které je zdravé lidské oko schopno vidět a podmnožiny barevného spektra, která jsou schopna zobrazit barevné modely RGB a CMYK.<sup>6</sup>

<sup>4</sup>Snímek byl převzat z <https://www.onlinefotoskola.cz/obrazky/305.jpg>

<sup>6</sup>Snímek byl převzat z <https://www.xart.cz/images/stories/Blog/Data-pro-tisk/Color-gamut.jpg>

### 3.2.8 Práce s pozadím

Snímky, s kterými pracujeme, obsahují černé pozadí, které by činilo problém při skládání, protože by nebyla možná návaznost jednoho snímku sítnice na druhý. Prvním způsobem, který se nabízel, bylo, že bychom ořezávali snímky podle předem definovaného tvaru a velikosti. To je ovšem problém z důvodů, že v mnohých prozkoumaných databázích v rámci projektu Viderize jsou nejméně dva druhy černého pozadí, tudíž by nemohla být předdefinována metoda ořezu. Tento fakt znemožňuje také použití alfa masky, což je metoda, kdy definujeme masku, kde je pouze určitá část průhledná, to znamená je nastaven hodnota alfa kanálu na 0 v určité oblasti, a to umožňuje, že poté, co aplikujeme alfa masku na jiný snímek, budeme mít snímek pouze v průhledné části alfa masky. To by se dalo použít tak, že bude vytvořen pouze snímek sítnice bez černých okrajů.

Dalším způsobem, který mě napadl a po prozkoumání databáze jsem se rozhodl ho použít, je nalezení všech černých pixelů a nastavení jejich alfa kanálu na hodnotu 0, tzn. průhledné. Všechny snímky obsahují černé pozadí, bez výjimky. Tohoto faktu mě napadlo využít, jelikož procházení snímku jako matice

je jednoduché a zjištění, že se jedná o černý pixel, a změnit jeho hodnoty, rovněž není složité. Tato metoda se mi zdá jako nejjednodušší a nejefektivnější. Dále ale musíme počítat s faktem, že černé pixely se mohou objevit i v částech sítnice. Tam je ovšem velice nepravděpodobné, že bychom našli opravdu černý pixel, a to z toho důvodu, že i tmavší objekty na sítnici například různé onemocnění či poškození nemají úplně černou barvu. Jsou spíše zbarveny do tmavě zelené nebo modré nebo tmavě červené, nikoliv do úplně černé. K tomuto ovšem potřebujeme snímek, který bude v RGBA modelu, abychom mohli přenastavovat jeho alfa kanál.



Obrázek 3.5: Na snímku je zobrazeno ořezaný snímek, bez černého pozadí vytvořením projití celého snímku a nastavení alfa kanálů na 0. Zajímavostí je, že pozadí nemělo hodnoty RGB 0,0,0 ale spíše okolo 10-15 u každé složky.<sup>8</sup>

<sup>8</sup>Snímek je z databáze v rámci projektu Viderize.



### 3.3 Detekce klíčových bodů, jejich popis, hledání shodných bodů a příkladů algoritmů

Následující informace v 3 podkapitolách jsou z [23] Rozhodl jsem se pro tento složitější nadpis, protože tyto oblasti spolu velice souvisejí. V následujících podkapitolách si zkusíme vysvětlit provázanost těch operací se snímky a význam jednotlivých dílčích operací. V úvodu lehce vysvětlíme operace, které dohromady tvoří postup při složení dvou obrázků. „Během fáze detekce vlastností (extrakce), jsou v každém obrázku vyhledány klíčové body, u kterých se dá předpokládat, že půjdou lehce rozpoznat na ostatních snímcích. Fáze definice vlastností, je každá oblast okolo klíčových bodů detekovaných v předešlé fázi zkonvertována do kompaktnějšího a stabilnějšího (invariantního) deskriptoru, který lze porovnat s ostatními deskriptory. Fáze hledání vzorů pak efektivně vyhledává pravděpodobně shodné kandidáty v ostatních snímcích.[23]

Dalšími kroky tohoto procesu, který vyhledá body, popíše je a najde korespondující body, je výpočet transformace snímku, aby korespondující body na sebe seděly a měly stejnou polohu, tudíž ze dvou částečně korespondujících snímků vznikne nový snímek, kde bude korespondující částí průnik zobrazený jednou a zbytek bude tvořit doplněk. Nebude tedy nutno zobrazovat dva snímky, ale vše zobrazíme v jednom. Tuto operaci budeme opakovat do té doby, pokud bude co přidávat, ale místo nutnosti zobrazení 7 snímků bude lékař schopen vše zobrazit na jednom.

#### 3.3.1 Definování a detekce klíčových bodů

Pokud se zkusíme zamyslet, které lokace snímků se dají použít jako klíčové body snímků, které bychom mohli najít v částečně korespondujících snímcích, brzy přijdeme na to, že oblasti bez textury nebo jednotné barvy bychom označit jako klíčové body nemohli. Příkladem je obloha, která je velmi monotónní. Pokud chceme najít klíčové body, budeme se muset zaměřit na oblasti, které jsou něčím zajímavé, něčím vynikají. Takové body jsou většinou oblasti velké změny kontrastu. Jedná se o oblast s rohem, s hranami nebo body neboli skvrnami. Pouze tyto body se využívají při skládání snímku. Nebylo by rozumné skládat snímky jako celek. Způsobů detekce je několik a my je konkrétně vyjmenujeme níže.

Klíčové body musejí být něčím rozdílné, něčím vystupovat oproti okolí, jak bylo již zmíněno výše, ovšem jen toto nestačí. Například rohů může být ve snímku mnoho, což by následně při nalezení stejných rohů mohlo být nepřesné, proto se využívá i okolí těchto snímků a používají se takzvané deskriptory popsané v kapitole 3.3.2, které vezmou okolí bodů a popíší je a následně se při hledání korespondujících znaků mezi

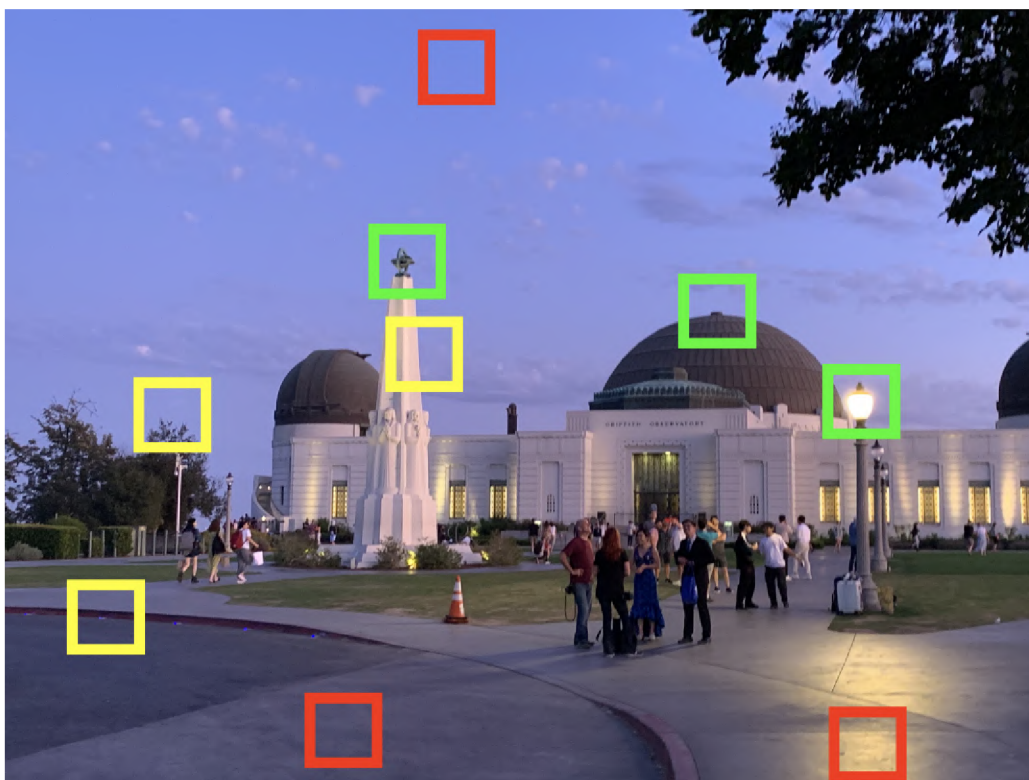
snímky porovnávají tyto mnohem na informace bohatší body. Dalším důležitým požadavkem, je invariantnost těchto bodů vůči změnám jasu, zvětšení snímku nebo naopak zmenšení, různé rotace a další úpravy. Dále si uvedeme příklady algoritmů hledajících tyto klíčové body. Důležitý je samozřejmě počet klíčových bodů. Nedá se říci, čím více tím lépe. Čím více klíčových bodů, tím více může vzniknout chyb při hledání korespondujících bodů, tudíž se musíme zaměřit na kvalitní body a nesoustředit se pouze na počet. Samozřejmě je také problém, když je bodů málo. Následně vypočítaná transformační matice nebude přesná a výsledek bude zničen.

### 3.3.2 Popis klíčových bodů

Dalším krokem je popsání těchto bodů. Body popisuje deskriptor. Jelikož je bod malý a neobsahuje dostatečnou informační hodnotu, využijeme i jeho okolí. Zde se přístup liší. Některé takzvané deskriptory popisují vždy stejně velikou oblast, některé deskriptory mají adaptivní velikost okolí kolem klíčového bodu. Důležité je, abychom při popisu uváděli okolí bodu dostatečně robustně a popis byl invariantní vůči měřítku, změně jasu nebo posunům a dalším transformacím viz snímek. 3.6

### 3.3.3 Hledání shodných bodů

Dostáváme se ke třetí fázi, kde budeme hledat shodné body. Využijeme faktu, že máme všechny klíčové body popsány a nyní můžeme porovnávat. Musíme ovšem počítat s problémem, že najdeme i body, které k sobě nepatří ovšem tuto problematiku budeme rozebírat až dále.



Obrázek 3.6: Na snímku je jsou zobrazeny oblasti ve čtvcích různých barev. Zelená barva jsou oblasti, které jsou ve snímku zcela unikátní, jsou velice kontrastní a jasně vyčnívají ve snímku a jsou dostatečně robustní. Žluté čtverce označují horší oblasti pro identifikaci ve snímku, ovšem v případě dobrého deskriptoru se je nám podaří nalést. V červených čtvcích se nacházejí oblasti, které nejsou klíčové. Jsou jednoduté, bez změn kontrastu. Většinou se jedná o oblohy, trávníky nebo podobné jednoduté objekty, které robustní nejsou. V případě snímků sítnice se jedná například o pozadí sítnice, které je téměř vždy jednoduté červené až oranžové.

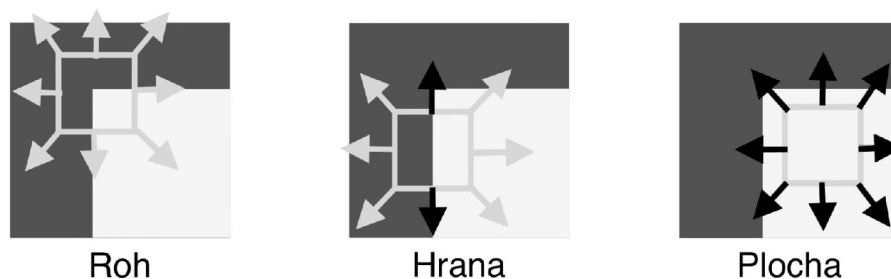
Nyní si řekneme o konkrétních způsobech hledání a popisování klíčových bodů využívaný v počítačovém vidění.

### 3.4 Harrisův detektor rohů

Jedná se o jeden z nejstarších a nejjednodušších, ale z důvodů široké využitelnosti dnes stále používaných detektorů klíčových bodů v případě panoramat. Zkusíme si blíže přiblížit vlastnosti Harrisova detektoru rohů.

Na úvod je nutno definovat roh. Rohem je pro Harrisův detektor bod, který spojuje 2 a více hran. Harrisův detektor hledá tedy rychlé změny jasu v určitém směru. Tudíž hledání klíčového bodu v případě Harrisova detektoru rohu probíhá tak, že se na snímek díváme oknem a v případě, že při pohybu každým směrem dojde ke změně intenzity, jedná se roh. Detailem tohoto faktu je, že hrany tento algoritmus neindikuje, protože při posunu ve směru hrany nedojde ke změně jasu.

Využití tohoto detektoru v našem případě by ovšem mělo jisté problémy, které popíšeme níže, a to i přes odolnost toho detektoru vůči rotacím a posunům. Problémem je invariantnost vůči jiným transformacím, například náklonům. Dále je zde velký počet bodů na vlasečnicích (nejjemnější cévy), kterých je na snímků nepřehledné množství a skládat snímky pomocí těchto vlasečnic nepřinese dobrý výsledek. Náš detektor by se potřeboval soustředit na větší cévy, kterých je méně a budeme si moci být jistější, které body spolu korespondují, a také se soustředit na okolí zornice, kde se všechny tyto větší cévy scházejí, a podle nich budeme moci složit 2 snímky, kde se objeví zornice kvalitněji a přesněji. Dále víme, že pro tuto metodu není definován deskriptor, ale že existují metody, které deskriptor definovaný mají, proto bychom raději využili tyto metody. Z těchto důvodů není Harrisův detektor rohů pro naše účely ideální, neboť existují lepší varianty. Tato podkapitola byla čerpána ze článku autora metody [10]



Obrázek 3.7: Na snímku je zobrazeno hledání rohů. Můžeme vidět, že v případě, že se v okně nachází roh, dochází ke změnám kontrastu ve všech směrech a v případě, že se nachází na ploše nedochází ke změně kontrastu v žádném směru. Zajímavý je případ hrany, kdy dochází ke změně ve všech směrech kromě směru hrany. Toto nemůže považovat za klíčovou oblast, jelikož se může nacházet kdekoli na konkrétní hraně.<sup>10</sup>

<sup>10</sup>Myšlenka snímek byla převzata z [HarrisCornerIllustration](#)

<sup>12</sup>Snímek je z databáze v rámci projektu Viderize.



Obrázek 3.8: Na snímku jsou zobrazeny rohy, které nalezneme v případě hledání rohů ve snímcích sítnice. <sup>12</sup>

### 3.5 SIFT

Pro porozumění metody SIFT jsem využíval [23] a samotného autora metody [16].

Při hledání jiného detektoru je možno narazit na dvě neznámější a nejpoužívanější metody, jsou to SIFT a SURF. Pro náš účel bude vhodnější SIFT (Scale Invariant Feature Transform), jelikož metoda SURF (Speeded-Up Robust Features) využívá z důvodu zrychlení oproti SIFT často pro real-time aplikaci, kterou nevytváříme, ačkoli jsou základní principy stejné. Detekované klíčové body jsou hodně výrazné, tudíž máme možnost nalézt a spárovat správně tyto klíčové body i ve větším množství.

Metoda SIFT je rozdělena na 4 etapy:

- scale-space detekce extrémů
- lokalizace klíčových bodů, v našem případě extrémů,
- přidělení orientace, kdy je přidělena jedna nebo více orientací každému klíčovému bodu
- popis klíčového bodu

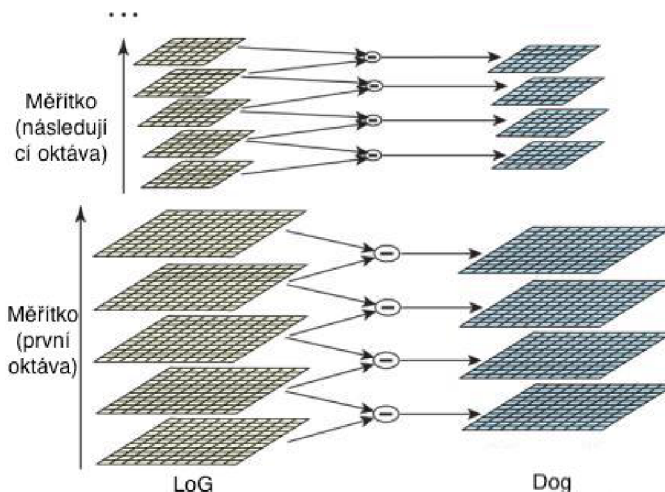
### 3.5.1 Scale-space detekce extrémů

"Objekty kolem nás jsou důležité v různém měřítku. Například strom jako celek má určitý tvar, kdy při pohledu z dálky vidíme korunu a kmen, když se k němu přiblížíme, uvidíme určité větve a že na větvích jsou listy, už nevnímáme ani tvar koruny a kmene, ale vnímáme jednotlivé větve a listy na nich. Takto můžeme pokračovat až k atomům a i dále." [15]

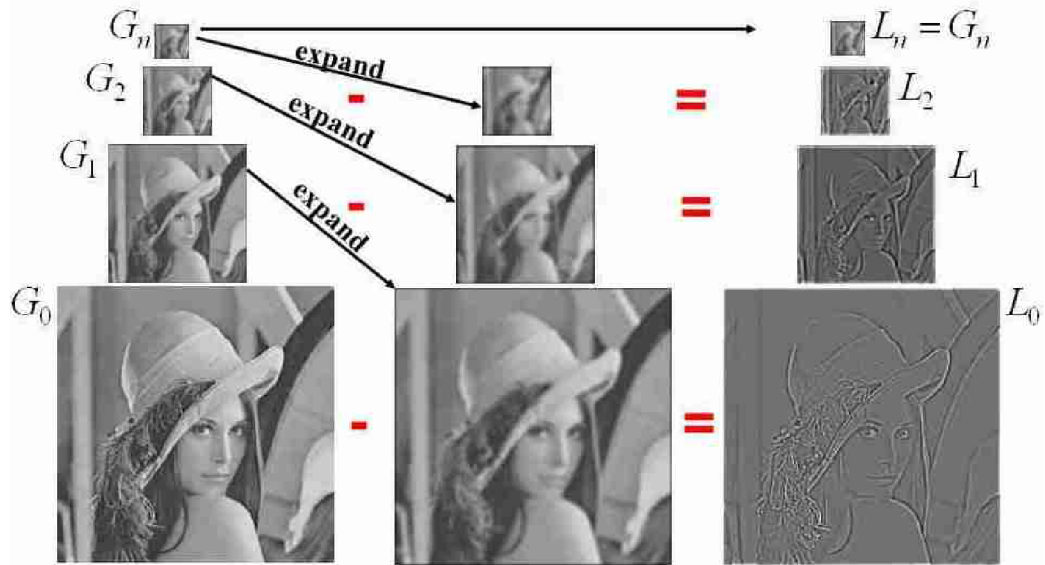
K detekci extrémů se v metodě SIFT využívá metody DoG neboli Difference of Gaussians. Difference of Gaussians je vylepšením metody LoG (Laplacian of Gaussian) ale jedná se pouze o její aproximaci. Důvodem tohoto vylepšení bylo zrychlení. Tyto metody jsou stejné až na jeden krok, který Difference of Gaussians obsahuje. Prvním krokem je rozmazání snímků pomocí Gaussovy funkce. Rozmazání je postupné a s jinou odchylkou  $\sigma$ . Tudíž vezmeme snímek I a rozmazáme ho Gaussovou funkcí s odchylkou  $\sigma$ . Tento postup aplikujeme 4krát s postupně rostoucí odchylkou  $\sigma$ , tudíž i větším rozmazáním, a vznikne sada 4 snímků, které se říká oktáva. Kdybychom používali metodu LoG, pokračovali bychom, zmenšili snímek na polovinu a vytvořili znovu oktávu. Ovšem v metodě DoG vezme 4 snímky, které nám vznikly, a sousední od sebe odečteme. Vzniknou 3 snímky na rozdíl od původních 4 snímků. D. G. Lowe, autor metody SIFT, doporučil 3 snímky vzniklé metodou DoG jako dostačující. Poté co nám vznikne jedna oktáva, zmenšíme snímek na polovinu a celý proces postupného rozmazávání Gaussovou funkcí opakujeme. Takto se vytvoří pyramida s oktávami po třech snímcích, kde každé vyšší patro je oktávou 3 snímků o polovinu zmenšených než patro předcházející.

Důvodem použití Gaussovy funkce je, že se jedná o lineární funkci, rozmazávání probíhá ve všech směrech stejně (symetricky), zachovává souřadnice extrémů a je rychlejší než jiné, nelineární filtry. DoG je oproti LoG rychlejší z důvodu konvoluce druhé derivace Gaussovy funkce, která je výpočetně náročná. Ještě si uvedeme Gaussovou funkci  $G(x, y, \sigma)$ :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



Obrázek 3.9: Na snímku vidíme Scale-space. Také vidíme rozdíl mezi DoG metodou a LoG metodou. Můžeme pozorovat i oktávy.<sup>14</sup>



Obrázek 3.10: Na snímku vidíme jak vypadá odečtení dvou snímků filtrovaných Gaussovou funkcí a tím pádem jak nám vzniká DoG. Vidíme, že snímek má zachované především hrany a rychlé změny kontrastu.<sup>16</sup>

### 3.5.2 Lokalizace klíčových bodů, v našem případě extrémů

Dalším krokem je hledání extrémů, buď minima, nebo maxima, což probíhá následovně. Vezmeme každý bod snímků vzniklých metodou DoG a porovnáme sousedních 8 a 9 bodů v sousedních měřítcích, viz. ilustrace 3.11. Pokud se bude jednat o extrém, označíme tento bod za klíčový. Ovšem problematikou této metody je, že vznikají nestabilní body podél hran. Tyto body odfiltrujeme Hessiánovou maticí, tudíž vznikne ještě lepší množina klíčových bodů. Metoda SIFT aplikuje i další metody vyřazování a upřesňování polohy klíčových bodů.

### 3.5.3 Přidělení orientace

Abychom zajistili jednu z důležitých vlastností klíčových vlastností, je nutno zajistit invariantnost vůči rotaci snímků. To jsme schopni zajistit přidělením orientace ke každému klíčovému bodu. Jako nejlepší přístup k přidělení orientace se ukázal tento. Vybere se vyhlazený snímek Gaussovou funkcí  $L$ , který má nejbližší měřítko, tím pádem všechny výpočty probíhají měřítkově nezávisle. Funkce  $L$  je rozdíl dvou sousedních snímků, které byly rozmazány Gaussovou funkcí  $G(x, y, \sigma)$  viz. snímek 3.9. Pro každý snímek se přepočítá použitím difference pixelů  $L(x, y)$  v jeho měřítku, velikost gradientu  $m(x, y)$  a v poslední řadě jejich orientace  $\theta(x, y)$ :

<sup>14</sup>Snímek převzat z [https://opencv-python-tutroals.readthedocs.io/en/latest/\\_images/sift\\_dog.jpg](https://opencv-python-tutroals.readthedocs.io/en/latest/_images/sift_dog.jpg)

<sup>16</sup>Snímek převzat z [https://cdn-images-1.medium.com/max/1000/0\\*Q1rpQiMMZkcpWJen.jpg?q=20](https://cdn-images-1.medium.com/max/1000/0*Q1rpQiMMZkcpWJen.jpg?q=20)

<sup>18</sup>Snímek převzat z [https://opencv-python-tutroals.readthedocs.io/en/latest/\\_images/sift\\_local\\_extrema.jpg](https://opencv-python-tutroals.readthedocs.io/en/latest/_images/sift_local_extrema.jpg)

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2},$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))).$$

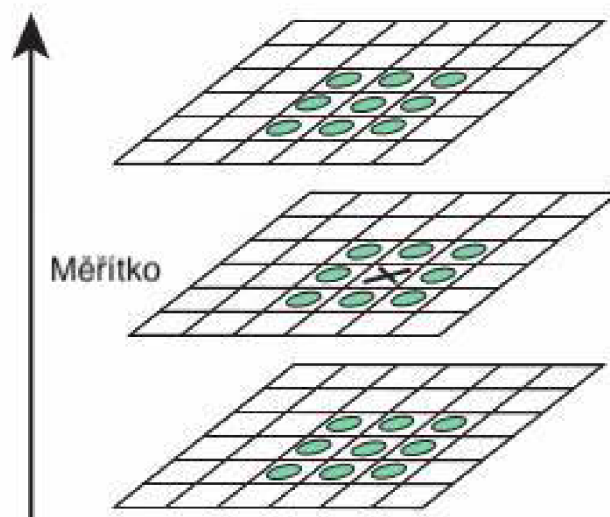
Dalším krokem je sestavení histogramu orientací. Ten se skládá z gradientů, které se nacházejí poblíž klíčového bodu, pro které jsme prováděli výpočty výše. Histogram se skládá z 36 zásobníků, které pokrývají 360 stupňů okolo klíčového bodu. Každý element, který je vložen do těchto zásobníků podle orientace, je váženě zprůměrován velikostí gradientu a součinitelem Gaussova okna ve tvaru kruhu s průměrem 1,5krát větším než měřítko významného bodu.

Zajímavostí tohoto postupu je, že z původního jednoho mohou vzniknout dva klíčové body. Ty se nacházejí na stejném místě, ale mají jinou orientaci, což podle D. G. Lowe výrazně zlepšuje robustnost celého programu na hledání a spojování klíčových slov. V histogramu se bude nacházet jeden zásobník, který bude mít největší hodnotu, a ten určuje orientaci, ovšem pokud jakýkoliv jiný zásobník bude mít hodnotu alespoň 80% nejvyšší hodnoty v histogramu, vzniknou 2 klíčové body. Tento jev se stává přibližně 15% klíčových bodů, ovšem je důležité, že tento algoritmus je zahrnut v metodě SIFT a zlepšuje hledání

klíčových bodů. Posledním krokem je proložení paraboly třemi nejbližšími hodnotami v okolí nejvyšší hodnoty orientace, a to pro ještě lepší přesnost orientace.

### 3.5.4 Popis klíčového bodu

Kroky, které jsem zatím provedl, jsou dostačující k invarianci vůči 2D operacím. Získali jsme lokaci klíčového bodu, měřítko, které zajistí invariantnost vůči změně měřítka a orientaci, která zaručí invariantnost vůči rotacím. Metoda SIFT je použita i na afinní transformace a změny úhlu snímacího zařízení, které pro charakteristiku našeho zadání potřebujeme.

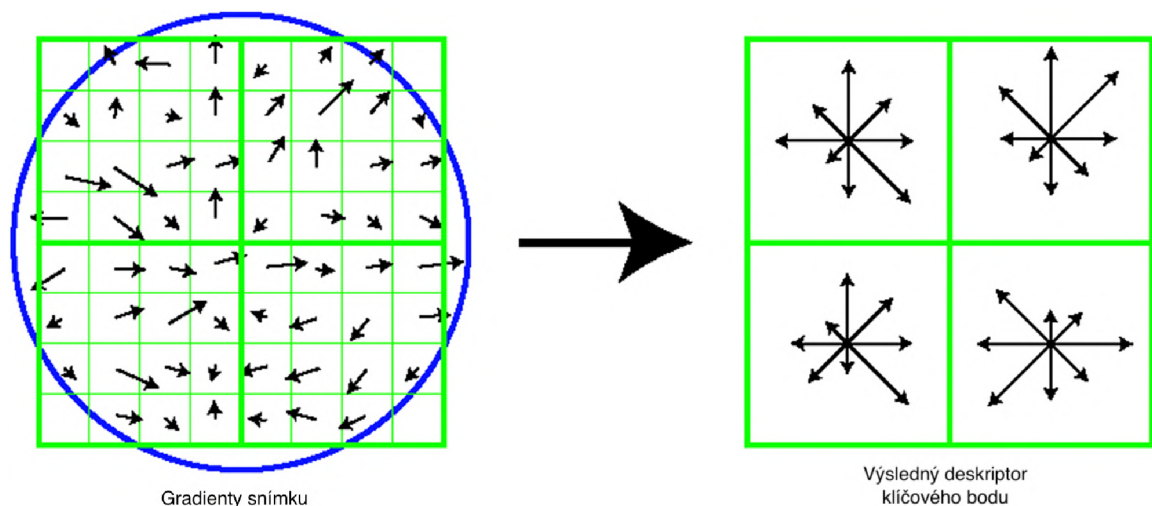


Obrázek 3.11: Na snímku vidíme které body jsou porovnávány v případě hledání extrémů. Pokud box označený X je minimum nebo maximum, označíme jej za klíčový bod.<sup>18</sup>

Je možno použít vzorek intenzit v okolí klíčového bodu v odpovídajícím měřítku, a hledat klíčové body mezi sebou pomocí normalizované korelační míry. Jednoduchá korelace není dostatečná a je obecně dost náchylná k afinní transformaci, změně úhlu pohledu kamery a nerigidní deformaci. Lepší přístup by byl založen na biologickém vidění na neuronech v primární vizuální kůře. Tento přístup publikovali Edelman, Intrator a Poggio v roce 1997. Jak tento přístup funguje podrobněji je zmíněno v [7], pro nás je však důležité, že tento experiment zlepšil rozpoznání 3D objektu otočeného v ose Z o 20 stupňů z 35% s použitím korelace gradientů na úctyhodných 94% při použití tohoto přístupu. D. G. Lowe se touto implementací sice inspiroval, ale použil jiný druh výpočtů.

Samotný popis klíčového bodu je následující. Prvním krokem je určení orientace a míry gradientů okolních bodů klíčového bodu. Orientace všech bodů v okolí je změněna podle orientace samotného klíčového bodu. Všechny gradienty jsou změněny podle úrovně, kde se nacházíme v scale-space. Z tohoto vzniknou gradienty obrazu, což můžeme pozorovat na ilustraci 3.12. Podle D. G. Lowe by bylo vhodné vzít to v potaz a předejít velkým změnám deskriptorů kvůli pozici okna. Další výhodou použití Gaussovy váhové funkce je kladení většího důrazu na gradienty nacházející se blíže středu než na gradienty nacházející se na okrajích.

Deskriptor, který vznikne, můžeme pozorovat na ilustraci 3.12. Jedná se o čtverec o straně  $8 \times 8$  okolo klíčového bodu, který je rozdělen na 4 samostatné čtverce, které se skládají ze  $4 \times 4$  nejmenších čtverců. Z těchto nejmenších čtverců je vypočítán histogram o 8 směrech podobně jako výše. Tím pádem vznikne finální histogram umístěný do  $2 \times 2$  čtverců. Tento histogram obsahuje 32 hodnot (8 hodnot krát 2 krát 2 - velikost čtverců). Toto byl pouze obecný postup. Experimentálně vyšlo jako nejlepší použití 2 krát více nejmenších čtverců. Tudíž 16 nejmenších čtverců, ze kterých vzniknou  $4 \times 4$  čtverce a finální histogram o 8 směrech. Tento používaný histogram obsahuje 128 hodnot (8 hodnot krát 4 krát 4 (velikost čtverců)), tudíž 4 krát více než v původním návrhu).



Obrázek 3.12: Na snímku vidíme v levé části gradienty snímku. Povšimněte si rozdělení  $8 \times 8$ , kdy na výsledný čtverec deskriptoru připadají  $4 \times 4$ , tudíž 16 čtverců. V pravé části snímku máme už výsledný deskriptor, který obsahuje 128 hodnot, které navzájem následně porovnáváme.<sup>20</sup>



### 3.5.5 Výstup SIFT algoritmu a jeho použití pro komerční účely

Výstupem SIFT algoritmu jsou klíčové body, které SIFT algoritmus objevil a odfiltroval. Obsahují polohu klíčového bodu, gradient a klíčový deskriptor, který jsme si popsali jako poslední.

Jelikož tento program vzniká pro komerční použití v rámci projektu Viderize, není použití této metody možné, ovšem problém je poněkud složitější. V rámci konzultace s právníkem o této problematice, která proběhla rovněž v rámci projektu Viderize, nám byla sdělena následující informace. SIFT se definuje

jako soubor matematických operací. Výsledkem těchto operací je nalezení klíčových bodů ve snímku. Ovšem Evropská unie zakazuje patentování jakýchkoliv matematických vzorců nebo postupů. Zbytek světa to umožňuje, a proto by například v Číně nebo v USA tento program není možný volně použít. Ještě složitější situace vznikla z důvodů způsobu použití v knihovně OpenCV. Tato knihovna se prezentuje, jak už z názvu vyplývá, jako open-source. Problémem je to, že v této knihovně je možno použít i tyto patentované algoritmy, které byly při vzniku programu použity. Důvodem je, že tato knihovna je zahrnovala v době, kdy ještě nebyly patentovány, a později tam tyto algoritmy zůstaly. V začátcích práce na tomto programu jsem použil pro hledání klíčových bodů SIFT, přes `xfeatures2d`, a to v domněnání, že se jedná o open-source řešení, jelikož mi tak byla prezentována knihovna OpenCV. Ovšem tato situace mě přivedla k jinému, modernějšímu řešení hledání klíčových bodů, které jsem následně použil i ve finální verzi programu.



Obrázek 3.13: Na snímku vidíme jak jsme našli ve snímku klíčové body metodou SIFT.<sup>21</sup>

<sup>20</sup>Snímek převzat z <https://iopscience.iop.org/book/978-0-7503-1457-2/chapter/bk978-0-7503-1457-2ch13/downloadHRFigure/figure/bk978-0-7503-1457-2ch13f5>

## 3.6 KAZE

Informace a poznatky z podkapitoly byly čerpány z článku autorů [3], [23] a [24].

Při hledání dalších detektorů a deskriptorů klíčových bodů jsem našel jednu z modernějších metod, která vychází ze stejných principů jako metoda SIFT, ale je volně dostupná, ale je jedním přístupem odlišnějším a ve finále možná lepším i detektorem a deskriptorem než SIFT. Pro výpočty je ovšem náročnější, tudíž ji nedoporučuji pro real-time aplikace.

Víceúrovňová měřítka jsou dnes velmi často využívána v počítačovém vidění z důvodu invariantnosti vůči měřítku. Základní myšlenka je jednoduchá. Použijeme pro filtrování například Gaussovu funkci s postupně se zvětšující odchylkou. Čím větší bude odchylka, tím vznikne jednodušší reprezentace snímků, které můžeme porovnávat s jinými snímky na různých úrovních rozmazání nebo měřítka. Samozřejmě jak tomu bývá, vše má svoje klady a zápory. Do kladů lze rozhodně zařadit jednoduchost metody, redukci šumu a také jak dobře tato funkce zachovává důraz na významné lokace snímku. Do záporů se ovšem rozhodně řadí snížení přesnosti lokalizace. Gaussovo rozmazání nerespektuje ostré hranice objektů ve snímku a rozmazává stejným způsobem šum i tyto hranice. Čím vyšší je úroveň rozmazání, tím vyšší je nepřesnost. Tuto nepřesnost se snaží řešit SIFT metoda, ale my ukážeme, že existují jiné způsoby, kterými dokážeme nejen redukovat šum, ale zachovat i ostré hranice objektů na snímku.

Lepším způsobem by bylo vytvořit nějaký způsob, který by se přizpůsobil snímku a vyhladil a redukoval šum, ale hrany a různé detaily by zůstaly nedotčeny. Proto byl prezentován jiný způsob, nyní už nelineární, pro tvoření scale-space navržený v [19] a [4]. Problémem nelineárního rozptylového filtrování je jeho špatná efektivita ve většině přístupů. Ovšem jedno pojetí navržené Weickertem představuje efektivní přístup pro nelineární rozptylové filtrování. Hlavním mechanismem je zde AOS (Additive Operator Splitting) Tento přístup je na rozdíl od ostatních velice efektivní, a proto ho algoritmus KAZE využívá. Dále uvedeme, jak tento algoritmus pracuje.

### 3.6.1 Nelineární rozptylové filtrování

Perona a Malik ukázali nelineární rozptylové filtrování v článku [19]. Klasická nelineární rozptylová funkce vypadá takto:

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, t) \cdot \nabla L),$$

$\text{div}$  a  $\nabla$  jsou odchylka a operátor gradientu,  $c$  je vodivostní funkce v rozptylové rovnici, která nám umožňuje adaptivní rozptyl a  $t$  je měřítko, čím vyšší hodnota tím jednodušší zobrazení.

Perona a Malik navrhli, aby funkce  $c$  závisela na velikosti gradientu, redukovali difúzi v místech hrany a soustředili se na oblasti bez těchto hran, aby zachovali hrany ve snímku a nezasahovali do nich rozmazáváním. Funkce  $c$ :

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|),$$

kde jasná funkce  $L_\sigma$  je gradient Gaussovy rozmazané verze originálního snímku  $L$ . Perona a Malik popsali dvě různé formulace pro vodivostní funkci  $g$ :

---

<sup>21</sup>Snímek převzat z [https://amroamroamro.github.io/mexopencv/opencv\\_contrib/SIFT\\_detector\\_01.png](https://amroamroamro.github.io/mexopencv/opencv_contrib/SIFT_detector_01.png)

$$g_1 = \exp\left(-\frac{|\nabla L_\sigma|^2}{k}\right),$$

$$g_2 = \frac{1}{1 + \frac{|\nabla L_\sigma|^2}{k^2}},$$

kde  $k$  parametr je kontrast, který kontroluje úroveň rozptylu. Rozdíl mezi funkcemi  $g_1$  a  $g_2$  je ten, že funkce  $g_1$  lépe pracuje s rychlými neboli vysoce kontrastními hranami a  $g_2$  lépe pracuje s širšími oblastmi oproti menším oblastem. Existuje i  $g_3$  funkce navržená Weickertem [26], která vyhlazuje okolí hran mnohem znatelněji než hrany samotné, definujeme ji takto:

$$g_3 = \begin{cases} 1 & , |\nabla L_\sigma|^2 = 0 \\ 1 - \exp\left(-\frac{3.315}{(|\nabla L_\sigma|/k)^8}\right) & , |\nabla L_\sigma|^2 > 0 \end{cases}$$

Problémem, který musíme vyřešit, je jak nastavíme konstantu  $k$ . Konstantu je možno nastavit ručně i automaticky. Perona a Malik se rozhodli pro nastavení 70% vyhlazování gradientů v histogramu. Z tohoto vyplývá, že čím je vyšší konstanta  $k$ , tím výraznější hrany zůstanou zachovány.

### 3.6.2 AOS schéma

Problémem parciálních diferenciálních rovnic, které se používají v nelineárním rozptylovém filtrování, je, že není možné tuto funkci řešit analyticky. Existují ovšem numerické metody, které aproximují diferenciální rovnice. Rovnice takzvaně částečně implicitní vypadá takto, za použití vektorové matice:

$$\frac{L^{x+1} - L^i}{\tau} = \sum_{l=1}^m A_l(L^i)L^{i+1},$$

Kde  $A_l$  je matice, která šifruje vodivost snímku pro každou osu. Pro vypočítání  $L^{x+1}$  potřebujeme vyřešit lineární systém rovnic kde  $L^{x+1}$  vypadá takto:

$$L^{i+1} = (I - \tau \sum_{l=1}^m A_l(L^i))^{-1} L^i$$

V tomto případě budeme muset vyřešit matici diagonálně a trigonálně dominantní, kterou jsme schopni vyřešit pomocí známé varianty Gaussovy eliminace, a to Thomasovým algoritmem.

### 3.6.3 Postup algoritmu KAZE

#### Výpočet nelineárního scale space

Hlavním rozdílem při tvoření scale space mezi SIFT a KAZE je to, že algoritmus KAZE na rozdíl od SIFT neupravuje rozlišení snímku, pracuje vždy se stejným rozlišením. Využijeme AOS a vytvoříme set oktáv  $O$  a podúrovně  $S$ .

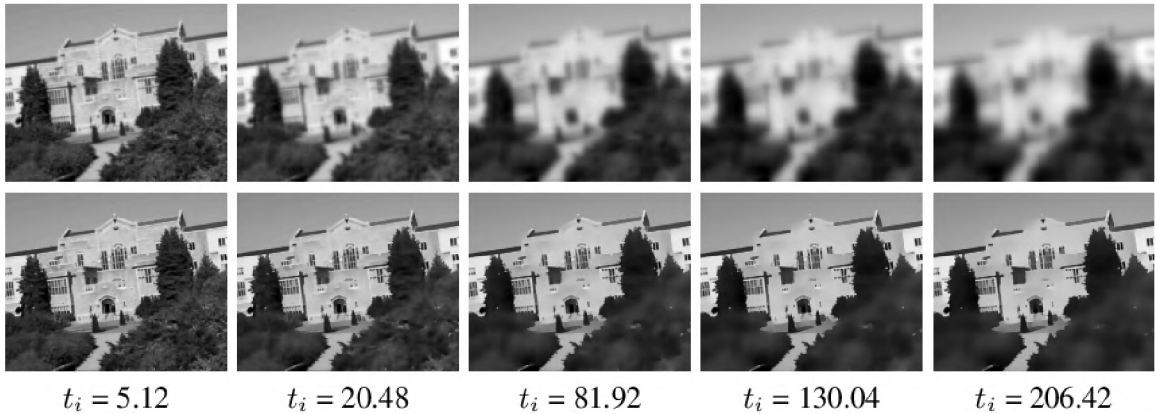
$$\sigma_i(o, s) = \sigma_0 2^{o+s/S}, o \in [0 \dots O - 1], s \in [0 \dots S - 1], i \in [0 \dots N],$$

Kde  $\sigma_0$  je základní úroveň a  $N$  je celkový počet filtrovaných snímků. Dalším krokem je převedení scale úrovní v pixelových jednotkách  $\sigma_i$  do časových jednotek. Důvodem je že,

nelineární rozptylové filtrování je definováno v časových jednotkách. Pro příklad snímek filtrovaný Gaussovou funkcí s odchylkou  $\sigma$  je stejný jako filtrování snímku po dobu  $t = \sigma^2/2$ . Namapování z  $\sigma_i$  na  $t_i$  probíhá následovně:

$$t_i = \frac{1}{2}\sigma_i^2, i = \{0 \dots N\},$$

V následující ilustraci můžeme porovnat, jak probíhá filtrování Gaussovým filtrem a nyní vysvětleným nelineárním odchylovým filtrováním.



Obrázek 3.14: Na snímku vidíme rozdíl mezi Gaussovým a nelineárním rozmazáváním. Můžeme pozorovat jak Gaussové rozmazávání, ve vrchní části, jak nerespektuje přirozené hranice objektů na snímku. V porovnání s nelineárním rozmazáváním vidíme jak hranice objektů, zůstávají zachovány ale skruktury a textury jsou rozmazány. Také vidíme jak postupně zvyšujeme  $t_i$ .<sup>23</sup>

### Detekce klíčových bodů

Pro detekci klíčových bodů zjistíme odezvu měřítkově normalizovaného determinantu Hessianovy matice ve více úrovních. Hledání extrémů probíhá tak, že prohledáváme všechny filtrované snímky kromě prvního a posledního. Důvodem je, že potřebujeme snímek o úroveň níž a snímek o úroveň výš abychom mohli detekovat klíčové body. Nejdříve prohledáme oknem  $3 \times 3$  pixely, abychom rychle vyřadili oblasti bez maxim. Dále snímky prohledáváme čtvercovým oknem o rozměrech  $\sigma_i \times \sigma_i$  ve třech vrstvách. Při nalezení maxima vypočítáme polohu s subpixelovou přesností.

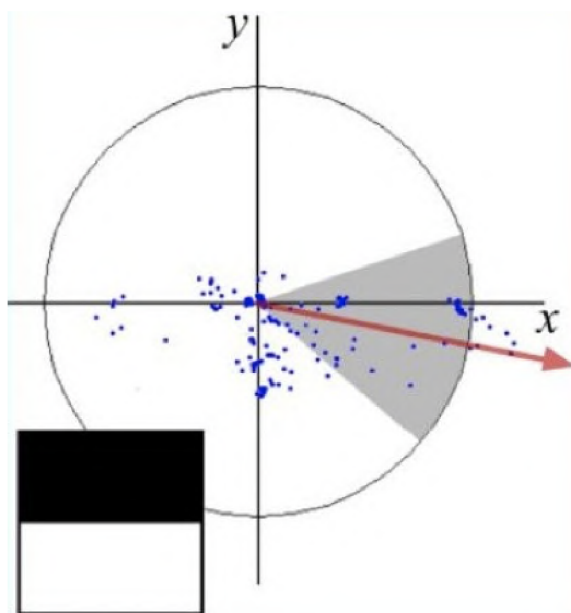
### Nalezení dominantní orientace

Protože podobně jako v metodě SIFT potřebujeme zaručit invariantnost vůči rotacím, musíme zjistit dominantní orientaci nebo směr klíčového bodu. Zde se inspirujeme metodou SURF a dominantní orientaci určují v okruhu  $6 \sigma_i$  se vzorkovacím krokem  $\sigma_i$ . Pro každý vzorek v oběžníkové oblasti je derivace prvního řádu  $L_x$  a  $L_y$  váženě zprůměrována Gaussovou funkcí se středem v místě klíčového bodu. Následně se najde pomocí posuvného kruhu a sečtením hodnot nacházejících se v kruhu dominantní směr. Vektor, který je nejdelší, je označen jako dominantní.

<sup>23</sup>Snímek byl převzat z <https://ai2-s2-public.s3.amazonaws.com/figures/2017-08-08/dd89b0332ae4f980d8200ec2c1230c50cd66259f/7-Figure2-1.png>

## Popis klíčového bodu

Algoritmus KAZE využívá M-SURF deskriptoru upraveného pro použití při nelineárním filtrování. Pro detekovaný klíčový bod v měřítku  $\sigma_i$  je spočtena první derivace  $L_x$  a  $L_y$  rozměru  $\sigma_i$  je v oblasti  $24\sigma_i \times 24\sigma_i$  mřížce. Mřížka je rozdělena na 4 suboblasti, tedy  $2 \times 2$ , ovšem zajímavostí je, že se velikosti těchto mřížek překrývají. Velikost jedné mřížky je  $9 \sigma_i \times 9 \sigma_i$  tudíž  $2 \sigma_i$  je oblast, která se překrývá. Provedeme derivaci v každé suboblasti váženou Gaussovou funkcí s odchylkou  $2,5\sigma_i$ . Poté se každý vektor suboblastí opět váhově zprůměruje Gaussovou funkcí s odchylkou  $1,5 \sigma_i$  definovanou přes masku  $4 \times 4$  a se středem v klíčovém bodě. Každý vektor se dále otočí ve směru dominantní orientace, kterou jsme si určili výše. Posledním krokem je, že výsledný vektor (o délce 64) a jednotkový vektor se normalizují pro dosažení invariance vůči kontrastu.



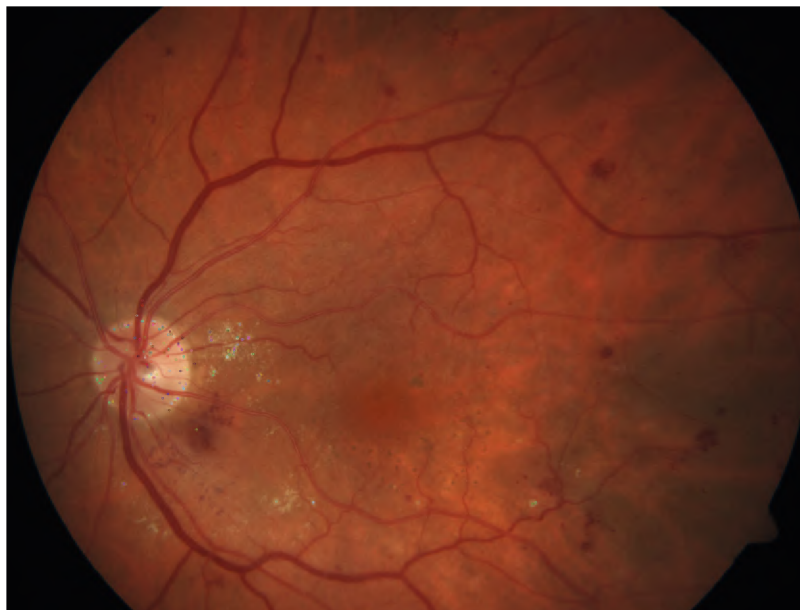
Obrázek 3.15: Na snímku vidíme znázornění popisu klíčového bodu. Červená šipka znázorňuje orientaci. Klíčový bod je popsán středem kruhu. Čtverec zobrazuje derivaci v ose y.  
<sup>25</sup>

## Použití

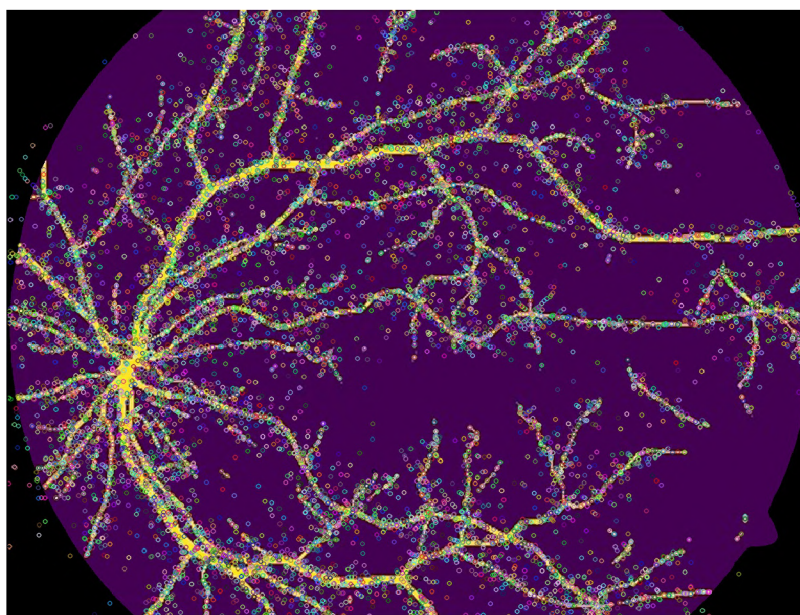
Algoritmus KAZE má jistou nevýhodu, a tou je rychlost a výpočetní náročnost. V porovnání s jinými algoritmy na hledání klíčových bodů patří z důvodů výpočetní náročnosti k těm nejpomalejším. Existuje varianta algoritmu, která je akcelerovaná a jmenuje se AKAZE. Mezi další algoritmy patří například ORB, BRISK, FREAK a jiné. Po finální rozhodnutí jsem ovšem zvolil KAZE, a to z toho z důvodů, že jsem vyzkoušel všechny dostupné algoritmy a KAZE se ukázal jako jeden z nejlepších a také jeden z nejnovějších. Tuto metodu výběru doporučuji, protože v našem případě byla tou nejlepší metodou na vybrání algoritmu přistupovat k výběru empiricky. Mnohé z algoritmů se zdají kvůli něčemu výhodné, ale ve finále je vždy potřeba experimentovat.

<sup>25</sup>Snímek i popis byl převzat z <http://www.diva-portal.org/smash/get/diva2:927480/FULLTEXT01.pdf>

<sup>27</sup>Snímek je z databáze v rámci projektu Viderize.



Obrázek 3.16: Na snímku vidíme snímek sítnice, na kterém se nachází pár klíčových bodů, které pro složení dvou snímku zcela určitě nestačí. Na dalším snímku si ukážeme kolik nalezneme klíčových bodů v případě extrakce žil.<sup>27</sup>

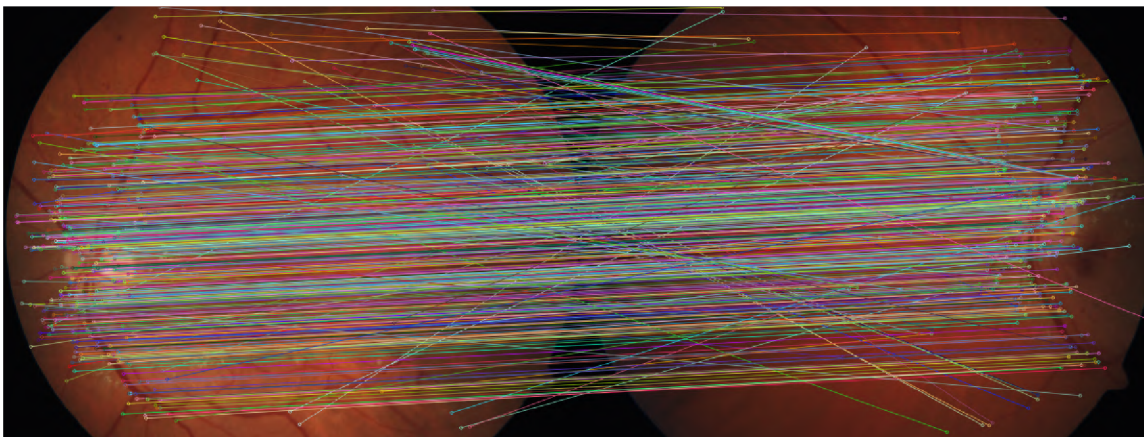


Obrázek 3.17: Na snímku můžeme vidět už mnohem více klíčových bodů kolem charakterických cév na sítnice. Tento počet byl docelený použím KAZE algoritmu na extrahované cévy ze snímku. Pro skládání použijeme tyto snímky.<sup>29</sup>

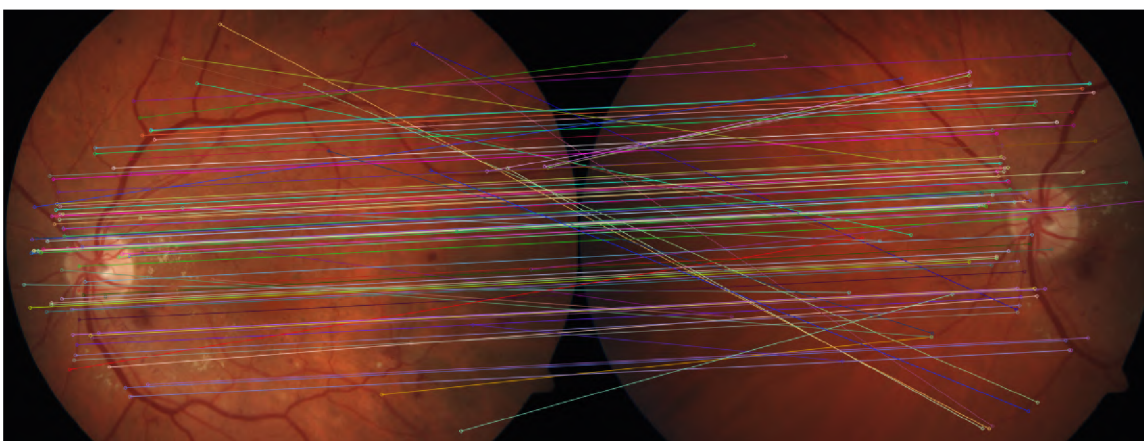
---

<sup>29</sup>Program pro extrakci žil mi byl poskytnut v rámci projektu Viderice panem Ing. Tomášem Vičarem. Ještě jednou mu tímto chci poděkovat za možnost použití programu. Snímek je z databáze v rámci projektu Viderize.

<sup>31</sup>Snímky je z databáze v rámci projektu Viderize.



Obrázek 3.18: Na snímků můžeme všechny nalezené společné klíčové body metodu KAZE. Shod je přesně 521.<sup>31</sup>



Obrázek 3.19: Na snímků můžeme upravenou verzi pouhých 50 shodných klíčových bodů. Můžeme pozorovat, že většina je opravdu stejná oblast ale objeví se například i špatné spárování bodů, které nevadí jelikož v poměru ku správných jsou špatně zanedbatelné.<sup>33</sup>

### 3.7 Homografie

Informace čerpány v rámci předmětu IZG z [28]. Homografie je namapování bodů z prvního snímku, v našem případě klíčových bodů, na body nacházející se na druhém snímku pomocí transformační matice. Pokud chceme vypočítat homografii mezi dvěma snímky, máme dvě možnosti, jaké to budou snímky. První možností je, že snímky jsou stejné, ovšem úhel, pod kterým se zaznamenaly, je jiný. Druhou možností je, že jsou snímky změněny, ale úhel zaznamenávání zůstal stejný. Toto nemusí platit pro celek, ale pouze na některou jejich společnou část. K využití dochází při skládání dvou snímků podle společné oblasti, kterou oba snímky obsahují. Po výpočtu homografie vznikne transformační matice, tuto aplikujeme na jeden snímek a přetransformujeme jej, aby do sebe společné oblasti zapadaly (klíčové

<sup>33</sup>Snímky je z databáze v rámci projektu Viderize.

body). Výstupem tohoto procesu je vytvoření nového snímku, který vzniknul spojením dvou původních, pokud obsahovaly společnou oblast.

Transformační matice je o obsahu 9 prvků ve  $3 \times 3$  matici. Každý prvek obsahuje jinou část transformace a použití matice dovoluje transformovat snímek aplikováním matice na každý bod snímku. Podoby matic jsou:

- Rotace

$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- změna měřítka

$$\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Translace

$$\begin{pmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{pmatrix}$$

- Zkosení

$$\begin{pmatrix} 1 & Z_x & 0 \\ Z_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

### 3.8 RANSAC

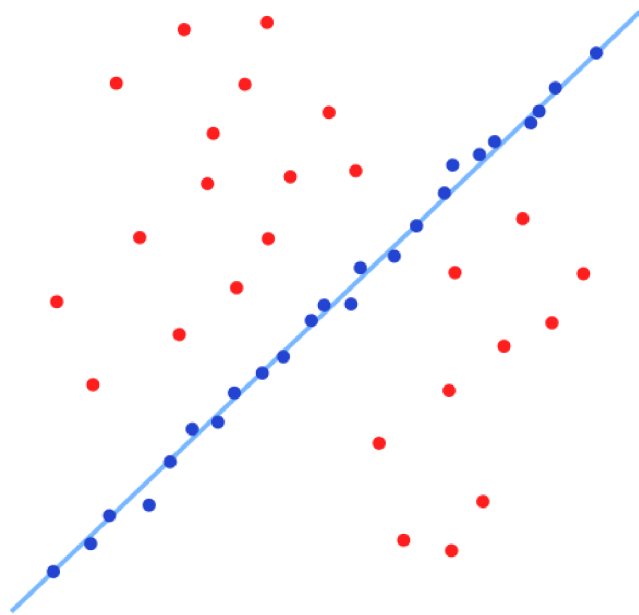
Informace čerpány z [22]. Pro výpočet homografie se často využívá metoda RANSAC neboli Random Sample Consensus. [23] Tato metoda nesouvisí pouze s počítačovým viděním, ale například také s hledáním vzorů ve statistice a podobně.

Jedná se o iterativní metodu hledání vzorů. Tudíž bere tato metoda ze snímků pouze klíčové body a snaží se najít nasazení jednoho modelu na druhý. Čím jednodušší model je vyhledáván, tím lépe se vyhledávají přímky, kružnice a jiné jednoduché tvary. Při proložení dvou modelů dojde na rozdělení bodů, v našem případě klíčových bodů, na dva druhy. Jeden druh je inliers a druhý je outliers. Outliers jsou poměrně nezájímavé, jak už z názvu vyplývá, jedná se o body, které v modelu sice jsou, ale nepodařilo se mezi nimi najít spojení. Inliers jsou ty důležitější body, mezi kterými se podařilo najít například přímku. Toto náhodné skládání modelů probíhá, dokud nejsou změny zanedbatelné nebo se nedosáhne konečného počtu iterací.

---

<sup>35</sup>Snímek převzat z [https://upload.wikimedia.org/wikipedia/commons/thumb/d/de/Fitted\\_line.svg/510px-Fitted\\_line.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/d/de/Fitted_line.svg/510px-Fitted_line.svg.png)





Obrázek 3.20: Na snímku můžeme vidět náhodně rozmístěné body, ve který jsou modré inliers a červené outlier. V našem případě, jsme našli přímku. <sup>35</sup>

## Kapitola 4

# Návrh implementace

První nápad, který přijde na mysl, když máme za úkol složit více snímků dohromady, je, že budeme skládat jednotlivé snímky tak, že složíme první dva, vznikne nám nový snímek, k tom přidáme další snímek a toto opakujeme, dokud máme snímky, které můžeme přidávat. Nyní si zkusíme navrhnout jednotlivé kroky algoritmu, který bychom chtěli poté implementovat.

Jediné, co dostaneme, je sada snímků, které nejsou nijak upravené, s černým pozadím, většinou ve formátu JPG. Dále je problém v pozadí snímků. V případě, že bychom přes sebe skládali snímky, které mají černé pozadí, tak to nebude fungovat. Černá část snímků bude překrývat snímek sítnice, což samozřejmě nechceme. Dále musíme vymyslet způsob, jak složit správně dva snímky, které obsahují částečně stejnou plochu.

### 4.1 Snímky a jejich formát

Na začátku dostaneme pouze sadu snímků, což jsou snímky jednoho oka ve formátu JPG. Prvním úkolem je tedy převést snímky z formátu JPG do formátu PNG. O upřesnění tohoto formátu a důvodu, proč jsem se rozhodl právě pro něj, bude podrobněji rozebráno v teoretické části práce. Nejdůležitější je zachování kvality fotografií.

Dalším krokem je zbavení se černého pozadí, které obsahuje každý snímek. Musíme oříznout, případně jinak odstranit černé okraje z důvodu, abychom mohli složit dva snímky a aby jejich návaznost nebyla narušena černým pozadím.

### 4.2 Skládání snímků

Když všechny snímky převedeme do vhodného formátu a podoby, budeme muset najít nějaký způsob složení dvou snímků. Prvotně mě napadlo najít snímek se slepou skvrnou a poté ji využívat jako

důležitý bod, který lze využívat pro skládání snímků, jelikož každá sada obsahuje snímek, kdy se pacient dívá přímo do objektivu, to znamená, že slepá skvrna je přímo uprostřed snímku. Toto ovšem má omezené využití, protože zornice se nenachází na všech snímcích, tudíž by byl problém u skládání dalších snímků, které zornici neobsahují. Navíc některé poškození nebo nemoci oka mohou vypadat podobně jako slepá skvrna, což by mohl být také problém.

Dalším objektem v oku, který by se mohl využít pro složení dvou obrázků, jsou cévy. Ty se nacházejí v každém oku, ale mohou být ovlivněny nějakým poškozením nebo One-

mocněním. Ovšem cév v oku se nachází více a je nepravděpodobné, že by byly poškozené všechny cévy. Tudíž bude pro nás klíčové zkusit segmentovat cévy, dále pracovat s nimi a nacházet stejné plochy podle nich. Dále budeme muset najít nějaký algoritmus, který nám bude schopen najít podobné tvary cév. Jelikož máme více snímků, pro první skládání bude nejlepší zkusit složit každý snímek s každým (samozřejmě vyjma sebe sama) a podle nějaké kontroly zjistit, které složení bude nejlepší, a tento snímek uložit, jelikož s ním budeme dál pracovat.

### 4.3 Cyklus skládání snímků

Dostaneme více než dva snímky, což pro nás znamená, že spojení dvou snímků není konečný výsledek. Nově vzniklý snímek použijeme jako jeden ze dvou snímků, které budeme opět skládat a další snímek vybereme opět stejnou metodou, jako v předešlém kroku. Dále je zde například problém chyb, které mohou vzniknout při skládání snímků, nebo otázka, jestli nově přidaný snímek není zbytečný. Jelikož jsou snímky focené manuálně, může se stát, že se vyfotí snímek téměř stejný, jako už byl vyfocen, a jeho přidáním do finálního snímku není jisté, že by se výsledek vylepšil.

Chyby vzniklé při skládání budeme muset nějak eliminovat. Jeden špatně složený snímek by mohl vyústit v celkově špatný snímek, i když se například složily 4 snímky správně. Je samozřejmě lepší mít složené 4 snímky správně, než mít složeno všech 6 snímků s nějakou chybou.

Jedná se o návrh řešení, tudíž nemám prozkoumané všechny možnosti, některé věci se mohou změnit, je to pouze prototyp.

## Kapitola 5

# Implementace

Pro implementaci jsem použil programovací jazyk Python, z důvodů práce v rámci projektu Viderize, byl v týmu dohodnut Python. Dalším důvodem je použití knihovny OpenCV, která se používá především v jazyku C++ a Python, tudíž bylo použití Pythonu jasná volba. Dalšími důležitými knihovnami je NumPy a PIL.

Implementace je rozdělena na 2 části. První část je před samotným skládáním snímků příprava vstupních dat a druhá část samotné skádání snímků.

### 5.1 Knihovna OpenCV

Jedná se o open-source knihovnu pro počítačové vidění. Je psaná v jazyku C a C++, ale je možné ji používat i v jazycích Python, Matlab a další.

Knihovna se soustřeďuje na efektivnost z důvodu soustředění na real-time aplikace, ovšem nevyklučuje samozřejmě ani jiné použití. Cílem této knihovny je poskytnout co nejvíce možností využití pro jakékoliv aplikace, které potřebují počítačové vidění. A vše toto poskytnout v jednoduché formě, v jednoduché podobě a vytvořit i složitější aplikace pomocí této open-source knihovny.

### 5.2 Knihovna NumPy

Numpy je knihovna, která se používá především pro matice a vícerozměrná pole. Byla navržena především pro vědeckou sféru, ale v naší práci najde využití pro práci se snímky, jelikož se snímky pracujeme jako s vícerozměrnými poli. Licence je BSD.

### 5.3 Pillow

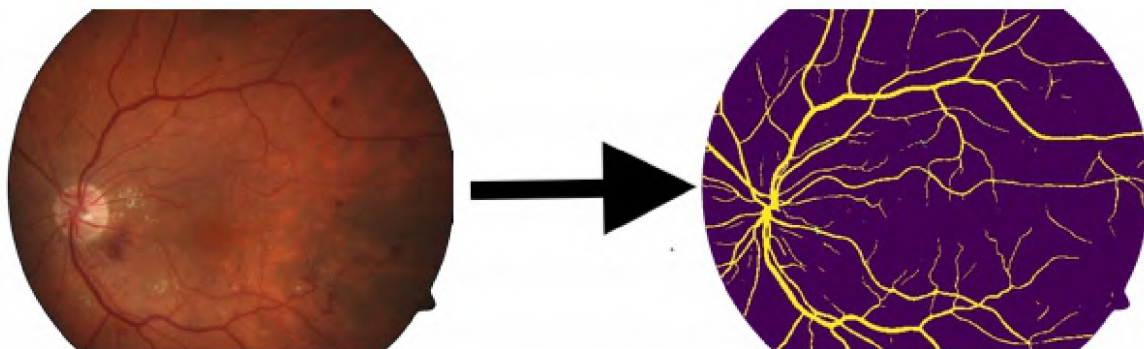
Je knihovna, která se využívá pro otvírání, ukládání a manipulaci. V této práci se využívá především na ukládání snímků. Je pod licencí Python Imaging Library license. Tato licence dovoluje za jistých podmínek využití této knihovny za jakýmkoliv účelem bez poplatku.

### 5.4 Příprava dat

Příprava vstupních dat je ještě rozdělena na dvě části. Úplně první částí této práce je extrakce cév ze snímků sítnice. Tento krok byl přidán v pozdější fázi práce, jelikož v první

verzi, kde se využíval algoritmus SIFT, nebyla potřeba extrakce cév, ovšem jelikož algoritmus SIFT nebylo možné použít z důvodů problému pro komerční použití, tak byl zvolen algoritmus KAZE. Tento algoritmus nebyl dostatečný v hledání klíčových bodů, ale v rámci projektu Viderize jsem pracoval s programem pro extrakci žil od pana Ing. Tomáše Vičara, za který mu děkuji. Jedná se o U-Net síť, která byla trénovaná na volně dostupných datech. Tato metoda se využívá v mnoha oblastech biomedicínských snímků pro extrakce. <sup>1</sup>

Jakmile získáme sadu vyextrahovaných snímků, můžeme pokračovat v přípravě dat. Budeme pracovat s dvěma sadami snímků, jedna sada je s vyextrahovanými cévami, druhá sada jsou snímky bez úprav.



Obrázek 5.1: Na snímku můžeme vidět extrakci výsledek extrakce cév ze snímku sítnice. <sup>2</sup>

Důležitým krokem přípravy je převedení všech snímků do stejného formátu. Formát PNG je pro nás nejvíce vyhovující, tudíž převedeme oboje sady snímků do formátu PNG. Další částí přípravy dat je oříznutí černého okolí u snímků. Jelikož se ve mně dostupných datech objevilo více druhů snímků, musel jsem použít jednodušší metodu než je alpha maska. Zvolil jsem iteraci přes celý snímek a nastavení alpha kanálů u každého pixelu, který se dá považovat za černý. Tato metoda není nijak rychlá, ale je zase jednoduchá, a jelikož jsem chtěl dosáhnout univerzálnosti a optimalizace nebyla důležitá, zvolil jsem tuto metodu. Sadu s extrahovanými cévami je potřeba aplikovat masku `mask.png` s černým pozadím, jelikož při extrakci žil vznikne jednotné pozadí a my potřebujeme vytvořit tvarově stejné snímky v obou sadách.

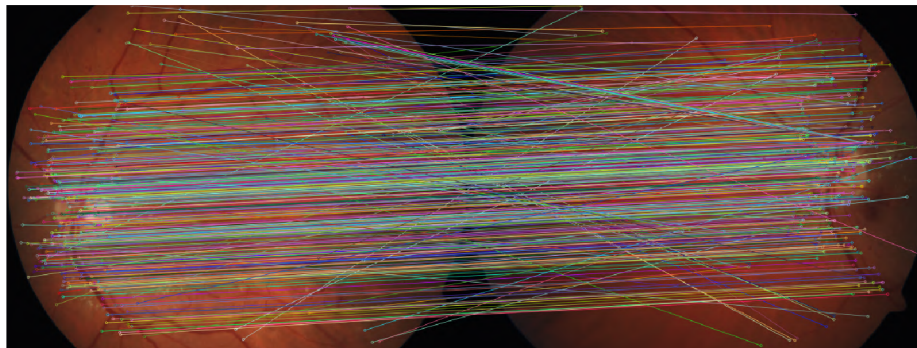
## 5.5 Skládání snímků

Když si připravíme obě sady bez černých okrajů, můžeme zahájit proces skládání. Prvním krokem je projití a nalezení vhodných snímků pro skládání. Rozhodl jsem se pro skládání jednotlivých snímků a následně skládání těchto dvojic snímků, kdy každá dvojice bude unikátní, to znamená, že nepoužiji jeden snímek vícekrát. Vše provádím po převodu snímků do černobílé podoby. Pokud má dvojice snímků určitý počet společných klíčových bodů, které naleznou metodou `cv2.BFMatcher`, uložím si, které snímky to jsou, a počet jejich společných klíčových bodů, které následně seřadím podle počtu klíčových bodů. Metoda `cv2.BFMatcher` naleznou dvojice stejných klíčových bodů ze dvou snímků hrubou silou, to znamená vyzkoušením každého bodu s každým. Čím více je společných bodů, tím je ve vět-

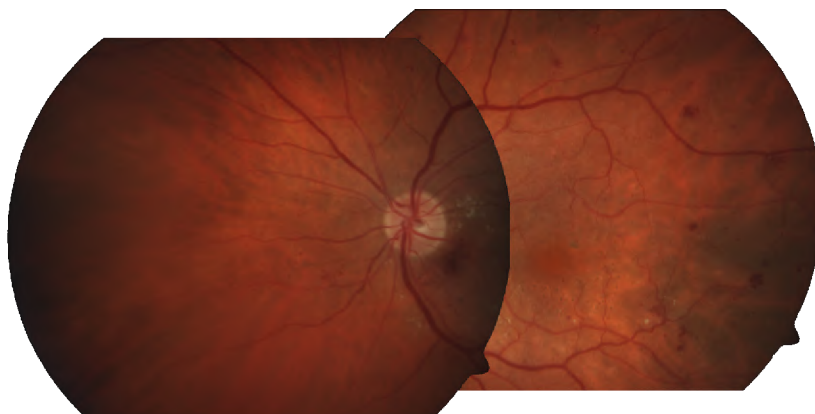
<sup>1</sup>Více informací o U-Net síti naleznete zde <https://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a/>

<sup>2</sup>Snímky je z databáze v rámci projektu Viderize.

šíně případů lepší složení snímků. Klíčové body hledám a popisuji algoritmem KAZE. Následně tyto body najdu vzájemně hrubou silou a také použiji metodu k-nejbližších sousedů pro získání ještě lepšího vzorku společných bodů. Zvolené hodnoty vzdáleností sousedních bodů a všechny proměnné jsou převzaty z [12], kterou jsem používal pro tvoření programu.



Obrázek 5.2: Na snímků můžeme všechny nalezené společné klíčové body metodu KAZE. Shod je přesně 521.<sup>4</sup>



Obrázek 5.3: Na snímku můžeme vidět všechny nalezené společné klíčové body metodou KAZE.<sup>6</sup>

V kódu programu se nacházejí funkce, které měly za úkol znázorňovat klíčové body a společné klíčové body ve snímcích. Využíval jsem je pro tvoření ilustrací do této práce, ale rozhodl jsem se je tam ponechat z důvodu užitečnosti pro budoucí použití.

Dalším krokem je vytvoření zmíněných dvojic, které budou skládány následovně. Nejdříve potřebujeme strukturu `dictionaryOfMatches` seřadit podle největšího počtu klíčových

<sup>4</sup>Snímky je z databáze v rámci projektu Viderize.

<sup>6</sup>Snímky je z databáze v rámci projektu Viderize.

bodů a následně udělat unikátní dvojice snímků. Při skládání snímků jsem ovšem narazil na menší problém, který se občas objevoval. Jednalo se o pořadí snímků při skládání. Pro příklad snímek x a snímek y a naopak snímek y a snímek x nevytvářely stejné složení snímků. Občas se stalo, že některé snímky se sice složily, ale výsledný snímek se skládal prakticky pouze z jednoho snímku, nebo se stávalo, že při menších počtech společných klíčových bodů byla vytvořena homeografie, které způsobila, že snímek se roztáhl do několikanásobku a výsledek vůbec nebyl správný. Z tohoto důvodu vytvářím složení ve dvou různých pořadích a následně vyberu správnou variantu, kterou jsem se rozhodl určit z velikosti snímků. Pokud má složený snímek příliš malou velikost, tím je myšleno, že je velice velikostně podobný původnímu jednotnému snímku, nebo naopak je jeho velikost několikanásobně větší než původní snímek, znamená to, že došlo k velkému roztažení. Správně složený snímek by měl mít alespoň 1,5násobnou velikost původního snímku, ovšem musí být menší než 2násobek původního snímku. A přesně takové snímky ukládám do další struktury pro další použití. Důležité je připomenout, že skládám snímky jak z extrahované sady, tak

i z neupravené sady. Všechny výpočty se provádí nad extrahovanými snímky, na neupravené snímky se pouze aplikuje aktuálně vypočtená matice, žádné úpravy ani hledání klíčových bodů se nad sadou neupravených snímků neprovádí.

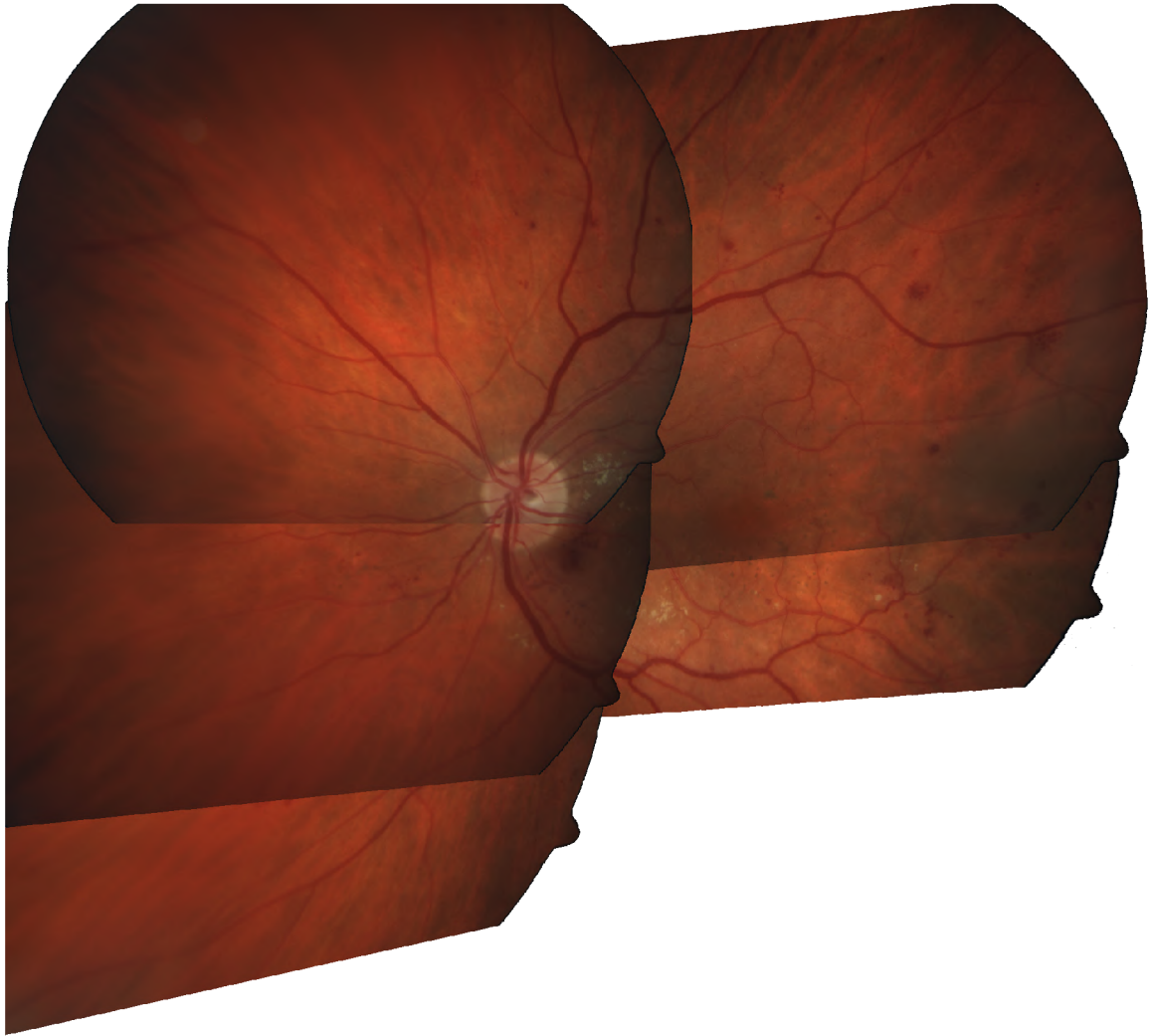
Když poskládám všechny dvojice a uložím si je do seznamu, začnu proces skládání znovu, ovšem nově s těmito snímky dvojic. Jakmile složím dvě dvojice snímků, které mají dostatečný počet klíčových bodů, opět je uložím na začátek stejného seznamu a pokračuji, dokud mi nezůstane pouze jediný snímek, který následně uložím jako výsledek, a ukončím běh programu.

---

<sup>8</sup>Snímky je z databáze v rámci projektu Viderize.

<sup>10</sup>Snímky je z databáze v rámci projektu Viderize.

<sup>12</sup>Snímky je z databáze v rámci projektu Viderize.



Obrázek 5.4: Na snímku můžeme výsledek skládání snímků sítnice, kterého jsme dosáhli použitím výše zmíněné implementace. Můžeme pozorovat celkem rozsáhlou výslednou fotku, kdy je pokryta skoro celá sítnice.<sup>8</sup>





Obrázek 5.5: Na snímku můžeme vidět další výsledek skládání snímků. Výsledek je celkem obsáhlý, ovšem uprostřed vidíme černý flek. Důvodem fleku je, že jelikož nijak neupravují fotky, tak některé fotky jsou vyfocené nekvalitně a tudíž tam mohou vznikat takové vady.<sup>10</sup>



Obrázek 5.6: Na snímku můžeme vidět další výsledek skládání snímků. Výsledek jsou pouze dva snímky, což se občas může stát, z důvodu dbání aby výsledek neobsahoval duplicitní snímky. I tak není snímek k zahoezení. Poskytne nám zhruba 40% celého pozadí sítnice. <sup>12</sup>

## Kapitola 6

# Sumarizace a konzultace výsledků s očním lékařem

Pro vyhodnocení výsledků skládání snímků jsem konzultoval výsledky s MUDr. Tomášem Mňukem, za což mu opět děkuji.

Prvním nedostatkem je způsob, jímž byly pořízeny snímky, se kterými jsem pracoval. Zdály se mu divné a některé málo kvalitní. Prvním problémem je jejich tvar, považoval za nestandardní, že nebyly kulaté, jak je snímání například u Fakultní nemocnice u sv. Anny v Brně.



Obrázek 6.1: Způsob snímání například u Fakultní nemocnice u sv. Anny v Brně.<sup>1</sup>

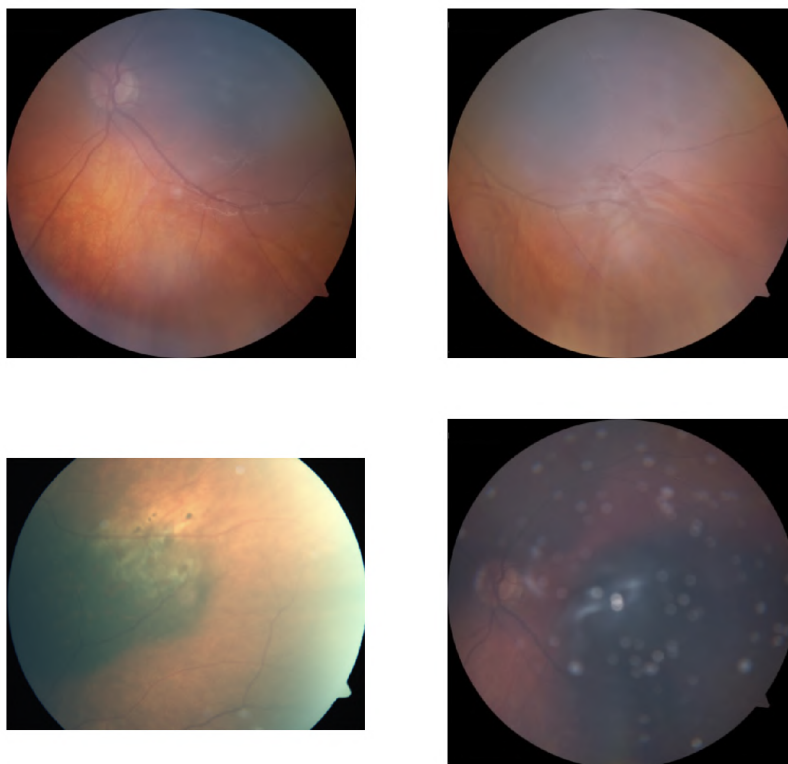
---

<sup>1</sup>Snímek je z databáze v rámci projektu Viderize.



Obrázek 6.2: Způsob snímání snímků, které jsem skládal v této práci.<sup>2</sup>

Dále jejich kvalita byla o poznání nižší než kvalita, s níž fotí Fakultní nemocnice u sv. Anny. Problém představovaly také některé nedostatky, které vznikaly při focení 6.3.



Obrázek 6.3: Na snímcích můžeme pozorovat nejčastější chyby, které vznikají při snímání sítnice. Patří mezi ně špatné nastavení světlosti, příliš tmavé nebo příliš světlé. Dále zde můžeme pozorovat prachové částice nebo jiné nečistoty. <sup>4</sup>

Kvalita samotných výsledků je podle MUDr. Mňuka pro komerční použití nedostatečná. Hlavní nedostatky jsou občasná špatná spojení snímků, nejsou zkrátka ideální a to je pro lékaře problém. Dalším problémem jsou zřetelné změny na přechodech mezi určitými snímky. Tyto dva problémy se dají vyřešit více způsoby, ale předtím si řekneme, jak docílíme tak dobrých snímků jako například z kamery ZEISS VISUCAM 224/524, která se používá pro snímání sítnice a představuje podle MUDr. Mňuka jednu z aktuálních špiček v oboru.

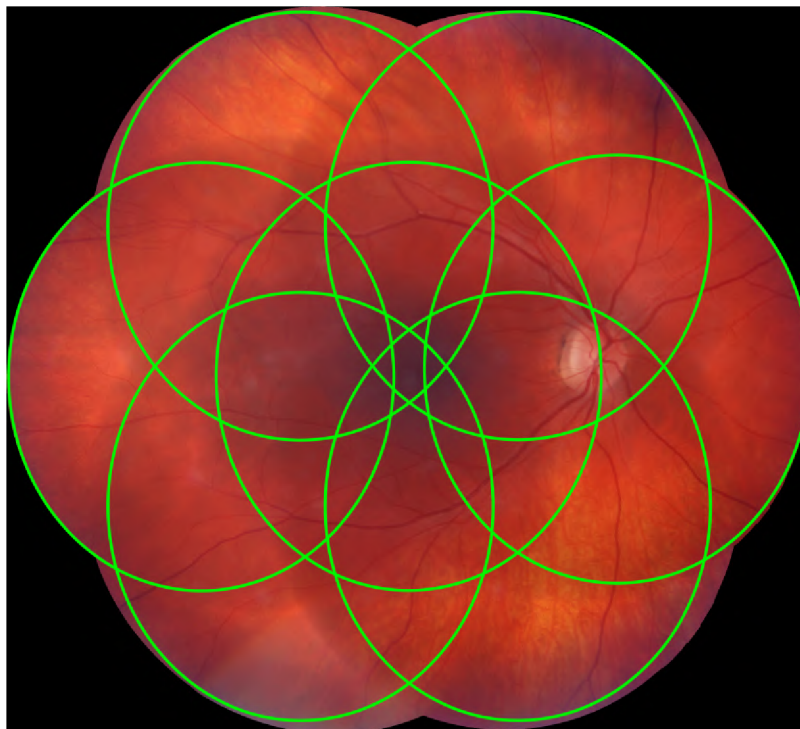
S kamerou ZEISS VISUCAM 224/524 se už od začátku pracuje jinak. Pokud lékař snímá celou sítnici a bude chtít složit snímky do jednoho, což je úkolem naší práce, kamera mu poskytuje instrukce, které části oka snímat. Dále je zde zelené světlo, které svítí pacientovi do oka. Toto světlo mu určuje, kam se má dívat, tudíž kamera může sestavit pravděpodobné rozložení snímků už při jejich pořizování. Mému programu nezáleží na pořadí či na způsobu snímání, pouze zpracovává vstupy a vytváří co nejlepší možný výsledek. Snímky, které byly v datech, s kterými jsem pracoval, nejsou vždy standardní, jak mi sdělil MUDr. Mňuk. Dle slov lékaře se snímá 7 snímků přesně určených oblastí 6.4.

Srovnávat výsledky z kamery, která je jednou ze špiček v oboru, s tímto programem se musí s určitou rezervou. Musíme vzít v potaz, že program, který obsahuje kamera ZEISS

<sup>2</sup>Snímek je z databáze v rámci projektu Viderize.

<sup>4</sup>Snímky je z databáze v rámci projektu Viderize.

<sup>6</sup>Snímky je z databáze v rámci projektu Viderize.



Obrázek 6.4: Zelené kruhy znázorňují jednotlivé snímky. Můžeme vidět mnohem lepší přechody i přesnost návaznosti například cév. Tento snímek vznikl kamerou ZEISS VISUCAM 224/524.<sup>6</sup>

VISUCAM 224/524 je navržený přímo pro tuto konkrétní kameru, tudíž je mnohem jednodušší pro program kamery složit snímky. Dále je důležité říci, že kamera říká doktorovi, které oblasti má oko má snímat. Tímto způsobem je program schopen určit přibližnou polohu snímků. Dále je důležité vzít v potaz, že kamera je schopna určit, o kolik se posune střed dalšího snímku, tudíž jeho polohu celkem přesně. Zelené světlo, které svítí pacientovi do oka, mu určuje, kam se má dívat, je velice užitečným nástrojem a podle mě je to jeden z klíčových detailů, jak dosáhnout lepšího skládání snímků. Kamera ví, o kolik posunula zelené světlo, a jelikož pacient má oko co nejbližší objektivu, můžeme celkem přesně určit posunutí snímků vůči původním.

Mezi jasné body, které se dají zlepšit v této práci, jsou přechody mezi snímky, o způsobech se píše zde [20]. O výsledcích mé práce MUDr. Mňuk řekl, že to určitě není dostatečně dobré, ovšem není to hrozné a základ je použitelný. V této práci mě napadá řada vylepšení. Například přechody snímků, lepší detektor klíčových bodů, lepší výpočet transformačních matic nebo také využití zeleného světla pro lepší vyfocení konkrétního místa sítnice. Nejlepší výsledky vznikají dle mého názoru skloubením hardwaru a softwaru. Je nutné vytvořit program, který bude využívat konkrétní hardwar, tudíž se maximálně využijí obě sféry pro co nejlepší výsledek.

# Kapitola 7

## Závěr

Cílem této práce bylo prostudování literatury pro skládání snímků, v našem případě sítnice, navržení algoritmu a jeho implementace, dále implementace navrženého algoritmu a sumarizace a vyhodnocení výsledků s očním lékařem. Výsledná práce se tyto body snaží splňovat.

Teoretická část práce představí všechny teoretické oblasti, které by byly potřebné pro základní skládání snímků, a také poskytuje obstojný výčet literatury, z níž jsou tyto znalosti čerpány. V mnoha případech jde o tituly přímo od autorů těchto algoritmů. Návrh implementace a samotná implementace jsou velmi podobné, až na případné další problémy, které byly nalezeny, jsou řádně okomentované v kódu, tudíž je vše řádně popsáno a zdokumentované a zájemci o prostudování či inspiraci naleznou srozumitelnou práci a celý postup.

Výsledky samotné, jak mi bylo sděleno MUDr. Mňukem, jsou dobré pouze jako základ, nikoliv pro komerční nasazení. Sám jsem doufal v trochu lepší výsledek, ovšem také jsem si vyhledal, jak by se lepšího výsledku dalo dosáhnout. Míchání a lepší přechod mezi snímky by určitě zlepšily dojem z celého výsledku, ale také by se musel vylepšit algoritmus hledání klíčových bodů. Musíme počítat s tím, že algoritmus SIFT, který je jedním z nejlepších, zde nebylo možné využít z důvodů patentu. Práce vzniká v rámci projektu Viderize a je požadováno volně dostupné řešení. Pro další vylepšení výsledků by se už muselo pracovat přímo s konkrétní kamerou, a jak bylo zmíněno v 6, skloubit software a hardware, kdy bude vznikat program přímo pro danou kameru. Budou se zde moct přidat různé způsoby pro vylepšení skládání snímků, například určení, kam se má pacient dívat, aby vznikl snímek konkrétní oblasti sítnice.

# Literatura

- [1] ADEL, E., ELMOGY, M. a ELBAKRY, H. Image Stitching based on Feature Extraction Techniques: A Survey. *International Journal of Computer Applications (0975-8887)*. 2014, sv. 99, č. 6. Dostupné z: [https://www.academia.edu/17493341/Image\\_Stitching\\_based\\_on\\_Feature\\_Extraction\\_Techniques\\_A\\_Survey](https://www.academia.edu/17493341/Image_Stitching_based_on_Feature_Extraction_Techniques_A_Survey).
- [2] AKRAM, M. Seamless Fundus image stitching using WLD to improve field of view. In: *2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. IEEE, s. 106–110.
- [3] ALCANTARILLA, P. F., BARTOLI, A. a DAVISON, A. J. KAZE Features. *Computer Vision – ECCV 2012*. 2012. DOI: 10.1007/978-3-642-33783-3\_16.
- [4] ALVAREZ, L., LIONS, P.-L. a MOREL, J.-M. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. II. *SIAM Journal on Numerical Analysis*. 1992, sv. 29, č. 3. DOI: 10.1137/0729052.
- [5] BIDGOOD, W. D., HORII, S. C., PRIOR, F. W. a SYCKLE, D. E. V. Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association*. Oxford University Press (OUP). květen 1997, sv. 4, č. 3, s. 199–212. DOI: 10.1136/jamia.1997.0040199. Dostupné z: <https://doi.org/10.1136/jamia.1997.0040199>.
- [6] BRADSKI, G. a KAEHLER, A. *Learning OpenCV : computer vision with the OpenCV library*. Sebastopol, CA: O’Reilly, 2008. ISBN 978-0-596-51613-0.
- [7] EDELMAN, S., INTRATOR, N. a POGGIO, T. *Complex Cells and Object Recognition*. Červen 1997.
- [8] FOLEY, J. *Computer graphics : principles and practice*. Reading, Mass: Addison-Wesley, 1995. ISBN 0-201-84840-6.
- [9] FORUTANPOUR, B., NGUYEN, L., PHI HUNG, B. a NING. Oriented image stitching for spherical image content. *U.S. Patent Application*. 10/325,391.
- [10] HARRIS, C. a STEPHENS, M. *A Combined Corner and Edge Detector*. 1988. Doi:10.5244/C.2.23.
- [11] HONSNEJMAN, P. *JPEG formát a vše o tomto letitém fenoménu* [online]. 2016. Dostupné z: <https://moje.tajemno.net/jpeg/>.
- [12] HOWSE, J. *Learning OpenCV 4 computer vision with Python 3 : get to grips with tools, techniques, and algorithms for computer vision and machine learning*. Birmingham: Packt Publishing, 2020. ISBN 1789530644.

- [13] JANČÁLEK, P. a JANČÁLEK, R. *Základy neuroanatomie a nervových drah*. I. 2., přeprac. vyd. Brno: Masarykova univerzita, 2014. ISBN 978-80-210-7426-2.
- [14] KUČYHNKA, P. *Oční lékařství*. 2 přeprac. a dopln. vyd. Praha: Grada Publishing, 2016. ISBN 978-80-247-5079-8.
- [15] LINDBERG, T. *Scale-Space Theory in Computer Vision*. Leden 1994. ISBN 0-7923-9418-6.
- [16] LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*. Hingham, MA, USA: Kluwer Academic Publishers. listopad 2004, sv. 60, č. 2, s. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94. ISSN 0920-5691. Dostupné z: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [17] MATHER, J. *Adding  $L^*$  to RGBG* [online]. 2008 [cit. 2020-28-05]. Dostupné z: <http://jeffmatherphotography.com/dispatches/2008/05/adding-t-to-rgb/>.
- [18] OTOMAR, K. *Lékařská fyziologie*. Praha: Grada Publishing, 2011. ISBN 978-80-247-3068-4.
- [19] PERONA, P. a MALIK, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1990, sv. 12, č. 7, s. 629–639.
- [20] RANKOV, V., LOCKE, R., EDENS, R., BARBER, P. a VOJNOVIC, B. An Algorithm for image stitching and blending. *Carol J. Cogswell*. Březen 2005, sv. 5701. DOI: 10.1117/12.590536.
- [21] REICHMANN, M. *Understanding RAW Files Explained - Luminous Landscape* [online]. 2011 [cit. 2020-28-05]. Dostupné z: <https://luminous-landscape.com/understanding-raw-files-explained/>.
- [22] SVOBODA, T., CHRUM, O. a MATAS, J. *RANSAC RANdom SAmple Consensus* [online]. 2009. Dostupné z: [https://cw.fel.cvut.cz/old/\\_media/courses/a3m33iro/ransac.pdf](https://cw.fel.cvut.cz/old/_media/courses/a3m33iro/ransac.pdf).
- [23] SZELISKI, R. *Computer vision - Algorithms and Applications*. Springer, 2011. ISBN 1848829345.
- [24] TAREEN, S. A. K. a SALEEM, Z. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. Březen 2018. DOI: 10.1109/ICOMET.2018.8346440.
- [25] TIŠNOVSKÝ, P. *PNG is Not GIF - Root.cz* [online]. 2006. Dostupné z: <https://www.root.cz/clanky/png-is-not-gif/#k02>.
- [26] WEICKERT, J. Efficient image segmentation using partial differential equations and morphology. *Pattern Recognition*. 2001, sv. 34, č. 9, s. 1813–1824.
- [27] ČECH, D. a ČECH., S. *Mikroskopická anatomie*. 3 přeprac. vyd. Brno: Masarykova univerzita, 2011. ISBN 978-80-210-5550-6.
- [28] ŠPANĚL, M., HULÍK, R. a HAVEL, J. *Základy počítačové grafiky - Geometrické transformace ve 2D a 3D* [online]. 2019. Dostupné z: [https://www.fit.vutbr.cz/study/courses/IZG/private/lecture/izg\\_slide\\_transformace\\_print\\_rev2019.pdf](https://www.fit.vutbr.cz/study/courses/IZG/private/lecture/izg_slide_transformace_print_rev2019.pdf).