



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**ŘÍZENÍ A AUTOMATIZACE OPTICKÉ  
JEDNOTKY AUTOFOKUSU PRO TRANSMISNÍ  
HOLOGRAFICKÝ MIKROSKOP**

CONTROL AND AUTOMATION OF THE AUTOFOCUS OPTICAL UNIT FOR A TRANSMISSION

HOLOGRAPHIC MICROSCOPE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MAREK PATÁK,**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. ZBYNĚK DOSTÁL, Ph.D.**

BRNO 2023



## **Abstrakt**

Bakalářská práce se zabývá automatizací autofokusační jednotky pro transmisní holografický mikroskop. V práci je popsána konstrukce stávající autofokusační jednotky, která je doplněna o robotizaci kolimátoru pomocí řemenového mechanismu. V práci je uveden algoritmus pro automatizaci kolimátoru. Dále je popsán algoritmus pro vyhodnocení míry rozladění mikroskopu. Pro tyto funkce je vytvořeno grafické rozhraní. Algoritmy byly experimentálně otestované na laboratorní sestavě pro další integraci do řídicího programu holografického mikroskopu.

## **Abstract**

The bachelor's thesis deals with the automation of the autofocus unit for a transmission holographic microscope. The work describes the construction of the existing autofocusing unit, which is supplemented by the robotization of the collimator using a belt mechanism. The work presents an algorithm for automating the collimator. An algorithm for evaluating the microscope detuning is also described. A graphical interface is created for these functions. The algorithms were experimentally tested on a laboratory setup for further integration into the control program of the holographic microscope.



## **Klíčová slova**

zpracování obrazu, automatizace, autofokusační jednotka, holografická mikroskopie

## **Keywords**

image processing, automation, autofocus unit, holographic microscopy



## **Citace**

PATÁK, Marek. *Řízení a automatizace optické jednotky autofokusu pro transmisní holografický mikroskop*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Ing. Zbyněk Dostál, Ph.D.





## **Prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná a že jsem ve své práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

.....

Marek Paták  
25. května 2023



## **Poděkování**

Rád bych poděkoval mému vedoucímu Ing. Zbyňkovi Dostálovi, Ph.D, za cenné rady a vedení při vytváření práce. Dále děkuji Sonymu za pomoc s justáží a laděním sestav. Nakonec bych chtěl poděkovat kolegům s firmy Telight za možnost podílet se na této práci a za jejich konzultace a motivaci k dokončení práce.





# Obsah

<b>Úvod</b>	<b>3</b>
<b>1 Cíle práce</b>	<b>11</b>
<b>2 Optomechanická sestava</b>	<b>13</b>
2.1 Sestava protipu autofokusační jednotky . . . . .	13
2.2 Postup při užívání autofokusační jednotky . . . . .	13
2.3 Laboratorní sestava . . . . .	14
2.4 Použitý software . . . . .	14
<b>3 Automatizace kolimátoru</b>	<b>17</b>
3.1 Návrh řemenového mechanismu . . . . .	17
3.2 Algoritmus pro nalezení ideální stopy . . . . .	18
<b>4 Algoritmus pro udržení ostrosti</b>	<b>25</b>
<b>5 Grafické rozhraní pro ovládání jednotky</b>	<b>29</b>
<b>6 Experimenty</b>	<b>31</b>
6.1 Ověření funkčnosti nalezení ideální stopy . . . . .	31
6.2 Test přenosti algoritmu pro udržení ostrosti . . . . .	31
<b>Závěr</b>	<b>33</b>
<b>Literatura</b>	<b>34</b>



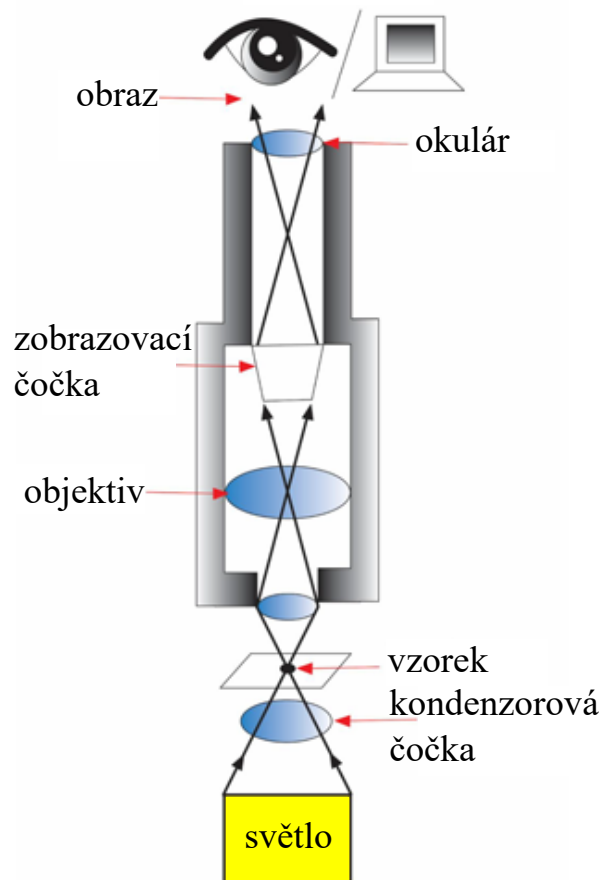
# Úvod

Jednou z hlavních aplikací světelných mikroskopů je pozorování živých organismů. Zvětšení mikroskopu umožňuje vědcům zkoumat stavbu nepopsaných mikroorganismů, popřípadě v lékařství mohou být využity k diagnostice. Viditelné světlo není při běžné intenzitě osvitu fototoxické a tudíž pokud je vzorek připraven v náležitém preparátu mohou mikroskopy pozorovat několika minutové až denní procesy živých organismů. Díky tomu mohou vědci zkoumat interakci mezi buňkami nebo procesy jako je například buněčné dělení. Při těchto dlouhodobých pozorováních, ale ve světelném mikroskopu dochází vlivem změn okolních podmínek (především malé změny teploty) nebo vlivem procesů živých buněk k postupnému rozostřování obrazu. Proto je potřeba udržovat dlouhodobé zaostření obrazu, a to prakticky pro všechny techniky světelných mikroskopů.

**Mikroskopie živých buněk** se zabývá zobrazováním objektů, které nejsou viditelné pouhým okem, jako jsou například živé buňky, tkáně a mikroorganismy. Na tato pozorování se nejlépe hodí světelné mikroskopy, jejichž cílem v této oblasti je pozorovat vzorky a zároveň získat detailní informace o jejich morfologii na mikroskopické úrovni. Na buňku jako takovou se z chemického hlediska nahlíží jako na membránami ohraničený vodný roztok, ve kterém se nachází především proteiny, lipidy a cukry, jejichž hmotnost je označována jako *suchá hmota buňky* [1]. Právě tyto látky jsou pro biology nejvíce zajímavé pro sledování. Jedná se, ale o látky amplitudově nekонтрастní, díky čemuž nijak neovlivní intenzitu prošlého světla. V čem se naopak liší od zbytku sledovaného preparátu je index lomu. To zapříčiní, že se světelnému paprsku po jeho průchodu mění fáze, nikoliv amplituda. Skrze to je problematické zobrazit živou buňku běžným světelným mikroskopem, ale je potřeba využít mikroskop, který umí měřit i fázi.

Tyto metody konstrukčně vychází z **metody světlého pole**. Ta je nejjednodušší mikroskopová metoda. Je založena na tom, že sledovaný vzorek absorbuje světlo a výsledný obraz je vytvořen pouze rozdílem amplitudového kontrastu sledovaného preparátu. Sestava je složena ze zdroje světla, kondenzoru, stolku se vzorkem, objektivu a okuláru viz obrázek 1. Plošný homogenní zdroj světla vytvoří svazek, který prochází kondenzorem a v rovině vzorku tak vytváří jeho obraz. Ve vzorku dojde k částečné absorpci světla podle toho, o jaký vzorek se jedná, tento upravený svazek poté prochází objektivem a okulárem. Tyto optické komponenty vytváří zvětšený obraz vzorku, který je okem promítán na sítnici lidského oka.

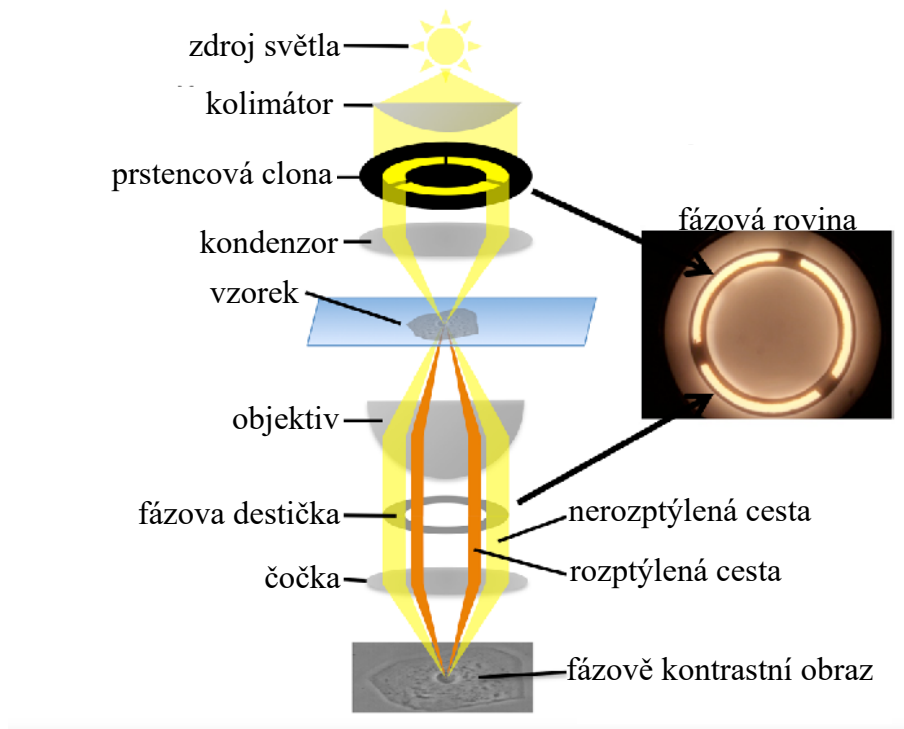




Obrázek 1: Schéma metody světelného pole převzato a upraveno z [2].

Z této metody jsou odvozeny metody pro pozorování **fázového kontrastu**. To jsou metody, které se zabývají vizualizací fáze pozorovaného vzorku, což je žádaný problém, neboť jak bylo již zmíněno výše transparentní vzorky nijak neovlivňují amplitudu světla a mění pouze jeho fázi. Touto problematikou se zabýval nizozemský fyzik Frits Zernike, který jako první vypracoval metody fázového kontrastu a zkonstruoval první fázový mikroskop [3, 4]. Za tento počín v roce 1953 obdržel Nobelovu cenu za fyziku. Metoda je založena na interferenci světla rozptýleného vzorkem a světla procházejícího vzorkem bez rozptylu. Sestava pro tento mikroskop je obdobná jako u metody světelného pole, ale před kondenzor je přidána prstencová clona a za objektiv je přidána fázová deska ve tvaru mezikruží viz obrázek 2. Světlo se po průchodu vzorkem rozdělí na světlo nerozptýlené (na obrázku žluté) a rozptýlené (na obrázku oranžové), které je zároveň vzorkem fázově posunuto. Pouze nerozptýlené světlo po průchodu objektivem prochází mezikružím fázové desky, která uměle posune fázi kladně nebo negativně podle toho, zda se jedná o pozitivní nebo negativní fázový kontrast. Interferencí tohoto dodatečně fázově posunutého světla a vzorkem rozptýleného, fázově posunutého světla, vznikne v obrazové rovině kontrast fázových objektů ve vzorku. Bohužel, ale tato metoda není schopna stanovit fázové posuny ve vzorku kvantitativně. Další nevýhodou této metody je halo-efekt, což je jev, který vzniká při velkých fázových

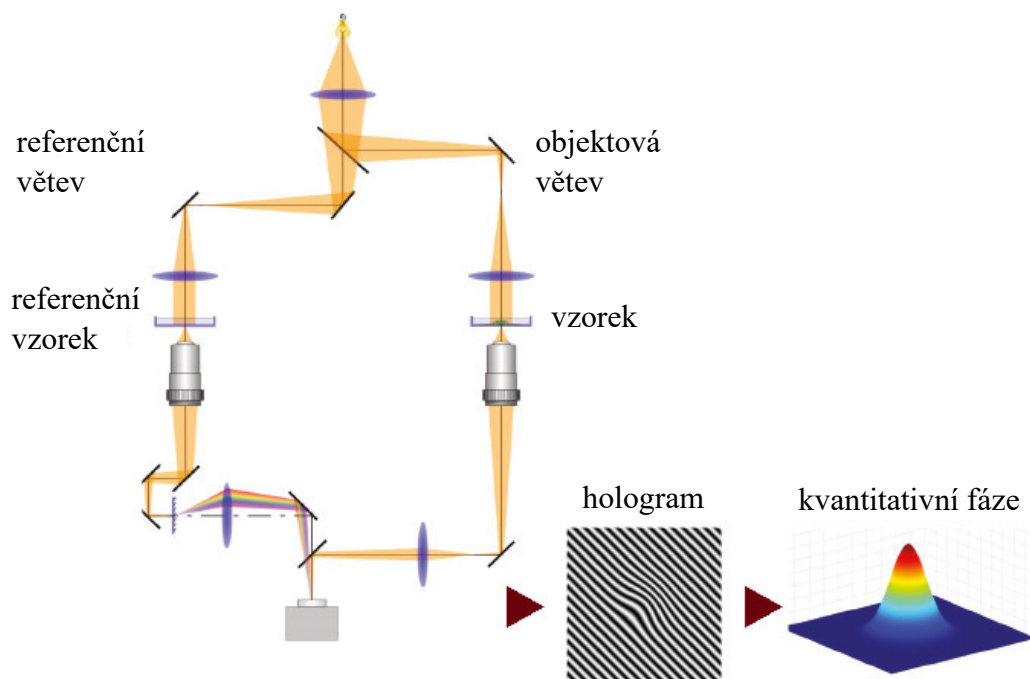
rozdílech vzorku a v těchto místech způsobuje ztrátu informace. Další metodou, která se zabývá fázovým kontrastem je například Nomarského diferenciální interferenční kontrast [5] nebo Hoffmanův modulační kontrast [6].



Obrázek 2: Schéma Zernikova fázového mikroskopu převzato a upraveno z [7].

Aby bylo možné fázi vlny procházející vzorkem měřit, jsou vyvíjeny **metody kvantitativního fázového zobrazování**. To jsou metody, které jsou schopny neinvazivně zobrazit živou buňku v jejím přirozeném prostředí a jsou schopny určit množství suché hmoty v buňce a její rozložení. Pro tyto požadavky je nejčastěji použita metoda digitální holografické mikroskopie [8]. Námi vyvíjený koherenci řízený holografický mikroskop (coherence controlled holographic microscope - CCHM) [9] se skládá se ze dvou optických cest. Cesta, ve které prochází svazek vzorkem, se nazývá objektová větev a druhá se nazývá referenční větev, viz obrázek 3. Jediné rozdíly mezi těmito větvemi je, že v objektové větvi je dán zkoumaný vzorek, v referenční větvi je vzorek nahrazen prázdným sklíčkem, také je v referenční větvi před výstupní čočkou difrakční mřížka, jinak se jedná o identické optické cesty. Interference světelných svazků těchto dvou větví vytvoří hologram, který je zaznamenán na CMOS senzor. Holografické snímky zaznamenávají vlnu jako celek, tím pádem jsme po rekonstrukci hologramu schopni určit jak amplitudu vlny, tak i její fázi, a to za využití *FFT metod* [10]. Rekonstrukcí hologramu jsme schopni určit suchou hmotu buňky a to s přesností na  $4,6 \text{ femtogram}/\mu\text{m}^2$  [11]. Díky této metodě a výkonnému hardwaru jsme schopni pozorovat živé buňky a získat veškeré informace, které potřebuje včetně dynamiky a vzájemné interakce buněk [12]. Pro dlouhodobé pozorování takto citlivých mikroskopů je potřeba vyřešit další problémy. První je, že musíme vytvořit prostředí, ve kterém budou buňky schopné dlouhodobě přežít a přirozeně fungovat, proto je potřeba kontrolovat teplotu, vlhkost a koncentraci  $\text{CO}_2$ . Tento problém se dá v dnešní době poměrně lehce konstrukčně vyřešit. I přes snahu udržet konstantní teplotu, ale může v mikroskopu docházet

k malým výkyvům teploty, což může zapříčinit druhý problém, a to je rozladování naladění hologramu a rozostřování obrazu. To zapříčini klesání holografického signálu. Na vyřešení tohoto problému byl mikroskop robotizován a byly přidány metody metody automatické justáže. Díky přidání těchto prvků je možné udržovat ostrost obrazu a vysoký holografický signál pomocí autofokusačních systémů.

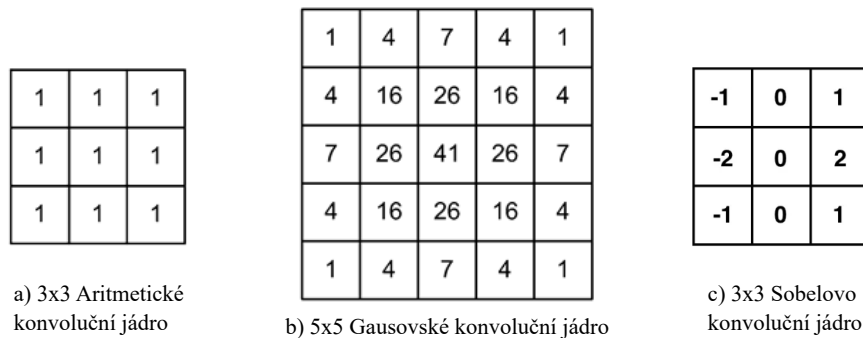


Obrázek 3: Schéma transmisního holografického mikroskopu [13].

**Autofokusační systémy** jsou nedílnou součástí dnešních mikroskopů, které jsou schopny dlouhých pozorování. Díky motorizovanému stolku jsou mikroskopy schopny automaticky udržovat vzorky v ideální pozici pro optimální zaostření. Pro zjištění ideální pozice jsou dva přístupy. První vyhodnocuje zaostření na základě zpracování obrazu a nazývá se softwarový autofokus. Druhý přístup pracuje s fyzickou polohou vzorku a nazývá se hardwarový autofokus.

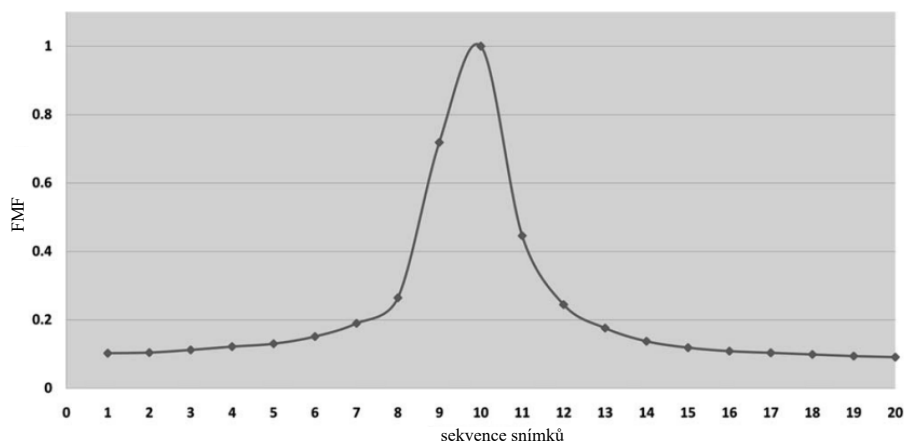
**Softwarový autofokus** používá funkce určující míru zaostření (focus measure function - FMF), pro určení zaostření. Obecně zjišťování informací z obrazu na softwarové úrovni spočívá v porovnání jednotlivých pixelů s pixely v jejich okolí, pro což se používá konvoluce a konvolučního jádra. Konvoluční jádro je čtvercová matice, která určuje, s jakou váhou bude daný pixel v okolí brán viz obrázek 4. Konvoluční jádro se aplikuje na každý pixel v obrazu čemuž se říká konvoluce, jejichž výsledkem bude nový obraz, který ulehčí získat požadované informace z obrazu. Většina dnešních algoritmů používá ve svých řešení těchto konvolucí několik, s několika různými konvolučními jádry viz obrázek 4 a to pro různé účely. Konvoluční jádra mohou být použity pro předzpracování obrazu jako je například rozmazání obrazu, čehož by bylo docíleno pomocí aritmetického konvolučního jádra (obrázek 4a)), kde se jedná o klasický aritmetický průměr, který by bral pixely z okolí se stejnou váhou jako daný pixel, na který je konvoluční jádro aplikováno nebo průměr pomocí Gausova rozložení (obrázek 4b)), kde pixely více vzdálené od sledované pixelu mají výrazně nižší váhu. Mezi

další základní aplikace konvolučních jader patří detekce hran v obrazu k tomu by mohl být použit například Sobelovo konvoluční jádro (obrázek 4c)), kde se porovnávají pixely vlevo od sledovaného pixelu vůči pixelů vpravo od sledovaného pixelu, což určí míru změny intenzity v horizontálním směru. Stejný postup se aplikuje pro vertikální směr, kde se berou hodnoty nad a pod pixelem a následně odmocněním součtu druhých mocnin těchto změn pro oba směry získáme gradient v daném bodě. Čím vyšší gradient je v daném místě, tím ostřejší hrana se zde vyskytuje.



Obrázek 4: a) konvoluční jádro pro klasický průměr b) konvoluční jádro pro gausianské rozložení c) konvoluční jádro pro Sobelův operátor.

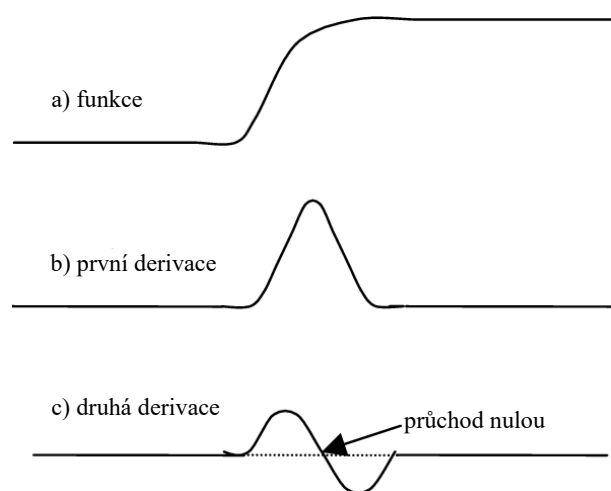
Algoritmus pro nalezení ideální roviny ostrosti pak spočívá v tom, že je vytvořeno několik snímků pro okolí aktuální polohy stolku a následně se snímky vyhodnotí. Po vynesení výsledků do grafu lze získat křivku s jedním výrazným maximem, ve kterém se nachází aktuální poloha pro optimální zaostření viz obrázek 5 [14]. Pro vyhodnocení ostrosti se používá několik různých metod, několik těchto metod se začalo již objevovat mezi roky 1970-80 [15, 16]. Pro co nejrychlejší nalezení optimální polohy je potřeba použít co nejméně kroků, to se dá řešit čistě pomocí matematické optimalizace [17, 18] nebo pomocí strojového učení [19]. Mezi dosavadní nepoužívanější druhy softwarových metod pro vyhodnocení ostrosti patří následující.



Obrázek 5: Výsledek softwarového autofokusačního algoritmu. Na svislé ose je vynesena výsledek FMF a na vodorovné ose je poloha stolku. Převzato a upraveno z [14].

**FMF založené na Gradientu** jsou metody které pracují s gradientem nebo první derivací obrazu. Výsledkem je přesná informace o ostrosti obrazu. Vychází z již dříve zmíněného principu, čím výraznější změna v gradientu obrazu, tím ostřejší je detekována hrana a tím ostřejší by měl být celý snímek [20].

**FMF založené na Laplaceově operátoru** [14] jsou metody fungující na stejném principu jako přechodové metody, akorát pracují s druhou derivací a ne první. Tudíž nedetekují maxima, ale detekují průchod funkce nulou viz obrázek 6. To nese výhody v tom, že se jedná o lineární operaci, ale na druhou stranu poskytuje pouze informaci o tom, kde se hrana nachází a neposkytne informaci o velikosti nebo směru hrany. Zároveň druhá derivace ještě více zesílí zaznamenaný obrazový šum.



Obrázek 6: a) funkce intenzity signálu b) první derivace signálu c) druhá derivace signálu.

**FMF založené na statistice** [14] jsou metody využívající směrodatné odchylky, rozptyl a autokorelaci k výpočtu stupně zaostření obrazu. Díky tomu nejsou tak náchylné na šum oproti metodám používající derivace.

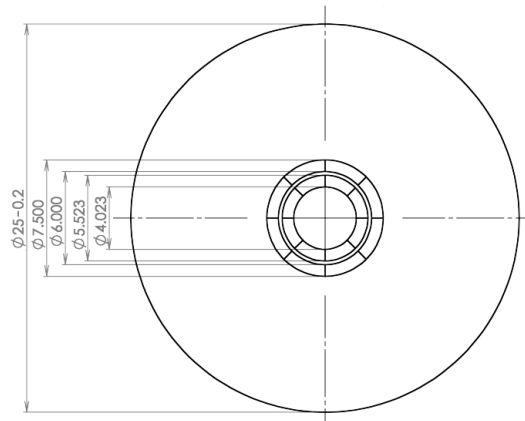
**FMF založené na strojovém učení** pracují s metodou hloubkového učení (deep learning). Tato metoda funguje na principu učení s učitelem (supervised learning), ten spočívá v navrhnutí architektury sítě, přípravy tréninkových dat, trénování sítě a případné ladění. Pro architekturu se ve zpracování obrazu používají převážně konvoluční neuronové sítě, které pracují s konvolučními jádry. Tréninková data se skládají z již nasnímaných snímků, které se musí ručně naanotovat a na těchto naanotovaných snímcích se natrénuje model. Poté co je model natrénovaný a vyladěný je schopný z jediného snímku určit míru zaostření a vzdálenost od optimální polohy [21]. Nevýhodou je náročnost sesbírání vhodných tréninkových dat, poté jejich následná anotace. Navíc mikroskopy používá několik různých vědců, kteří mají rozlišné vzorky.

Obecně ale všechny softwarové metody selhávají v tom, že určují ostrost podle obrazu. Buňky se, ale v průběhu měření mohou pohybovat, a to jak horizontálně, tak vertikálně, z čehož vyplývá, že každá buňka může mít jinou rovinu zaostření. Tento jev lze pozorovat například při buněčném dělení, kdy se buňka zakulatí a vzdálí se od roviny zaostření. Je-li softwarový autofokus zaměřen právě na takovou buňku, dojde k rozostření zbytku obrazu, což může poškodit data celého pozorování.

Tuto problematiku řeší **hardwarový autofokus**. Ten pracuje s fyzikální polohou sklíčka, na kterém jsou umístěny buňky. Jeho metody se tedy zabývají měřením této polohy. Pro

tato měření se používá světlo v blízkosti infra červeného spektra, neboť viditelné světlo je v cestě mikroskopu použito pro zobrazení vzorku. Toto světlo je do mikroskopu přivedeno pomocí dichroického zrcátka. Pro řešení této problematiky bylo vyvinuto několik metod. A to například pomocí dírkové komory [22, 23] nebo sestava pracující s půlkruhovými maskami vyvinutými panem Reinhamerem [24], ze které vychází systém od firmy Nikon *Perfect Focus System* [22, 25].

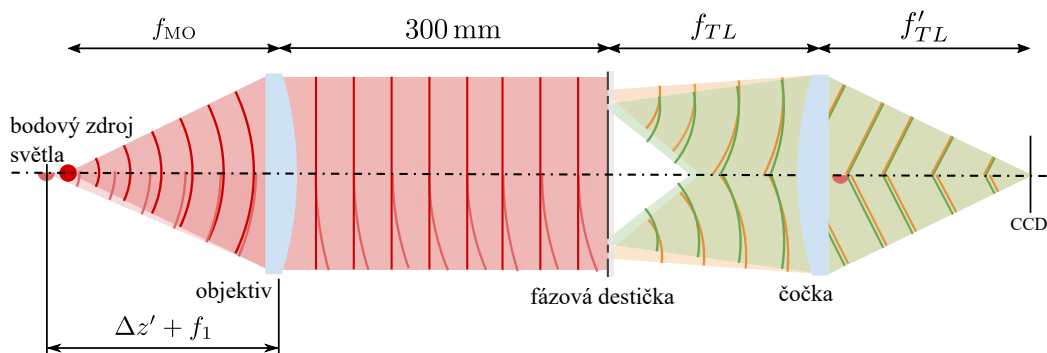
Námi vyvíjený systém vychází z metody vírových svazků [26, 27]. Vír v optickém slova smyslu je typem fázové nespojitosti, ve kterém má vlnoplocha tvar šroubovice postupující okolo nespojitého centra na optické ose [28]. Pro vznik optických víru je potřeba posunout fázi svazku, toho je v našem řešení dosaženo pomocí fázové destičky viz obrázek 7, která se skládá z dvou mezikruží, kde každé má 8 zón ve kterých, dojde k mírně rozlišnému posunutí fáze.



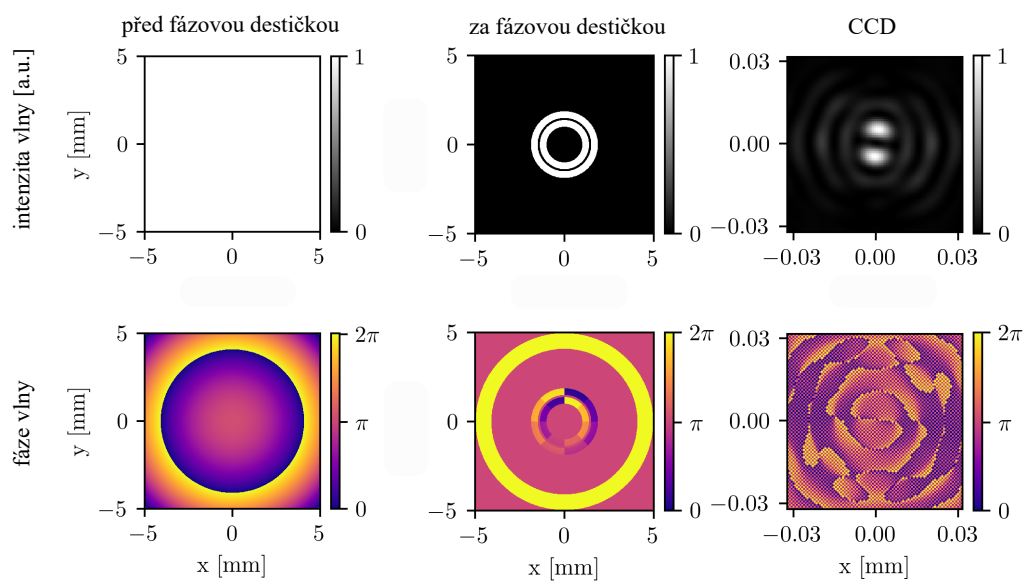
Obrázek 7: Fázová destička převzato z [28].

Mezikruží generují vírové vlny s opačným topologickým nábojem (s opačným smyslem stoupání šroubovice vlnoplochy). Interferencí těchto vln za čočkou vzniká stopa s dvoučetnou symetií. Celkové znázornění optické sestavy, intenzity a fáze vlny ve vybraných polohách jsou graficky znázorněny na obrázku 8. Bodový zdroj světla vytvoří kulovou vlnu, která je po průchodu objektivem kolimovaná, následně po průchodu destičkou vzniknou 2 vírové vlny se stejnou fází. Vlny jsou čočkou fokusovány na kameru. Pokud dojde k mírnému posuvu zdroje světla  $\Delta z'$  (v našem systému se bude jednat o posun sklíčka) vlna po průchodu objektivem bude mírně divergentní nebo konvergentní, což zapříčiní rozdíl fáze vírových svazků a to se projeví jako rotace vzniklé stopy. V této práci budu pracovat z obrazem intenzity stop, která bude snímána na CCD kameru, jak můžeme vidět na obrázku 8b).

a)



b)



Obrázek 8: Optické schéma paprsku při průchodu fázovou destičkou převzato a upraveno z [28].

# 1 Cíle práce

Jak bylo v úvodu zmíněno, je systém autofokusu pro CCHM nezbytný. Není-li použit při dlouhodobých pozorování, klesá holografický signál a CCHM ztrácí kontrast fázového obrazu. V rámci vývoje byl navržen a ozkoušen nový systém autofokusu s využitím výrových svazků. K němu je nezbytné vytvořit sadu algoritmů pro zpracování obrazu rotující stopy a tak určovat rozostření obrazu a míru výsledné korekce. Aby bylo možné toho dosáhnout musím provést tyto jednotlivé kroky:

1. automatizace kolimátoru,
2. vyvinutí algoritmu pro udržení ostrosti,
3. realizace jednoduchého grafického rozhraní pro ovládání jednotky,
4. experimentální ověření vyvinutých funkcí.





## 2 Optomechanická sestava

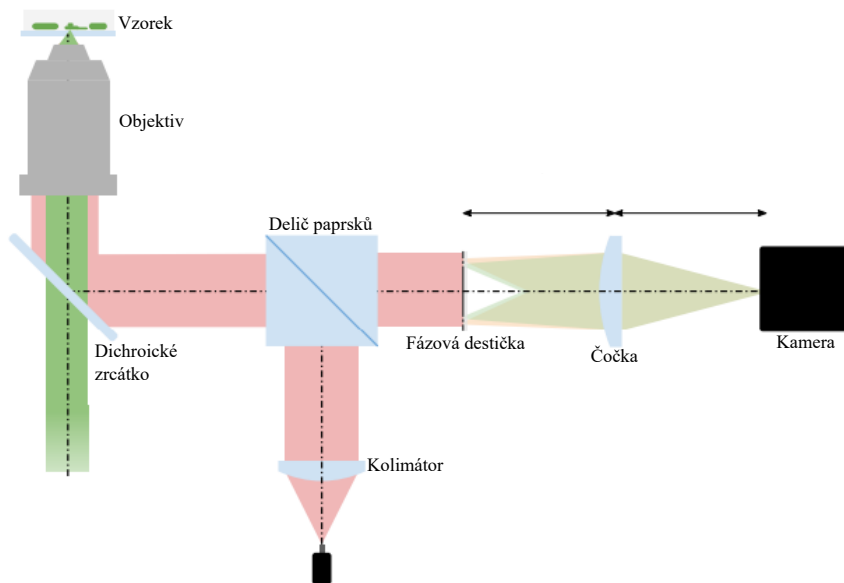
V této části budou popsány dvě sestavy, postup při používání jednotky a použitý software. První sestava je funkční prototyp autofokusační jednotky. Ten, ale lze testovat pouze ve spolupráci s mikroskopem, což skrze probíhající vývoj nového mikroskopu není zatím možné. Skrze tento problém byla sestavena laboratorní sestava, která simuluje funkční prototyp autofokusační jednotky a je na ní tedy možné plně testovat algoritmy.

### 2.1 Sestava prototypu autofokusační jednotky

Sestava viz obrázek 2.2 je založena na metodě interference vřivých svazků, jak již bylo popsáno v úvodu. Princip sestavy spočívá v tom, že laserové světlo v blízkosti infra červeného spektra je z optického vlákna kolimováno a poté je děličem paprsků a dichroickým zrcátkem odraženo do objektivu. Objektiv vytvoří obraz vlákna na spodní části sklíčka se vzorkem, který se nachází na motorizovaném stolku. Část světla je od sklíčka odražena a stejnou optickou cestou se vrátí zpět. Po průchodu děličem v tomto směru pokračuje paprsek do fázové destičky a po jejím průchodu je čočkou fokusován na kameru. Tato sestava vychází z původní práce [28] a schéma ze kterého jsme vycházely lze vidět níže viz obrázek 2.1. Do této sestavy byl ještě před kameru přidán objektiv, neboť rotující stopa generovaná bez objektivu byly moc malé pro přesné počítačové zpracování. Dále byl do sestavy přidán mechanismus, který umožní ovládání kolimátoru pomocí počítače, to je popsáno ve 3 části. Co se týče neoptických prvků bylo potřeba vybrat vhodný servo motor na pohánění mechanismu kolimátoru a kameru. Výběr servo motoru je popsán ve třetí části. Při výběru kamery byli důležité hlavně dvě věci. Zaprvé kamera musí být citlivá na světlo v blízkosti infra červeného záření. Druhý důležitý faktor je velikost pixelu, pro naše algoritmy chceme co nejmenší velikost pixelu. Skrze tyto faktory byla vybrána monochromatická kamera od firmy Basler daA3840-45um, která má velikost pixelů 2  $\mu\text{m}$  a má dostatečnou citlivost na světlo v blízkosti infra červeného záření. Kamera má rozlišení 3840x2160, ale vzniklá informace se nachází pouze na cca 100x100 pixelech uprostřed obrazu. Proto budeme používat výřez obrazu (region of interest - ROI) a to pouze 500x500 pixelů ze středu obrazu. toto nastavení probíhá hned při inicializaci kamery. Díky použití ROI je kamera schopná snímat rychlostí přes 100 FPS a dojde i k urychlení algoritmů, neboť budou pracovat s menší maticí. Na tomto základě by naše jednotka neměla mít problém s rychlostí. Kameře dále nastavíme bitvou hloubku na 8 bitů, což znamená, že budeme pracovat s úrovněmi šedi v rozmezí mezi 0-255. Celkové nastavení a práce s kamerou je k nahlédnutí v příloze práce v souboru Camera\_control.cpp/.h.

### 2.2 Postup při užívání autofokusační jednotky

Před použitím autofokusační jednotky bude potřeba naladit mikroskop. To udělá uživatel pomocí grafického rozhraní mikroskopu, kde pomocí kombinace automatických procedur a manuálního ladění najde nejlepší rovinu ostrosti. Následně uživatel pustí experiment. Než začne snímat, musí se jednotka samostatně naladit na aktuální polohu pozorovaného



Obrázek 2.1: Schéma původní sestavy autofokusu převzato a upraveno z [28]

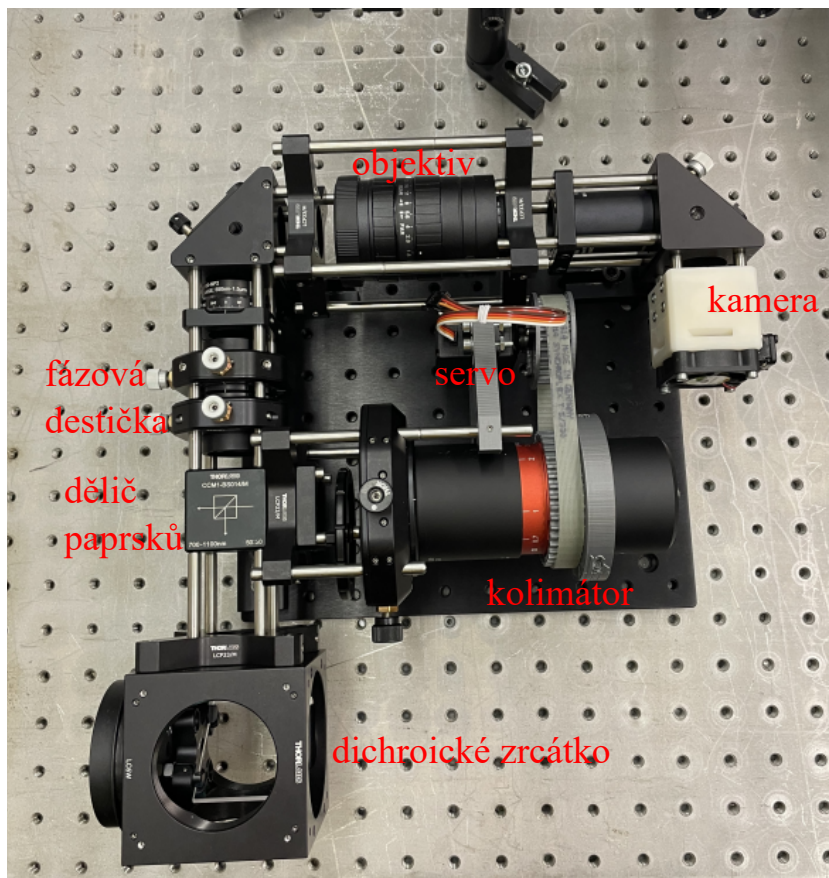
sklíčka. Proto bude v jednotce kolimátor automatizován. Jeho laděním lze najít počáteční referenční polohu, to znamená co nejvýraznější a nejsouměrnější rotující stopa. Z tohoto obrazu vychází algoritmus pro udržení ostrosti. Ladění kolimátoru je podrobněji popsáno ve třetí části. Následné udržování ostrosti pak bude spočívat v určování natočení během experimentu a porovnání tohoto zjištěného natočení s původním referenčním natočením. Pomocí určení rozdílu těchto dvou natočení a komunikací s mikroskopem bude jemným posouváním motorizovaného stolku dosaženo udržování ideální ostrosti. Určování natočení půlměsíček je popsáno ve čtvrté části.

## 2.3 Laboratorní sestava

Pro vývoj a testování algoritmů je potřeba sestava ve které lze simulovat, jak budou vypadat snímky pro automatizaci kolimátoru a vytvořit dataset z okolí již naladěné stopy pro algoritmus k určování natočení. Na základě toho byla sestavena experimentální sestava viz obrázek 2.3. Jedná se o zjednodušenou sestavu, kde je možné jedním manuálním posuvem simulovat jak kolimátor, tak rozostření obrazu. Zdroj světla je zde helium-neonový (He-Ne) laser. Ten je z optického vlákna fokusován do dírkové clony mikroskopového objektivu. S tím můžeme manipulovat manuálním posuvem a tím budeme simulovat rotaci stopy a ladění kolimátoru. A za tímto objektivem je již sestava stejná jako u prototypu za děličem paprsků.

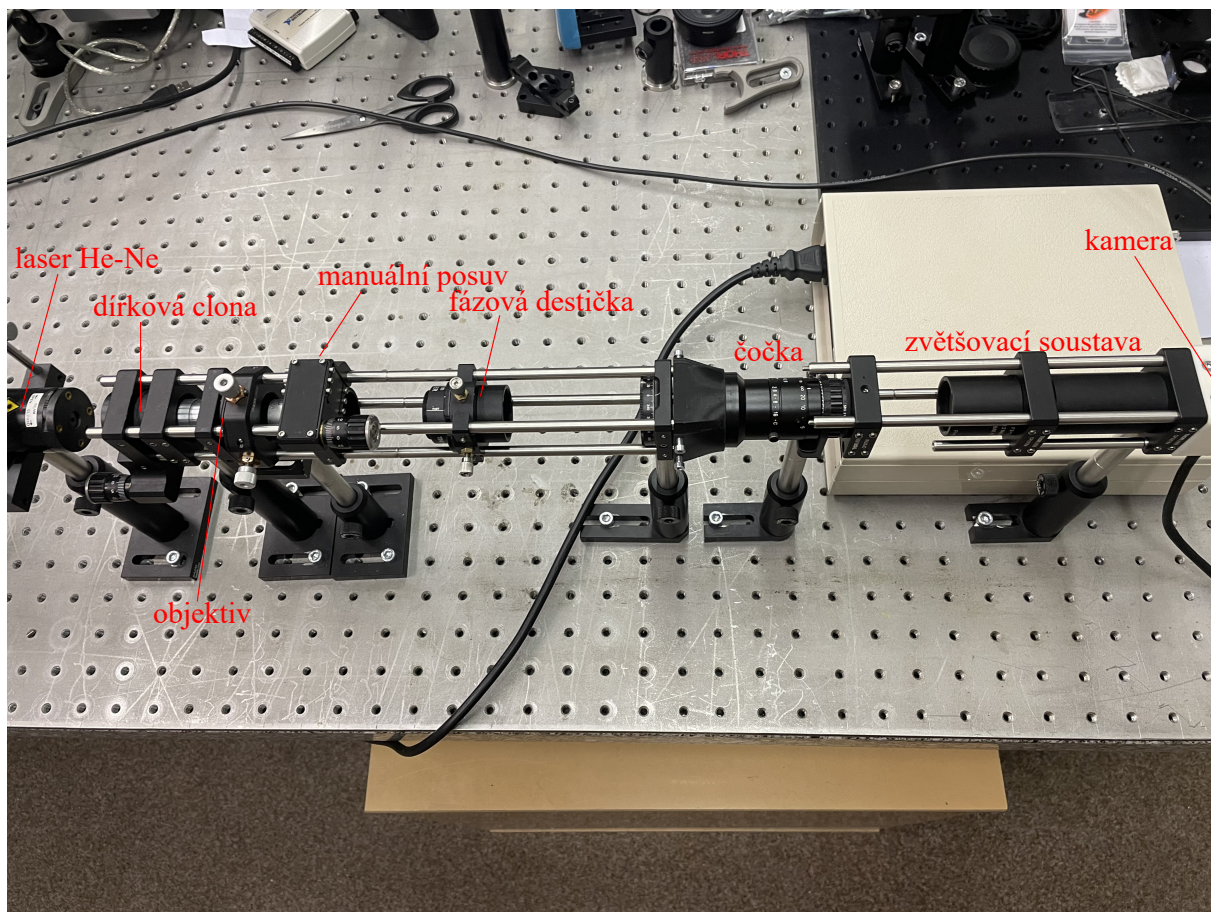
## 2.4 Použitý software

Veškerý kód, který bude v této práci použit, jsem psal v jazyce C++ pomocí Visual Studio IDE. Pro práci s obrazem jsem použil framework OpenCV a pro vytvoření grafického rozhraní jsem použil framework Qt. Pro komunikaci s kamerou jsem použil drivers od firmy Basler a pro nastavení kamery jsem použil funkce z jejich API. Servo motor je řízen pomocí



Obrázek 2.2: Prototyp autofokusu.

PWM signálu, který bude generován z řídicí desky mikroskopu, tudíž jeho řízení není zahrnuto v zdrojovém kódu, neboť zdrojový kód pro ovládání CCHM není volně přístupný. Pro experimentální ověření funkčnosti řemenového mechanismu bylo použito pouze Arduino a vzorový kód z technické dokumentace motoru [29].



Obrázek 2.3: Experimentální sestava pro vývoj a testování.

## 3 Automatizace kolimátoru

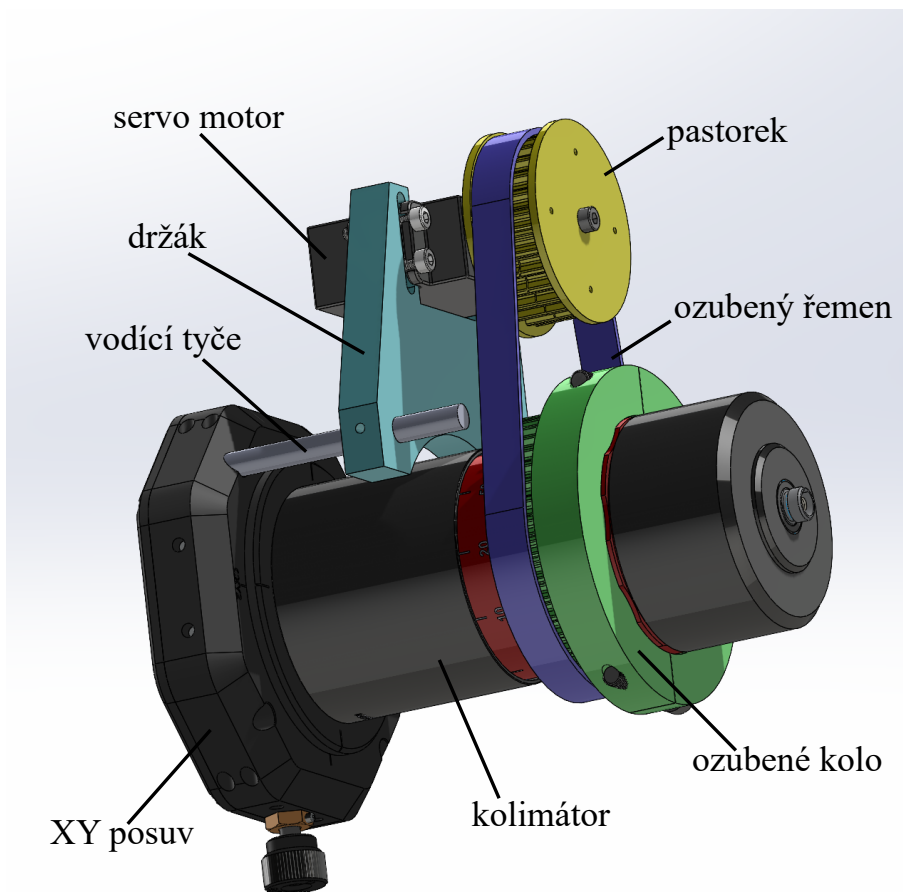
Pro naši jednotku byl vybrán nastavitelný kolimátor C80APC-A od firmy Thorlabs viz obrázek 3.1. Jeho červenou částí, lze otáčet, čímž je dosaženo změny vzdálenosti zaostření. V této části je popsán návrh mechanismu pro otáčení kolimátoru pomocí počítače a jak funguje algoritmus pro naladění autofokusační jednotky na mikroskop.



Obrázek 3.1: Nastavitelný kolimátor od firmy Thorlabs.

### 3.1 Návrh řemenového mechanismu

Po konzultaci s vedoucím práce jsme se na základě jeho zkušeností s touto problematikou rozhodl pro řemenový mechanismus. Model řemenového mechanismu lze vidět níže viz obrázek 3.2 a reálné připojení do sestavy již bylo vidět výše viz obrázek 2.2. Nejdříve jsem pomocí momentového klíče zjistil, že na otáčení pohyblivé části kolimátoru je potřeba vyvinout moment o velikosti 0,45 Nm. Tuto hodnotu jsem ještě pro zajištění funkčnosti vynásobil bezpečností 2. Pracoval jsem tedy s tím, že pro otáčení je potřeba 0,9 Nm. Na tomto základě jsem vybral servo motor DSS-M15S, tento motor by podle specifikací měl vyvinout moment 1,5 Nm, tím pádem by měl být schopen za pomoci řemenového mechanismu nastavovat kolimátor. To jsem experimentálně ověřil pomocí Arduina [29]. Servo je řízeno pomocí pulzně šířkové modulace (PWM). Servo má i analogovou zpětnou vazbu, ze jejichž pomoci jsme schopni ověřit, že došlo k požadované rotaci. Servo je schopné otáčení v rozmezí 270°. Pro jednotku je potřeba nastavovat kolimátor v rozmezí 180°, skrze to byl zvolen převod do pomalu s převodovým poměrem 1,5. Ozubené kolo má 42 zubů a pastorek má 28 zubů. Pro přenos momentu jsem zvolil ozubený řemen 10 T5 330 CONTI® SYNCHROFLEX od firmy Continental [30], o délce 330 mm a šířce 10 mm.



Obrázek 3.2: Model řemenového mechanismu.

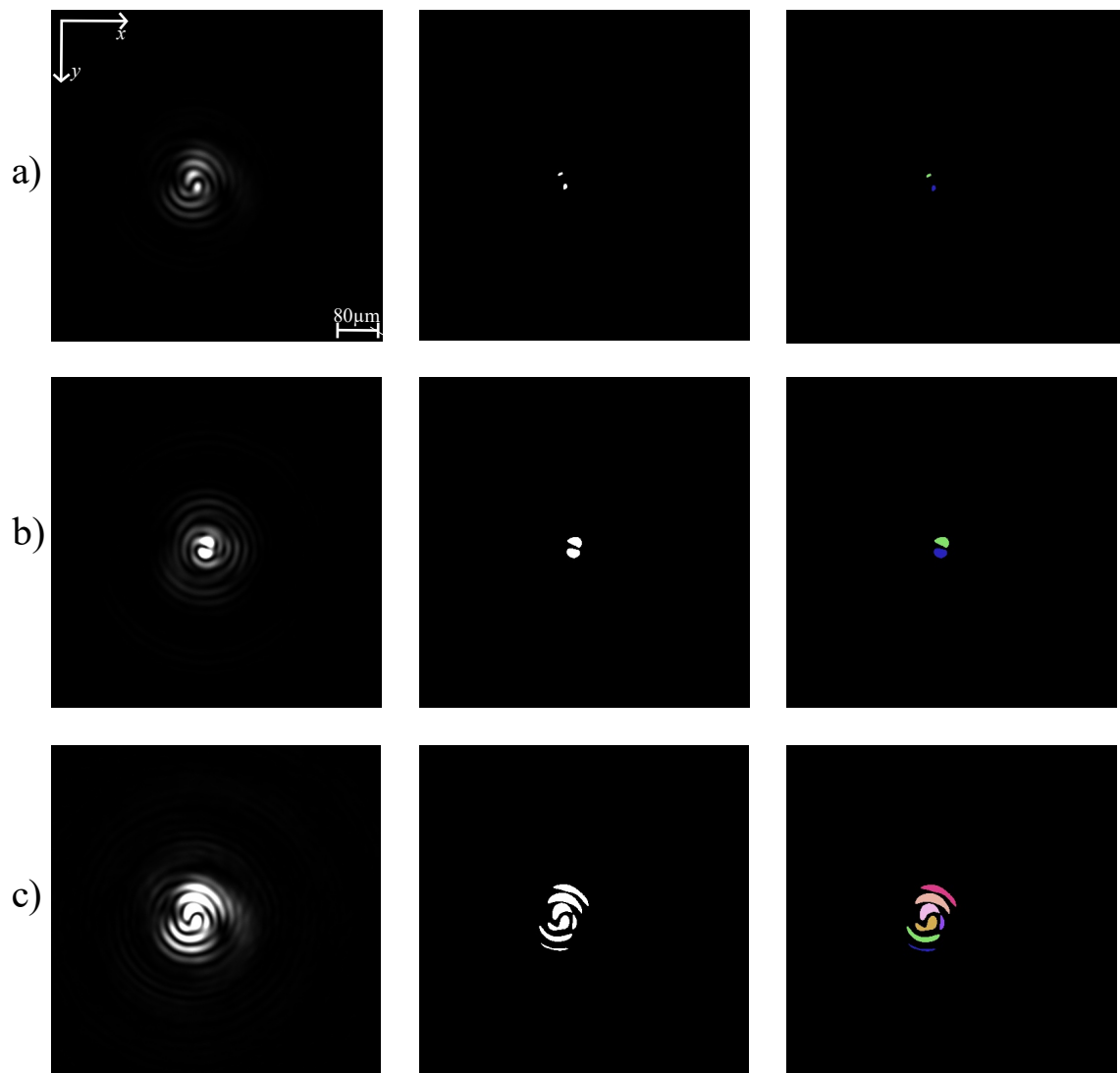
### 3.2 Algoritmus pro nalezení ideální stopy

Algoritmus bude jako první krok aplikovat na získaný snímek threshold. Threshold je funkce, která podle zadaného vstupního parametru vytvoří binární snímek. Toho docílí tím, že hodnotu každého pixelu porovná se vstupním parametrem, pokud je hodnota pixelu menší pixel bude černý, pokud je hodnota pixelu větší pixel bude bílý. Aplikace thresholdu lze vidět na obrázku 3.3, kde druhý snímek ukazuje, jak vypadá výstup thresholdové funkce. Touto funkcí se ze snímku odstraní nepotřebná informace. Z principu thresholdu vychází, že jeho výsledek se bude znatelně měnit s dobou expozice kamery. Jak hranice thresholdu, tak doba expozice jsou hodnoty, které musíme určit. Pro automatizaci nastavení těchto hodnot je potřeba jednu hodnotu určit konstantně. V našem případě to bude threshold. A to protože pro robustní funkci našeho systému, je potřeba dostávat snímky, které budou mít výrazné rozdíly mezi důležitou informací v našem případě vzniklé půlměsíčky a nedůležitou informací, která se skládá převážně ze vzniklého šumu. Cíl je tedy mít půlměsíčky, jejichž intenzita pixelů je těsně před saturací a šum, který má hodnotu co nejnižší pod hranicí thresholdu. Tohoto cíle bude dosaženo nalezením ideální doby expozice pro dané nastavení mikroskopu. Pomocí testování bylo zjištěno, že ideální konstantní hodnota thresholdu pro naši aplikaci je úroveň šedi o hodnotě 220.

Algoritmus pro nalezení ideální stopy má tedy za cíl najít ideální expozici pro dané nastavení mikroskopu a polohu kolimátoru, při které vzniká nejlepší počáteční poloha. Nyní bude rozebráno, jak přesně algoritmus funguje, během popisu se budu odkazovat na zdrojový kód, který je přiložen níže viz výpis 3.1. Jako první věc nastavíme globální proměnou `ref_found` na `false`. Díky tomu při příštím zavolání funkce pro určení natočení, bude výsledný úhel nastaven jako referenční. To znamená, že tento úhel bude brán jako úhel ve, kterém je vzorek zaostřen, a tudíž při rozostřování během pozorování, což zapříčiní mírnou rotaci, budeme hýbat stolkem tak, aby se natočení půlměsíček vrátilo zpět na tento referenční úhel. Další nastavení se týká globálních proměnných `curr_size` a `curr_diff` tyto proměnné budou použity níže, ale nejdříve je potřeba je nastavit na tyto hodnoty, kdyby byli změněny z předchozího použití tohoto algoritmu a důvod nastavení bude vysvětlen také níže pro lepší pochopení. Následně si vytvoříme tři lokální proměnné, které blíže popíšu, až budou použité. Poté již můžeme začít hledat ideální expozici a polohu. Toho bude dosaženo pomocí dvou for cyklů, jeden zařizuje nastavení polohy a druhý přepíná expozice kamery. Cyklus pro nastavení expozice je vnořený do cyklu pro nastavení polohy, z čehož vyplývá, že na každé poloze se vytvoří snímky pro všechny testované expozice a až poté se otočí kolimátor na další pozici. Expozice jsou uloženy v globální proměnné objektu kamery `exposure`, což je vektor o 4 místech, kde na prvním místě bude uložena výsledná ideální expozice a zbylé tři místa jsou expozice, které se testují v tomto algoritmu. Expozice bude nastavovat program mikroskopu, který bude mít uloženy v paměti pro jaké objektivy má testovat jaké rozmezí expozic a ty ještě před zavoláním toho algoritmu nastaví do této proměnné. Nyní k tomu, co se bude dít v cyklech. První vnější cyklus bude pouze otáčet motorem řemenového mechanismu, pro to bude použita funkce programu mikroskopu, která jak již bylo výše zmíněno není veřejná, je zde tedy pouze komentář. Vnitřní cyklus pak začne tím, že nastaví expozici odpovídající dané iteraci a pro tu vytvoří snímek. Funkce pro vytvoření snímku vrací hodnotu 1 pokud vše proběhlo v pořádku a hodnotu 0 pokud by došlo k chybě. Proto zkontrolujeme, že vše proběhlo v pořádku podmínkou. Snímek je uložen do globální proměnné objektu kamery `pbuffer`. Celkové nastavení a práce s kamerou je k dohledání v příloze bakalářské práce v souboru `Camera_control.cpp/.h`. Vytvořený snímek načteme do lokální proměnné `src`, která je typu `Mat`, který používá OpenCV pro práci s obrazem. Na vytvořený snímek následně aplikujeme `threshold`, což je funkce již popsána výše. Na binární obrázek vytvořený `thresholdem` aplikujeme funkci `findContours` od OpenCV. Tato funkce má šest vstupů. První obsahuje snímek, na který chceme funkci aplikovat. Do druhého a třetího budou uloženy výsledky této funkce. A čtvrtý, pátý a šestý nastavují v jakém módu bude funkce pracovat. Funkce v binárním obraze najde všechny obrysy. Obrysy lze jednoduše vysvětlit jako křivku spojující všechny spojitě body (podél hranice), které mají stejnou barvu nebo intenzitu [31]. Nalezené obrysy jsou uloženy do proměnné `contours` jako vektory bodů. Skrze to, že používáme mód `RETR_LIST` funkce nepracuje s hierarchií obrysů, a tudíž třetí proměnná hierarchie pro nás není důležitá. A nastavení módu `CHAIN_APPROX_SIMPLE` zapříčiní, že vzniklé obrysy mají díky aproximaci plynulejší ohraničení. Šestý vstup nastavuje `offset`, který nepotřebujeme tudíž je nastaven na nulu. Přejít vytvořeného snímku na `threshold` a následné nalezení obrysů, je pro různé expozice graficky zobrazeno na obrázku 3.3. V dalším kroku podmínkou zkontrolujeme počet vzniklých obrysů. Pro naši aplikaci vyžadujeme, aby vznikaly pouze dva obrysy (obrázek 3.3b), pokud vznikne více je snímek přeexponován (obrázek 3.3c), pokud vznikne méně je kolimátor ve špatné poloze nebo je snímek podexponován (obrázek 3.3a)). Zároveň vyžadujeme pro robustní funkci jednotky, aby oba vzniklé obrysy měly velikost větší než 15 pixelů. Tyto dva požadavky jsou ohlédány podmínkou na 19 řádce, pokud není splněna pokračuje se na další iteraci. Pokud



je podmínka splněna je vypočtena absolutní hodnotu rozdílu velikostí nalezených obrysů. Nyní vysvětlím nastavení proměnných na začátku této funkce. `Curr_diff` bylo nastaveno na vysokou hodnotu, aby první rozdíl velikostí, byl určitě menší, `curr_size` bylo nastaveno na nulu, aby první velikost obrysů byla určitě větší. Po vypočtení rozdílu velikostí je totiž tato hodnota porovnána, zda je menší nebo rovna aktuálnímu diferenci nejlepší polohy a expozici (pro první iteraci, která splní předešlé podmínky je „nejlepší“ pozice simulována právě hodnotami nastavenými na začátku). Pokud je rovna zjistíme, zda je větší obrys této dvojice větší než velikost aktuálního nejlepšího obrysu, když je podmínka splněna, zjistíme, který z obrysů je větší a uložíme jeho velikost do `curr_size` jejich rozdíl do `curr_diff` a tuto expozici a polohu uložíme jako aktuální nejlepší kombinaci do proměnných `best_position` a `best_exposure`. Pokud vyšlá difference není stejná, zjistíme, zda není menší. Pokud ano, jen zjistíme, který obrys je větší a přepíšeme proměnné stejně jako při rovnosti. Poté co proběhnou všechny iterace obou cyklů nastavíme expozici kamery na zjištěnou nejlepší expozici a otočíme kolimátor na nejlepší zjištěnou pozici. Nakonec zavoláme funkci `EvaluateImage`, která zjišťuje natočení mezi obrysy a skrze nastavení proměnné `ref_found` bude toto natočení uloženo jako referenční.



Obrázek 3.3: Porovnání různých expozičních nastavení při hledání ideální stopy. V prvním sloupci jsou obrázky snímků z kamery, ve druhém jsou snímky po tresholdu a ve třetím jsou snímky po nalezení samostatných objektů a) podexponovaného snímku b) ideálně exponovaného snímku c) přexponovaného snímku.

```

1 int Algorithm::PreFocus() {
2   ref_found = false;
3   curr_diff = 100;
4   curr_size = 0;
5   std::vector<std::vector<cv::Point> > contours;
6   std::vector<cv::Vec4i> hierarchy;
7   cv::Mat tresh;
8   for (int i = 0; i < prefocus_cnt; i++) {
9     //zde bude funkce mikroskopu, které se stara o generovani PWM
10    for (int ii = 1; ii < 4; ii++) {
11      cam.SetExposure(cam.exposure[ii]);
12      if (cam.TakeShot() != 1) {

```

```

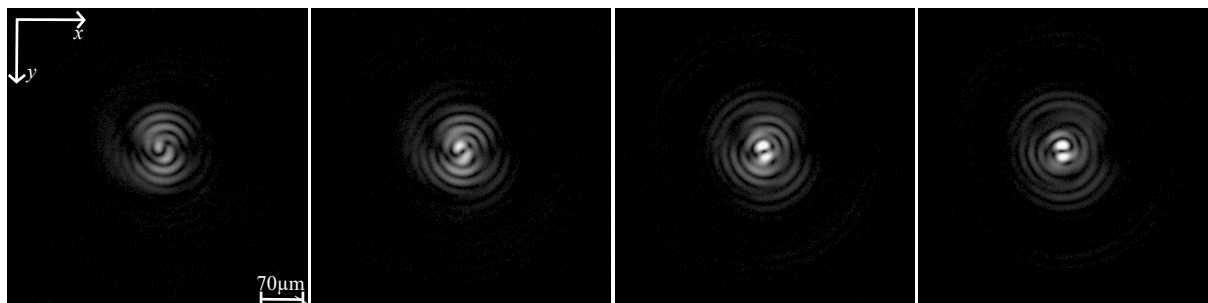
13     return -1;
14 }
15 cv::Mat src(cam.width, cam.height, CV_8UC1, cam.pbuffer, 0);
16 cv::threshold(src, tresh, 220, 255, cv::THRESH_BINARY);
17 // find contours
18 cv::findContours(tresh, contours, hierarchy, cv::RETR_LIST, cv::CHAIN_APPROX_SIMPLE,
cv::Point(0, 0));
19 if (contours.size() == 2 && contours[0].size() > countour_size && contours[1].size()
> countour_size) {
20     diff = contours[0].size() - contours[1].size();
21     diff = abs(diff);
22     if (diff == curr_diff) {
23         if (contours[0].size() > curr_size || contours[1].size() > curr_size) {
24             if (contours[0].size() > contours[1].size()) {
25                 curr_diff = diff;
26                 curr_size = contours[0].size();
27                 best_position = i;
28                 best_exposure = ii;
29             }
30             else {
31                 curr_diff = diff;
32                 curr_size = contours[1].size();
33                 best_position = i;
34                 best_exposure = ii;
35             }
36         }
37     }
38     else if (diff < curr_diff) {
39         if (contours[0].size() > contours[1].size()) {
40             curr_diff = diff;
41             curr_size = contours[0].size();
42             best_position = i;
43             best_exposure = ii;
44         }
45         else {
46             curr_diff = diff;
47             curr_size = contours[1].size();
48             best_position = i;
49             best_exposure = ii;
50         }
51     }
52 }
53 }
54 }
55 cam.exposure[0] = cam.exposure[best_exposure];
56 //zde bude opet funkce mikroskopu na generovani PWM, ktera vrati motor na best_position
57 EvaluateImage();
58 return 1;
59 }

```

Výpis 3.1: Zdrojový kód pro nalezení nejlepší stopy.

Tento algoritmus byl vyvinut na základě toho, že pro nejpřesnější určení natočení algoritmem popsaným v části 4 je důležité, aby byly obrysy co nejvíce podobné skrze to, že pracuje s těžištěm. A na to byl tedy tento algoritmus optimalizován. Pro co nejpřesnější určení ideální polohy a expozice je možné tento algoritmus použít vícekrát, s tím že na základě získaných výsledků se výrazně zmenší rozptyl testovaných expozic a budou se tes-

toovat pouze na okolí získané rotace kolimátoru a získané expozice. Jak vypadají snímky pro různé natočení kolimátoru lze vidět na obrázku 3.4



Obrázek 3.4: Ukázka stop pro různá otočení kolimátoru.



## 4 Algoritmus pro udržení ostrosti

Pro udržování ostrosti bylo potřeba vymyslet metriku pro určení natočení vzniklých obrysů. V našem případě jsme se rozhodly tuto problematiku řešit pomocí těžišť vzniklých obrysů. K těmto dvou bodům přidáme třetí, který bude mít  $y$ -ovou souřadnici jako těžiště obrysu, který vznikl níže a  $x$ -ovou souřadnici jako těžiště obrysu, který vznikl výše. Vznik toho bodu je graficky znázorněn jako bod P na obrázku 4.2. Tím nám vznikne pravoúhlý trojúhelník, ve kterém známe všechny strany, takže není problém dopočítat jakýkoliv úhel. Pro toto řešení jsem vytvořil funkci EvaluateImage, jejichž kód je níže viz výpis 4.1. Začátek je stejný jako u předešlého algoritmu. Dáme příkaz kameře, aby vytvořila snímek, ten si načteme do proměnné typu Mat a aplikujeme na něj threshold. V binárním snímku opět pomocí funkce findContours nalezneme obrysy. I přes to, že by měli být nalezeny pouze dva, skrze předchozí použití funkce PreFocus, je potřeba zkontrolovat podmínkou, zda tomu tak je, jinak by mohlo dojít k pádu celé aplikace. Po této kontrole si vytvoříme vektor o velikosti nalezených obrysů, kde jednotlivé členy budou typu Moments od openCV [32]. Obrazový moment M je vážený průměr intenzit pixelů  $I(x,y)$ , kde  $x$  a  $y$  jsou souřadnice pixelu. Jeho výpočet je pospán vztahem

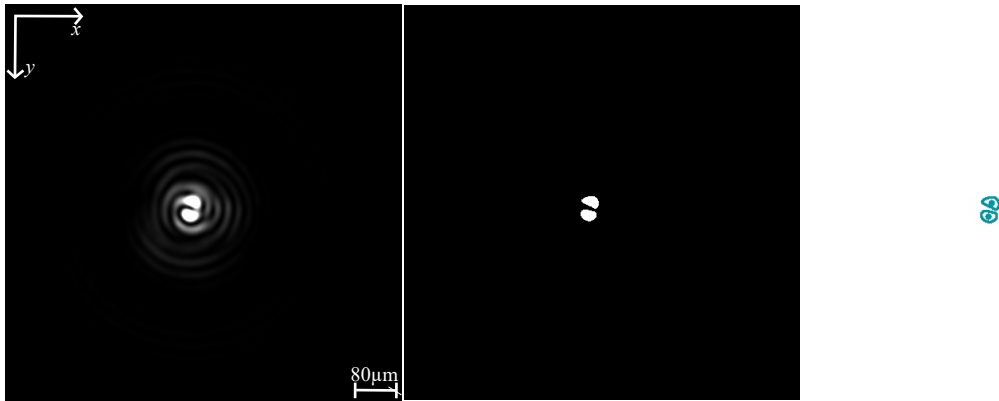
$$M_{i,j} = \sum_x \sum_y x^i y^j I(x,y), \quad (4.1)$$

kde  $i$  a  $j$  jsou řády momentu. OpenCV vypočte momenty nultého až třetího řádu [32], ty mohou být použity pro výpočty různých vlastností zadaných pixelů, v našem případě momenty použijeme pro zjištění těžiště. Pomocí for cyklu vypočteme momenty všech nalezených obrysů. Pro těžiště si vytvoříme vektor o velikosti nalezených obrysů, kde jednotlivé členy mají typ Point2f od OpenCV. Tento typ je bod o dvou souřadnicích, kde souřadnice jsou typu float. Následně for cyklem pomocí momentů vypočteme souřadnice těžiště  $\bar{x}$  a  $\bar{y}$  nalezených obrysů pomocí vztahů 4.2 a 4.3. Ty vychází z toho, že máme binární obrázek, tudíž intenzita je v našem případě vždy 1, a tak nultý řád je počet pixelů a první řád je součet hodnot daných souřadnic. Potup od počátečního snímku přes threshold až po těžiště je graficky znázorněn níže viz obrázek 4.1.

$$\bar{x} = \frac{M_{1,0}}{M_{0,0}}, \quad (4.2)$$

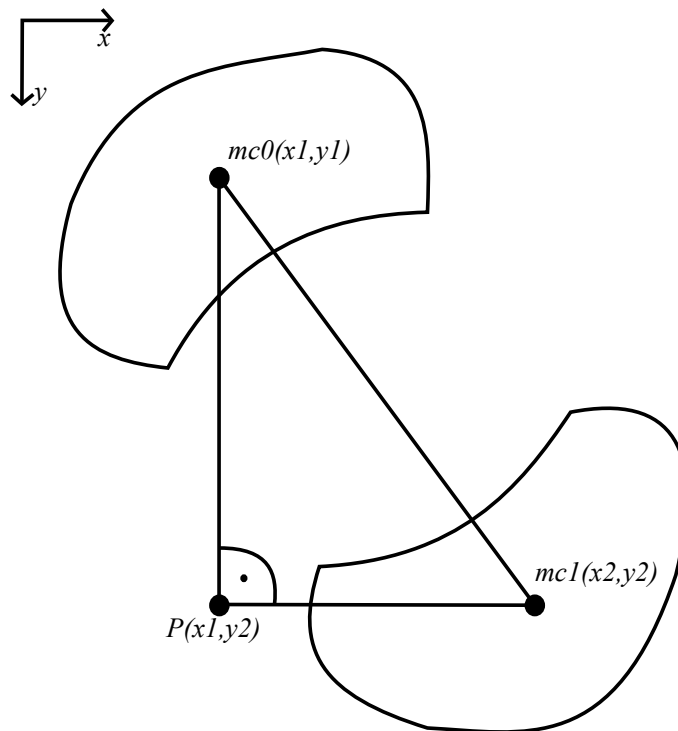
$$\bar{y} = \frac{M_{0,1}}{M_{0,0}}. \quad (4.3)$$

Z těchto vzniklých těžišť obrysů  $mc0$  a  $mc1$  jsme si principem pospaným na začátku této části schopni vytvořit pravoúhlý trojúhelník o vrcholech  $mc0(\bar{x}_1, \bar{y}_1)$ ,  $mc1(\bar{x}_2, \bar{y}_2)$  a  $P(\bar{x}_1, \bar{y}_2)$  viz obrázek 4.2 a v něm si vypočteme délku protilehlé a přilehlé strany pro úhel u těžiště nižšího těžiště v ukázkovém obrázku se jedná o  $mc1$ . Z délek těchto dvou stran pomocí funkce arkus tangens vypočteme úhel u bodu  $mc1$ . Arkus tangens vypočte úhel v radiánech, takže tento úhel převedeme na stupně. Tyto výpočty jsou ve funkci na řádcích 28-30.



Obrázek 4.1: Postup při hledání natočení.

Pro určení, zda se horní obrys nachází v prvním nebo druhém kvadrantu, porovnáme hodnoty  $y$ -ových souřadnic a  $x$ -ových souřadnic těžišť. A podle toho v jakém kvadrantu se horní obrys nachází bude úhel kladný nebo záporný. V našem řešení je druhý kvadrant brán jako záporný a první jako kladný. Na konci funkce je podmínka pro zjištění, zda se jedná o určení referenčního úhlu nebo pouze o určení aktuální polohy a podle toho je úhel po přetypování na char zapsán na patřičné místo v GUI.



Obrázek 4.2: Pravoúhlý trojúhelník vytvořen pomocí těžišť.

```
1 double Algorithm::EvaluateImage() {
2   cntr++;
```

```

3 cv::Mat tresh;
4 if (cam.TakeShot() != 1) {
5     return -1;
6 }
7 cv::Mat src(cam.width, cam.height, CV_8UC1, cam.pbuffer, 0);
8 cv::threshold(src, tresh, 220, 255, cv::THRESH_BINARY);
9 std::vector<std::vector<cv::Point> > contours;
10 std::vector<cv::Vec4i> hierarchy;
11 cv::findContours(tresh, contours, hierarchy, cv::RETR_LIST, cv::CHAIN_APPROX_SIMPLE, cv::
    Point(0, 0));
12 if (contours.size() > 3 || contours.size() < 2) {
13     return -1;
14 }
15 // get the image moments
16 std::vector<cv::Moments> mu(contours.size());
17 for (int i = 0; i < contours.size(); i++)
18 {
19     mu[i] = moments(contours[i], false);
20 }
21 // get the centroid of figures.
22 std::vector<cv::Point2f> mc(contours.size());
23 for (int i = 0; i < contours.size(); i++)
24 {
25     mc[i] = cv::Point2f(mu[i].m10 / mu[i].m00, mu[i].m01 / mu[i].m00);
26 }
27
28 double protilehla = abs(mc[0].y - mc[1].y);
29 double prilehla = abs(mc[0].x - mc[1].x);
30 angle = atan2(protilehla, prilehla) * 180 / 3.14159265;
31 // check if top bulb is in first or seconde quadrant, idf its in second make angle
    negative
32 if (mc[0].y > mc[1].y) {
33     if (mc[0].x > mc[1].x) {
34         angle *= -1;
35     }
36 }
37 else {
38     if (mc[0].x < mc[1].x) {
39         angle *= -1;
40     }
41 }
42 // condition to find if its refernce image
43 if (ref_found == false) {
44     std::string t = std::to_string(angle);
45     char const* n_char = t.c_str();
46     ui.Beast_angle->setText(n_char);
47     ref_found = true;
48 }
49 else {
50     std::string t = std::to_string(angle);
51     char const* n_char = t.c_str();
52     ui.curr_angle_value->setText(n_char);
53     return angle;
54 }
55 }

```

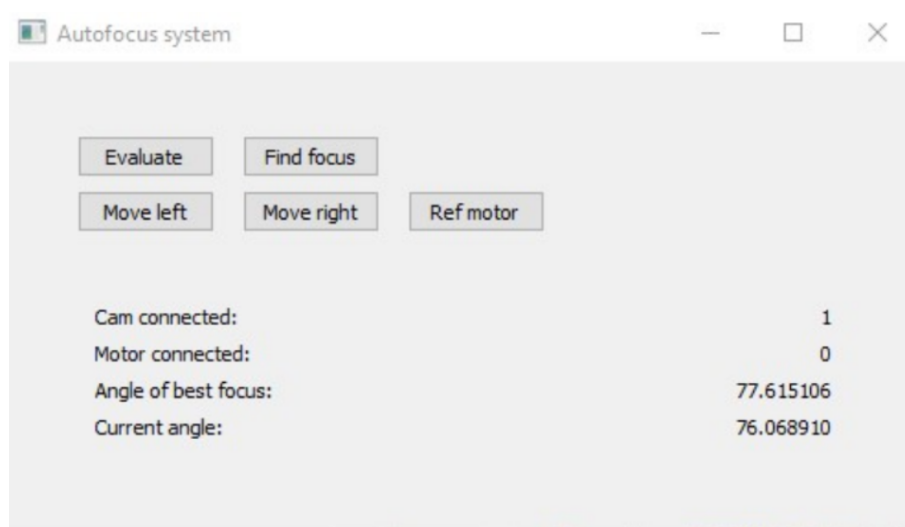
Výpis 4.1: Zdrojový kód pro nalezení nejlepší stopy.





## 5 Grafické rozhraní pro ovládání jednotky

Pro ovládání a testování jednotky jsem vytvořil jednoduché grafické rozhraní jako prototyp pro implementaci do řídicího softwaru CCHM, které je ukázáno na obrázku 5.1. K jeho vytvoření jsem použil framework Qt, pomocí něhož jsem i propojil rozhraní s kódem funkcí. Na první řádku jsou dvě tlačítka Evaluate spustí funkci EvaluateImage a tím pádem vypočítá aktuální natočení. Tlačítko find focus spustí funkci PreFocus, čímž se započne algoritmus pro nalezení ideální polohy. O řádek níže jsou tři tlačítka pro ovládání serva kolimátoru. Ty budou podle jejich názvu propojeny s funkcemi softwarového řešení mikroskopu, které budou generovat PMW pro dosažení daného cíle. Následují 4 informační řádky. Na prvním je popsán status kamery, pokud je 1 program našel kameru a je připraven s ní komunikovat, pokud by bylo 0 program nenalezl kameru a nebude schopen fungovat. Stejný princip je na druhém informačním řádku, kde se akorát vypisuje informace o motoru. Poslední dva řádky ukazují informaci o natočení. První vypisuje, jakou hodnotu má referenční úhel, tedy úhel nalezen pomocí funkce PreFocus. Na druhé je pak vypsán poslední úhel který byl zjištěn pomocí EvaluateImage. Kód, ze kterého je rozhraní generováno, lze dohledat v příloze po názvem Main\_Window.h.



Obrázek 5.1: Grafické rozhraní jednotky.



## 6 Experimenty

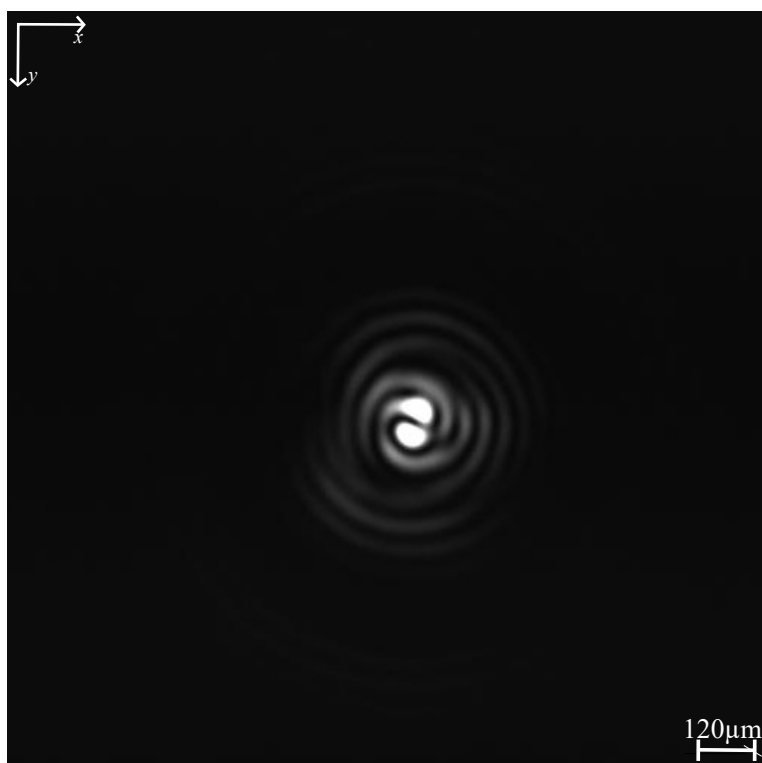
Mnou vyvinuté algoritmy jsem experimentálně ověřil. Nejdříve jsem ověřen algoritmus pro nalezení ideální polohy a poté jsem testoval s jakou přesností funguje algoritmus pro určení natočení v okolí nalezené ideální polohy. Testování probíhalo na experimentální sestavě, kde se posuvy řeší manuálním posuvem. Skrze to mohlo v měření docházet k menší chybovosti, ale pro ověření funkčnosti těchto algoritmů to stačilo.

### 6.1 Ověření funkčnosti nalezení ideální stopy

Kvůli manuálnímu posuvu experimentální sestavy, byla pro testování do kódu algoritmu v hlavním for cyklu místo komentáře přidána funkce od OpenCV `Waitkey(0)`, díky ní se algoritmus zde zastaví a čeká na zmáčknutí jakékoliv klávesy. Takže ještě před spuštěním algoritmu jsem otočil posuv na počáteční polohu, poté jsem spustil algoritmus pomocí GUI. Algoritmus udělal na této poloze tři snímky pro testované expozice a vyhodnotil je. Pak se zastavil a čekal na zmáčknutí klávesy, v tento moment jsem ručně posunul posuv na další pozici a toto jsem opakoval až do poslední pozice. Počáteční polohu jsem našel tak, že jsem ručně našel okolí kde se mi jeví ideální stopa a od ní jsem posunul manuální posuv o 2 mm. Z této pozice jsem poté udělal 9 posuvů s krokem 400 $\mu$ m. Testované expozice byli 1, 3 a 5 ms. Takto bylo vytvořeno a vyhodnoceno 30 snímků. Nejlépe hodnocenou stopu, lze vidět níže viz 6.1. Snímek byl vytvořen s expozicí 3ms. Jak bylo zmíněno již při popisu algoritmu, tento proces by mohl být opakován vícekrát s tím, že při druhé iteraci, by byl zvolen jemnější krok posuvu a menší rozdíly expozic. S tím, že by se testovalo již jen okolí nalezené stopy. Tato testování budou provedena až ve spolupráci s mikroskopem a motorizovaným kolimátorem. Ale již podle nalezené stopy lze hodnotit, že algoritmus by měl být funkční. Navíc pro naše řešení není potřeba najít tu nejlepší stopu, stačí se dostat alespoň do jejího okolí, neboť cílem je dostat se do místa v jehož okolí budeme schopni přesně určovat natočení stopy, ale jestli bude naše referenční stopa ta nejlepší není až tak důležité.

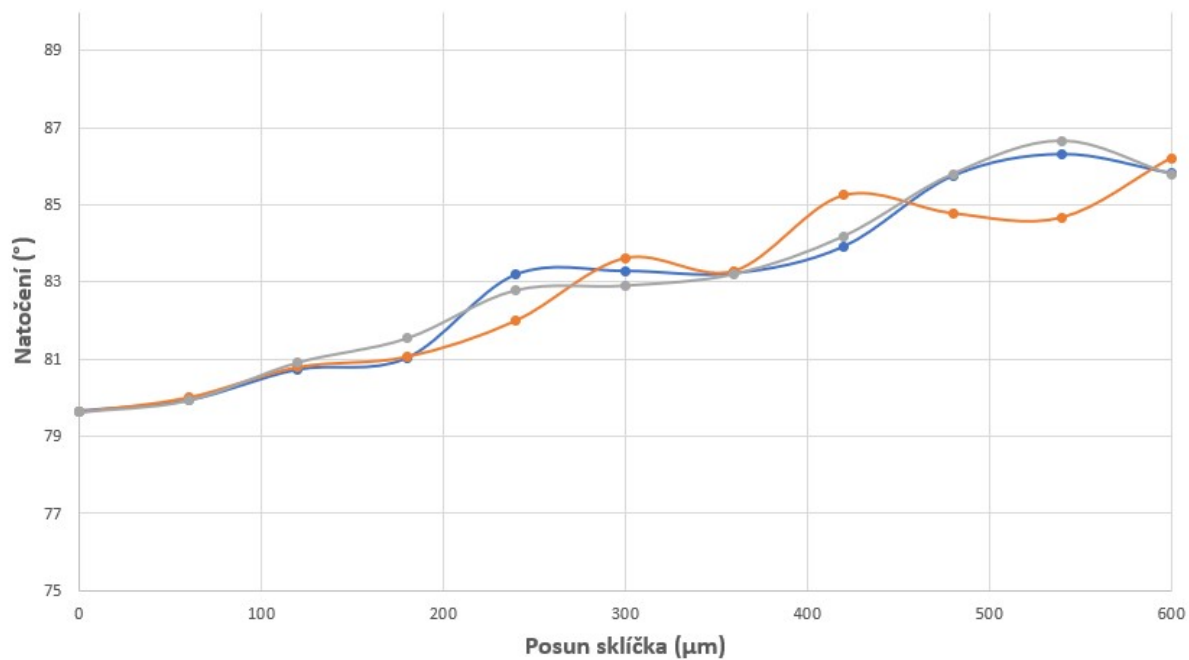
### 6.2 Test přenosti algoritmu pro udržení ostroty

Testování algoritmu pro zjištění natočení, probíhalo tak, že po nalezení ideální stopy předešlým algoritmem jsem začal pomocí posuvu stopu vychylovat. Pro tento test byl zvolen krok 0,003 cm a počet kroků 10. Po změření těchto 11 poloh, jsem se vrátil na počáteční polohu a měření provedl ještě dvakrát. Výsledky těchto třech měření jsou vyneseny do grafu viz obrázek 6.2 Z grafu lze vidět, že měření neměli úplně stejné výsledky, ale na druhou stranu se rozlišnost zvětšuje až s mírou odchýlení od referenční stopy. Pro nás to nejdůležitější je, že referenční stopa byla při všech měření vyhodnocena stejně. Navíc rozladování mikroskopu není rychlý jev a díky tomu je možnost úhel vyhodnotit několikrát a pracovat s průměrem, což by mělo vést k, zlepšení přesnosti.



Obrázek 6.1: Nalezená stopa.

### Závislost natočení na posunu



Obrázek 6.2: Graf závislosti natočení na posunu.

# Závěr

Nejdříve jsem zautomatizoval kolimátor. Toho jsem dosáhl pomocí řemenového mechanismu poháněného servo motorem. Pro tento mechanismus jsem napsal algoritmus, který vyhodnocením snímků najde ideální počáteční nastavení kolimátoru a kameře nastaví délku expozice pro dané nastavení mikroskopu.

Dále jsem napsal algoritmus který vyhodnocuje úhel natočení vzniklé stopy. Úhel určí pomocí těžišť vzniklých obrazců, které jsou vypočteny pomocí obrazového momentu. Zjištění úhlu natočení bude použito pro zjištění natočení referenční pozice a následné udržování naladění mikroskopu pomocí porovnávání aktuálního natočení s natočením referenčním.

Pro autofokusační jednotku jsem vytvořil jednoduché grafické rozhraní, které může spouštět výše zmíněné algoritmy, vypisuje výsledky algoritmu pro určení polohy a informuje o statusu kamery a motoru. Toto rozhraní bude integrované do softwarovým řešením CCHM.

Oba algoritmy jsem experimentálně ověřil pomocí laboratorní sestavy. Algoritmus pro nalezení ideální polohy a expozice dokázal najít polohu, kterou bych vybral i já a expozici která vedla k výrazným rozdílům mezi důležitou informací a šumem, což bylo cílem. Do budoucna by mohl být algoritmus spuštěn ve více iteracích, kde se při dalších iteracích sníží rozmezí testovaných poloh a expozic na základě výsledků předešlé iterace. Tím by mohlo být dosaženo nalezení optimálních výsledků. Algoritmus pro určení polohy byl testován v okolí ideální polohy. Lze vidět, že při oddalování od ideální polohy se algoritmus pro různá měření lišil, a to až v rozmezí  $1,5^\circ$ . To mohlo být způsobeno statistickou chybou při ovládní manuálního posuvu, kdy mohlo být natočení vyhodnocováno pro mírně rozlišné posuny. Pro funkčnost jednotky je dostačující, aby algoritmus poznal přesně referenční polohu a při rozladění určil na jakou stranu se stopa odchyluje, a to algoritmus splňuje.

# Literatura

- [1] DAVIES, H. G., WILKINS, M. H., CHAYEN, J. a COUR, L. The Use of the Interference Microscope to Determine Dry Mass in Living Cells and as a Quantitative Cytochemical Method. Dostupné z: <https://www.semanticscholar.org/paper/The-Use-of-the-Interference-Microscope-to-Determine-Davies-Wilkins/258cb739776e8f88c5aed0aa36f6521b596f21c6>.
- [2] 2.1. *Light Microscopy*. Dostupné z: <https://sisu.ut.ee/heritage-analysis/21-light-microscopy>.
- [3] ZERNIKE, F. Phase Contrast, a New Method for the Microscopic Observation of Transparent Objects Part II. *Physica*. 1942, sv. 9, s. 974,IN1,981,IN3,983–980,IN1,982,IN3,986. DOI: 10.1016/S0031-8914(42)80079-8. Dostupné z: <https://ui.adsabs.harvard.edu/abs/1942Phy.....9..974Z>.
- [4] ZERNIKE, F. Phase Contrast, a New Method for the Microscopic Observation of Transparent Objects. *Physica*. 1942, sv. 9, č. 7, s. 686–698. DOI: 10.1016/S0031-8914(42)80035-X. ISSN 00318914. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S003189144280035X>.
- [5] GEORGES, N. *Interferential Polarizing Device for Study of Phase Objects*. patentus 2924142A. 1960. Dostupné z: <https://patents.google.com/patent/US2924142A/en>.
- [6] HOFFMAN, R. a GROSS, L. The Modulation Contrast Microscope. *Nature*. 1975, sv. 254, č. 5501, s. 586–588. DOI: 10.1038/254586a0. ISSN 0028-0836.
- [7] WEBB, K., FILIPPI, G. a JOHNSTON, N. S. Condenser-Free Zernike Phase Contrast Imaging for Scanning Probe Microscopy. In: Dostupné z: <https://www.semanticscholar.org/paper/Condenser-free-Zernike-phase-contrast-imaging-for-Webb-Filippi/d34dde4a6021aeced2ae832e9a701edc87cd76>.
- [8] SHERIDAN, J. T., KOSTUK, R. K., GIL, A. F., WANG, Y., LU, W. et al. Roadmap on Holography. *Journal of Optics*. IOP Publishing. 2020, sv. 22, č. 12, s. 123002. DOI: 10.1088/2040-8986/abb3a4. ISSN 2040-8986. Dostupné z: <https://dx.doi.org/10.1088/2040-8986/abb3a4>.
- [9] KOLMAN, P. a CHMELÍK, R. Coherence-Controlled Holographic Microscope. *Optics Express*. 2010, sv. 18, č. 21, s. 21990. DOI: 10.1364/OE.18.021990. ISSN 1094-4087. Dostupné z: <https://opg.optica.org/oe/abstract.cfm?uri=oe-18-21-21990>.
- [10] KVASNICA, L. *Digitální Metody Zpracování Trojrozměrného Zobrazení v Rentgenové Tomografii a Holografické Mikroskopii; Ing. Lukáš Kvasnica, Ph.D. ( 2015 - 89406) – VUT*. 2015. Dostupné z: [https://www.vut.cz/studenti/zav-prace?zp\\_id=89406](https://www.vut.cz/studenti/zav-prace?zp_id=89406).

- [11] ZICHA, D. a CHMELIK, R. Testing Anti-Cancer Drugs with Holographic Incoherent-Light-Source Quantitative Phase Imaging. *Methods in Enzymology*. 2023, sv. 679, s. 255–274. DOI: 10.1016/bs.mie.2022.08.017. ISSN 1557-7988.
- [12] KRIZOVA, A., COLLAKOVA, J., DOSTAL, Z., KVASNICA, L., UHLIROVA, H. et al. Dynamic Phase Differences Based on Quantitative Phase Imaging for the Objective Evaluation of Cell Behavior. *Journal of Biomedical Optics*. 2015, sv. 20, č. 11, s. 111214. DOI: 10.1117/1.JBO.20.11.111214. ISSN 1560-2281.
- [13] *Teligt Q-Phase • Teligt*. Dostupné z: <https://teligt.eu/products/teligt-q-phase/>.
- [14] SHAH, M., MISHRA, S., SARKAR, M. a ROUT, C. Identification of Robust Focus Measure Functions for the Automated Capturing of Focused Images from Ziehl–Neelsen Stained Sputum Smear Microscopy Slide. *Cytometry Part A*. 2017, sv. 91, č. 8, s. 800–809. DOI: 10.1002/cyto.a.23142. ISSN 1552-4930. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.23142>.
- [15] BRENNER, J. F., DEW, B. S., HORTON, J. B., KING, T., NEURATH, P. W. et al. An Automated Microscope for Cytologic Research a Preliminary Evaluation. *The Journal of Histochemistry and Cytochemistry: Official Journal of the Histochemistry Society*. 1976, sv. 24, č. 1, s. 100–111. DOI: 10.1177/24.1.1254907. ISSN 0022-1554.
- [16] MASON, D. C. a GREEN, D. K. Automatic Focusing of a Computer-Controlled Microscope. *IEEE transactions on bio-medical engineering*. 1975, sv. 22, č. 4, s. 312–317. DOI: 10.1109/tbme.1975.324450. ISSN 0018-9294.
- [17] KEHTARNAVAZ, N. a OH, H. J. Development and Real-Time Implementation of a Rule-Based Auto-Focus Algorithm. *Real-Time Imaging*. 2003, sv. 9, č. 3, s. 197–203. DOI: 10.1016/S1077-2014(03)00037-8. ISSN 1077-2014. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1077201403000378>.
- [18] ZIOU, D. a DESCHENES, F. Depth from Defocus Estimation in Spatial Domain. *Computer Vision and Image Understanding*. 2001, sv. 81, č. 2, s. 143–165. DOI: 10.1006/cviu.2000.0899. ISSN 1077-3142. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1077314200908993>.
- [19] REN, Z., LAM, E. Y. a ZHAO, J. Acceleration of Autofocusing with Improved Edge Extraction Using Structure Tensor and Schatten Norm. *Optica Publishing Group*. Optica Publishing Group. 2020, sv. 28, č. 10, s. 14712–14728. DOI: 10.1364/OE.392544. ISSN 1094-4087. Dostupné z: <https://opg.optica.org/oe/abstract.cfm?uri=oe-28-10-14712>.
- [20] DIMEO, P., SUN, L. a DU, X. Fast and Accurate Autofocus Control Using Gaussian Standard Deviation and Gradient-Based Binning. *Optics Express*. 2021, sv. 29, č. 13, s. 19862–19878. DOI: 10.1364/OE.425118. ISSN 1094-4087. Dostupné z: <https://opg.optica.org/oe/abstract.cfm?uri=oe-29-13-19862>.
- [21] LIAO, J., CHEN, X., DING, G., DONG, P., YE, H. et al. Deep Learning-Based Single-Shot Autofocus Method for Digital Microscopy. *Biomedical Optics Express*. 2022, sv. 13, č. 1, s. 314–327. DOI: 10.1364/BOE.446928. ISSN 2156-7085. Dostupné z: <https://opg.optica.org/boe/abstract.cfm?uri=boe-13-1-314>.



- [22] BIAN, Z., GUO, C., JIANG, S., ZHU, J., WANG, R. et al. Autofocusing Technologies for Whole Slide Imaging and Automated Microscopy. *Journal of Biophotonics*. 2020, sv. 13, č. 12, s. e202000227. DOI: 10.1002/jbio.202000227. ISSN 1864-0648. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jbio.202000227>.
- [23] LIRON, Y., PARAN, Y., ZATORSKY, N. G., GEIGER, B. a KAM, Z. Laser Autofocusing System for High-Resolution Cell Biological Imaging. *Journal of Microscopy*. 2006, sv. 221, s. 145–151. DOI: 10.1111/j.1365-2818.2006.01550.x. ISSN 0022-2720.
- [24] REINHEIMER, G. *Arrangement for Automatically Focussing an Optical Instrument*. patentus 3721827A. 1973. Dostupné z: <https://patents.google.com/patent/US3721827A/en>.
- [25] *Nikon Perfect Focus System (PFS)*. Dostupné z: <https://www.microscopyu.com/applications/live-cell-imaging/nikon-perfect-focus-system>.
- [26] BARÁNEK, M., BOUCHAL, P., ŠILER, M. a BOUCHAL, Z. Aberration Resistant Axial Localization Using a Self-Imaging of Vortices. *Optics Express*. Optica Publishing Group. 2015, sv. 23, č. 12, s. 15316–15331. DOI: 10.1364/OE.23.015316. ISSN 1094-4087. Dostupné z: <https://opg.optica.org/oe/abstract.cfm?uri=oe-23-12-15316>.
- [27] BOUCHAL, P. a BOUCHAL, Z. Flexible Non-Diffractive Vortex Microscope for Three-Dimensional Depth-Enhanced Super-Localization of Dielectric, Metal and Fluorescent Nanoparticles. *Journal of Optics*. IOP Publishing. 2017, sv. 19, č. 10, s. 105606. DOI: 10.1088/2040-8986/aa87fb. ISSN 2040-8986. Dostupné z: <https://dx.doi.org/10.1088/2040-8986/aa87fb>.
- [28] *Systém pro Udržování Ostrosti Obrazu Světelného Mikroskopu; Bc. Jakub Dokulil (2021 - 140039) – VUT*. 2022. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/140039>.
- [29] *DSS-M15S\_270°\_15KG\_DF\_Metal\_Servo\_with\_Analog\_Feedback\_SKU\_\_SER0044-DFRobot*. Dostupné z: [https://wiki.dfrobot.com/DSS-M15S\\_270%C2%B0\\_15KG\\_DF\\_Metal\\_Servo\\_with\\_Analog\\_Feedback\\_SKU\\_\\_SER0044](https://wiki.dfrobot.com/DSS-M15S_270%C2%B0_15KG_DF_Metal_Servo_with_Analog_Feedback_SKU__SER0044).
- [30] *Continental Industry – Our Products & Solutions*. Dostupné z: <http://www.continental-industry.com/en/solutions/power-transmission/industrial-applications/drive-belts>.
- [31] *OpenCV: Contours : Getting Started*. Dostupné z: [https://docs.opencv.org/3.4/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html).
- [32] *OpenCV: Cv::Moments Class Reference*. Dostupné z: [https://docs.opencv.org/4.x/d8/d23/classcv\\_1\\_1Moments.html](https://docs.opencv.org/4.x/d8/d23/classcv_1_1Moments.html).