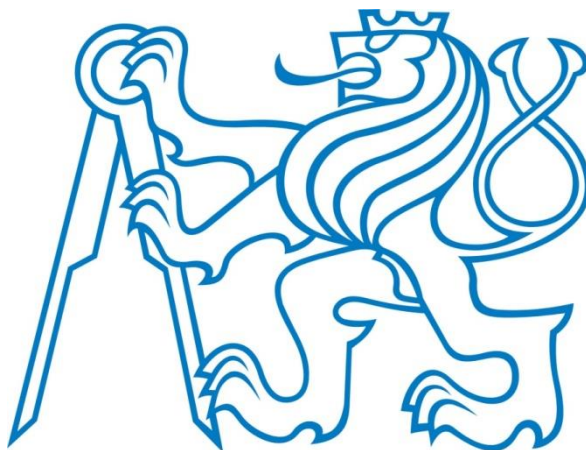


# České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra měření



## Diplomová práce

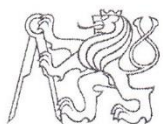
Průběžné zpracování obrazu v DSP pro  
sledování objektů

Autor: Jiří Hladík

Vedoucí: doc. Ing. Jan Fischer, CSc.

Obor: Kybernetika a robotika

Praha Květen 2015



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	<b>Bc. Jiří Hladík</b>
Studijní program:	<b>Kybernetika a robotika</b>
Obor:	<b>Senzory a přístrojová technika</b>
Název tématu česky:	<b>Průběžné zpracování obrazu v DSP pro sledování objektů</b>
Název tématu anglicky:	<b>DSP Based Image Processing on the Fly for Object Tracking</b>

### Pokyny pro vypracování:

Navrhněte a do kamery se signálovým procesorem ADSP-BF533 implementujte algoritmy pro vyhodnocení a sledování stínových obrazů a polohy obrazů jasových značek. Zpracováním obrazu se má určit též plocha, tvar, poloha středu výrazných objektů ve snímané scéně. Orientujte se na rychlé metody průběžného zpracování obrazu, které bude možno realizovat pouze s využitím malé vnitřní paměti signálového procesoru bez potřeby velké snímkové paměti. Demonstrujte funkčnost metody též na případu snímání, vyhodnocení a elektromechanické regulaci polohy optické značky.

### Seznam odborné literatury:

- [1] Analog Devices: ADSP-BF533 Blackfin® Processor Hardware Reference, [www.analog.com](http://www.analog.com)
- [2] Michálek, A.: Bakalářská práce, ČVUT - FEL, Praha
- [3] Heijden, F.: Image Based Measurement Systems: Object Recognition and Parameter Estimation, ISBN 10: 0471950629, Wiley, 1995

Vedoucí diplomové práce: doc. Ing. Jan Fischer, CSc.

Datum zadání diplomové práce: 25. listopadu 2013

Platnost zadání do<sup>1</sup>: 31. srpna 2015

Prof. Ing. Vladimír Haasz, CSc.  
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 25. 11. 2013

<sup>1</sup> Platnost zadání je omezena na dobu tří následujících semestrů.

## Čestné prohlášení

Prohlašuji, že jsem předloženou diplomovou prací vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Tato práce vznikla v laboratoři videometrie, katedry měření ČVUT – FEL v Praze pod vedením doc. Ing. Jana Fischera, CSc.

V Praze dne .....

.....

Podpis autora

## **Poděkování**

Rád bych na tomto místě poděkoval doc. Ing. Janu Fischerovi, CSc. za odborné vedení práce, velmi cenné rady, náměty a vstřícnost během celé doby spolupráce. Také děkuji své rodině a přátelům za neocenitelnou podporu během studia.

## Abstrakt

Tato práce se zabývá metodami zpracování obrazu za účelem parametrizace objektů v něm. Uvažované obrazy jsou rozděleny na stínové obrazy, u nichž je určována plocha, tvar, souřadnice těžiště a další parametry, a na obrazy jasových značek, kde je určován jejich střed s přesností lepší než 0,1 pixelu. Vysoké přesnosti je dosaženo aproximací jasových značek pomocí Gaussovy křivky dvou proměnných. Všechny algoritmy navržené v práci jsou též metodami rychlého průběžného zpracování implementovány do signálového procesoru ADSP-BF533 a optimalizovány s ohledem na rychlost zpracování.

V práci je též navržena metoda polohování využívající určení polohy jasové značky na CMOS senzoru. Tato metoda je ověřena v mikropolohovacím systému, který je schopen určit polohu v rozsahu několika milimetrů s přesností na 500 nm v rovině senzoru a 6  $\mu\text{m}$  ve směru normály.

## Abstract

This thesis deals with methods of image processing for the purpose of parameterization of objects in image. There are considered two types of images. First of it are shadowed images, where are determined area, shape and coordinates of centroid of the objects in image. Second type are images with bright signs, where the position of sign is found with accuracy better than 0,1 pixel. High accuracy is achieved by approximation of bright signs by Gaussian curve with two variables. All of the algorithms are implemented into digital signal processor ADSP-BF533 using fast on-the-fly methods and optimized for the speed of processing.

There is designed method for positioning, utilizing determination of position of bright sign on CMOS sensor. This method is verified in micro positioning system, which is able to determine position in range of couple millimeters with accuracy 500 nm in plane of sensor and 6  $\mu\text{m}$  in the direction of the normal to sensor.

# Obsah

<b>1</b>	<b>ÚVOD.....</b>	<b>1</b>
1.1	Současný stav a cíle práce .....	2
<b>2</b>	<b>TEORETICKÝ ROZBOR.....</b>	<b>3</b>
2.1	Parametrizace stínových obrazů.....	3
2.1.1	Konvoluce pro zpracování obrazu.....	4
2.1.2	Hranová detekce.....	5
2.1.3	Vytvoření binárního obrazu.....	6
2.1.4	Segmentace obrazu.....	6
2.1.5	Labeling .....	7
2.1.6	Získané parametry objektů .....	8
2.2	Určení polohy jasových značek.....	10
2.2.1	Difrakční jevy na štěrbině.....	10
2.2.2	Aproximace difrakčních obrazců .....	14
<b>3</b>	<b>NÁVRH POLOHOVACÍHO SYSTÉMU A JEHO SOUČÁSTÍ.....</b>	<b>20</b>
3.1	Procesorový a obrazový modul .....	21
3.2	Pojezdy pro polohování .....	22
3.3	Řízení laserů.....	23
3.4	Mechanická konstrukce.....	24
<b>4</b>	<b>HARDWAROVÉ ŘEŠENÍ.....</b>	<b>25</b>
4.1	Vytvoření mechanické konstrukce .....	25
4.2	Nepřesnosti mechanické konstrukce .....	26
4.3	Geometrie soustavy .....	27
4.4	Návrh budiče laserů .....	28
<b>5</b>	<b>IMPLEMENTACE ALGORITMŮ DO DSP .....</b>	<b>30</b>
5.1	Zpracování stínových obrazů.....	30
5.1.1	Rozložení paměti.....	30
5.1.2	Detekce hran.....	31
5.1.3	Získání binárního obrazu .....	31
5.1.4	Labeling .....	32
5.2	Zpracování jasových značek.....	33
5.2.1	Expoziční doba a nastavení prahování.....	33

5.2.2	Labeling a zpracování objektů.....	34
5.2.3	Aproximace obrazce a nalezení středu.....	35
5.2.4	Rychlost implementovaných algoritmů.....	36
<b>6</b>	<b>ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ.....</b>	<b>38</b>
6.1	Měření homogenity osvětlení.....	38
6.2	Měření opakovatelnosti.....	39
6.3	Odchylka od linearity.....	41
6.4	Měření ve směru normály.....	43
6.5	Návrhy na další vývoj.....	44
<b>7</b>	<b>ZÁVĚR .....</b>	<b>45</b>
<b>8</b>	<b>SEZNAM POUŽITÝCH ZKRATEK .....</b>	<b>47</b>
<b>9</b>	<b>SEZNAM POUŽITÝCH SYMBOLŮ .....</b>	<b>48</b>
<b>10</b>	<b>SEZNAM OBRÁZKŮ A TABULEK .....</b>	<b>50</b>
<b>11</b>	<b>BIBLIOGRAFIE .....</b>	<b>52</b>
	<b>PŘÍLOHA A: MĚŘENÍ DIFRAKČNÍCH OBRAZCŮ.....</b>	<b>54</b>
	<b>PŘÍLOHA B: NÁVRH DPS BUDIČE LASERŮ .....</b>	<b>55</b>
	<b>PŘÍLOHA C: FOTODOKUMENTACE .....</b>	<b>60</b>
	<b>PŘÍLOHA D: OBSAH PŘILOŽENÉHO CD .....</b>	<b>63</b>

# 1 Úvod

Moderní technologie okolo nás vyžadují nové metody, výkonnější a menší elektroniku a chytřejší algoritmy. Stejně tak se technologie vyvíjí i v oblasti bezkontaktní měřicí techniky a pro nás zajímavé oblasti videometrie. Zatím ani nejlepší smart kamery nebo celé počítačové sítě nejsou schopny vnímat sémantiku komplexních obrazů každodenního světa, jež je schopen vnímat člověk, nicméně díky vývoji je i v oblasti videometrie možné zkoumat složitější obrazy než dříve. Zejména v průmyslových aplikacích je možné pomocí jednoduchých vestavěných systémů rozpoznávat objekty, parametrizovat je nebo přesně určovat jejich polohu.

Malé vestavné kamerové systémy nejčastěji rozpoznávají obrazy, které by se daly rozdělit na dva typy. Jsou to stínové obrazy s jasnými hranami objektů a jasové značky s malou velikostí a velkým jasem. Stínové obrazy reprezentující objekt, jehož vlastnosti jako velikost, tvar atd. chceme znát, vznikají většinou osvětlením objektů na kontrastním pozadí nebo zadním nasvícením. Jasové značky na druhou stranu reprezentují spíše ukazatel na místo, které je něčím zajímavé, a jehož polohu chceme znát. Vznikají například posvícením laserovým ukazovátkem.

Kvůli zrychlujícím se procesům výroby je u obou výše uvedených typů obrazů kladen nárok na co nejvyšší rychlost zpracování. Mnoho kamerových systému proto zpracovává obraz takzvaně on-the-fly, kdy je obraz zpracováván v průběhu ukládání dat do paměti. Využitá paměť pro zpracování obrazu je tedy menší než obraz sám a výsledek zpracování je znám krátce poté, co je přenesena poslední část obrazu.

Motivační úlohou pro tuto diplomovou práci je rychlé a přesné určení polohy systému na základě znalosti polohy jasové značky. Velmi přesná znalost polohy se dá využít například při měření teplotních roztažností, zjištění mechanických deformací, monitorování nízkofrekvenčních vibrací nebo třeba pro určení polohy preparátu v mikroskopu.



## 1.1 Současný stav a cíle práce

V současné době se k určení polohy ve 2D používá několik typů senzorů. Nejjednoduššími z nich jsou více kvadrantové fotodiody nebo PSD senzory. Jejich přesnost je však omezená přesností použitých AD převodníků a měří na principu těžiště světelné stopy. Dalším způsobem měření může být například použití dvou na sebe kolmých inkrementálních snímačů. Výhodou tohoto systému je velký rozsah měřitelných vzdáleností a přesnost měření, ale ne vždy je zaručena kolmost souřadnic a bývají velmi drahé. Jednou z levnějších metod je určení polohy pomocí objektivu a CCD či CMOS senzoru, avšak použité objektivy vždy mají optické vady.

Metoda zkoumaná v této práci je nejvíce podobná poslední zmíněné možnosti. Objektiv způsobující chyby lze úplně odstranit a promítat úzkou laserovou jasovou značku přímo na senzor. Tím vznikne systém, kde poloha přímo úměrně odpovídá poloze jasové značky na senzoru. Hlavním cílem práce tedy je prozkoumat možnosti této metody přesného polohování a vytvořit autonomní měřicí systém, který jí bude využívat. Pro získání konkrétní představy je tento hlavní cíl rozdělen do několika dílčích kroků.

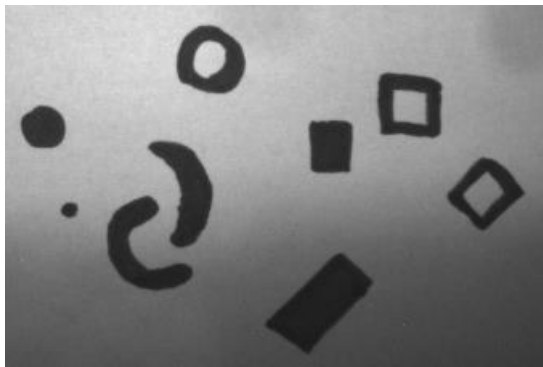
- Analýza metod zpracování stínových obrazů a jasových značek použitelných pro danou aplikaci.
- Provedení teoretického rozboru obrazce vzniklého za štěrbínou.
- Implementace metod zpracování obrazu a algoritmů pro nalezení středu jasové značky a parametrizaci stínových obrazů do signálového procesoru ADSP-BF533 Blackfin.
- Návrh a realizace mechanické a elektrické části experimentálního systému pro mikropolohování, který ověří možnosti metody.
- Vyhodnocení dosažené přesnosti mikropolohování, diskuze vzniklých chyb a návrh na další postup práce.

## 2 Teoretický rozbor

V této kapitole budou rozebrány různé metody zpracování obrazu jak pro parametrizaci stínových obrazů, tak pro určení polohy jasových značek. Metody budou uvedeny postupně v pořadí, tak jak mohou být aplikovány na obraz a k některým bude uveden potřebný teoretický základ, případně bude diskutována vhodnost implementace do signálového procesoru.

### 2.1 Parametrizace stínových obrazů

Jak již bylo naznačeno dříve, obraz se stínovými obrazy objektů může vzniknout buď nasvícením objektů na pozadí s rozdílným jasnem, nebo pomocí zadního osvětlení. Takto vzniklý obraz s hranově významnými objekty může vypadat například tak jako na obr. 2.1



Obr. 2.1 – Příklad obrazu se stínovými objekty

Na první pohled vidíme, že jasové hodnoty některých objektů jsou nižší než jas pozadí ve spodní části obrazu. Proto nestačí obraz zkoumat pouze na základě jasové informace jednotlivých objektů neboli pomocí prahování [1], ale je nutné použít pokročilejší metody. Velmi mocným nástrojem pro zpracování obrazu je konvoluce, proto je jí zde věnována krátká kapitola.

### 2.1.1 Konvoluce pro zpracování obrazu

Pomocí konvoluce lze provádět nejrůznější předzpracování signálů jako filtraci šumu, detekci hran, frekvenční filtry atd. Podrobně je konvoluce popsána například v [2]. My se omezíme pouze na základ důležitý pro zpracování obrazu. Konvoluce pro signály ve spojitém čase je definována vztahem:

$$g(t) = (h * f)(t) = \int_{-\infty}^{\infty} h(\tau)f(t - \tau)d\tau \quad (1)$$

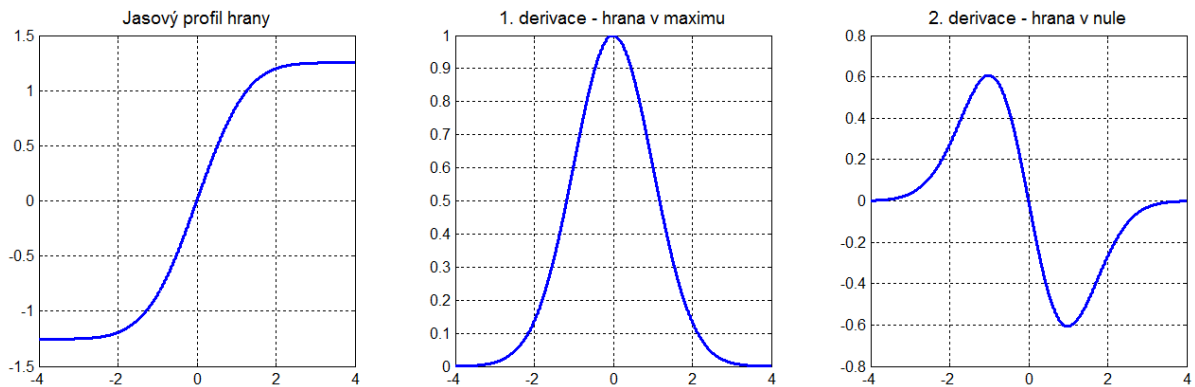
Kde  $h(t)$  a  $f(t)$  jsou signály, jejichž konvoluci  $g(t)$  chceme znát. Přejdeme-li do diskretní oblasti a zavedeme-li druhou integrační proměnnou, kterou potřebujeme pro přechod od 1D signálů k 2D signálům, jež vyjadřují obrazovou funkci, dostaneme konvoluci ve tvaru:

$$g(m, n) = (h * f)(m, n) = \sum_{i=-N}^N \sum_{j=-M}^M h(i, j)f(m - i, n - j) \quad (2)$$

Kde  $f(m, n)$  je obrazová funkce,  $h(m, n)$  je označována jako jádro konvoluce nebo maska o velikosti  $h(2N + 1, 2M + 1)$  a funkce  $g(m, n)$  je obraz s aplikovanou konvoluční maskou. Právě tato maska  $h(m, n)$  nám určuje, jakou operaci s obrazem provádíme. Například maska aproximující Gaussovu funkci provádí filtraci obrazu. Maska s derivací Gaussovy funkce v určitém směru provádí detekci hran v tom daném směru. Avšak existují i další masky například pro rozmazání nebo naopak doostření obrazu. Pro zpracování stínových obrazů se hodí maska provádějící hranovou detekci.

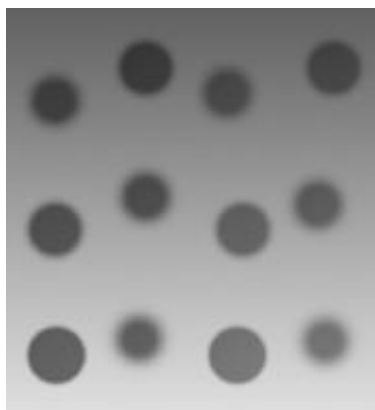
## 2.1.2 Hranová detekce

Pomocí různých konvolučních masek (Prewitt, Sobel atd.) [3] lze hledat maximum první derivace nebo průchod druhé derivace nulou například pomocí tzv. LoG (Laplacian of Gaussian) [3] operátoru. Výstup z obou metod hledajících hrany je vidět na obrázku obr. 2.2

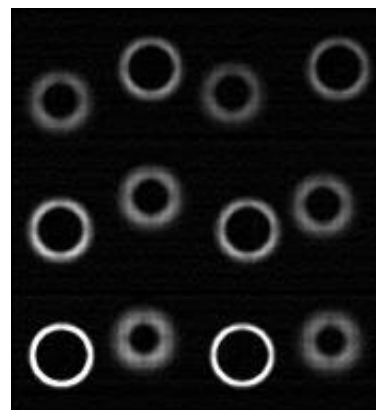


Obr. 2.2 – Zobrazení hran různými metodami

Pro implementaci do procesoru není vhodné použít detekci založenou na druhé derivaci, protože hledání průchodů obrazové funkce nulou je netriviální operace na rozdíl od prahování, které lze s výhodou použít u detekce pomocí první derivace. Pro detekci hran ve všech směrech je nutné aplikovat konvoluční masku vícekrát: 2x nebo 4x pro různé směry. Jelikož pro další zpracování není nutné znát směr hrany, lze použít absolutní hodnotu. Díky tomu získáme obraz, ve kterém jsou všechny hranově významné objekty ohraničené linkou. Viz obr. 2.4 a obr. 2.3



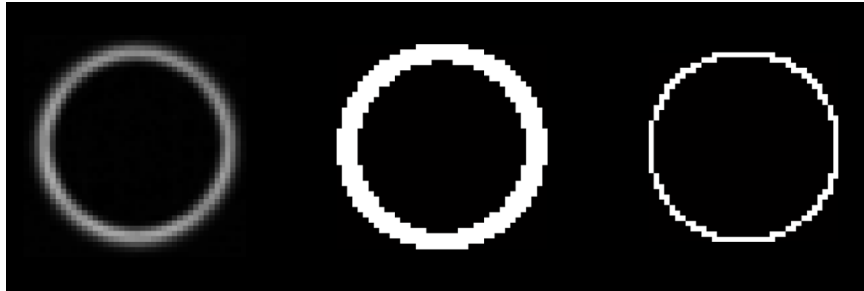
Obr. 2.4 – Vstupní data



Obr. 2.3 – Detekované hrany

### 2.1.3 Vytvoření binárního obrazu

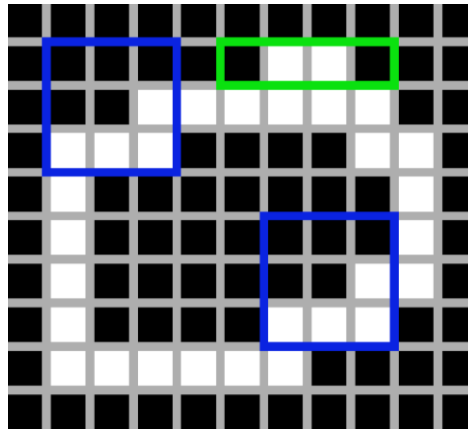
Pro určení, které pixely jsou hranové a které ne, je nejjednodušší použít metodu prahování. Tím je ovšem hrana velmi široká a tedy špatně lokalizovaná. Pro lepší určení polohy hrany je vhodné použít metodu non-maxima-suppression [4], která kromě prahování hledá lokální maximum hrany a to ve 4 směrech. Výsledek nalezení hranových pixelů je vidět na obrázku obr. 2.5



Obr. 2.5 – vlevo: vstup, uprostřed: prahování hran, vpravo: nalezení maxima

### 2.1.4 Segmentace obrazu

Díky výše provedeným operacím vznikl obraz s dobře lokalizovanými hranami jednotlivých objektů, avšak ztratila se informace o tom, které pixely patří objektu a které pozadí. K jednoznačnému určení objektů od pozadí existují algoritmy segmentace hledající ohraničené objekty. V tomto případě je však nelze použít, protože nemáme uložený celý obraz, ale pouze pár řádků. Obecně lze říci, že z 8-okolí pixelu nejsme schopni rozpoznat, zdali se jedná o objekt, nebo o pozadí jak je patrné na obr. 2.6, kde modré případy jsou z hlediska 8-okolí identické, ale v jednom případě jde o objekt a v druhém ne. Nepomůže ani zapamatování, jestli předtím nastala hrana či nikoliv viz obr. 2.6 případ zelený, kdy nezačíná objekt, ale je zde hrana.

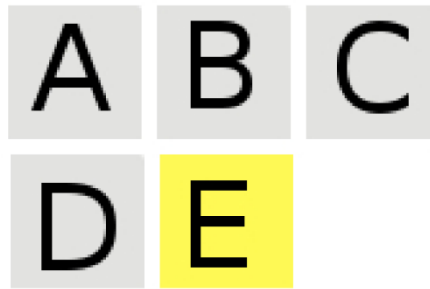


Obr. 2.6 – Znázornění problémových situací při segmentaci

K jednoznačnému určení pozadí a objektů by tedy bylo nutné mít v paměti uložený celý obrázek, což v daném případě velikost paměti použitého procesoru neumožňuje. Segmentaci obrazu na jednotlivé objekty však není potřeba provádět a lze za objekt prohlásit vše, co není hranový pixel, tedy i pozadí a případné vnitřky objektů s dírou. Následnému zpracování pomocí labelingu tento fakt nijak nevádí. Naopak je díky tomu získáno více informací, jako třeba plocha pozadí, zdali objekt má díru nebo ne atd.

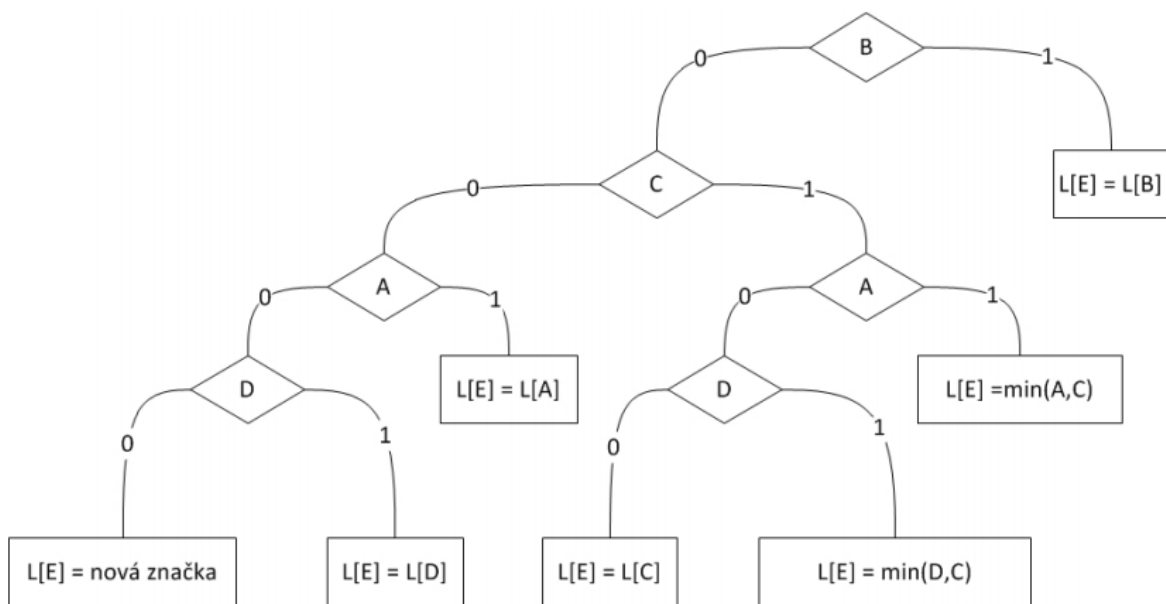
### 2.1.5 Labeling

Metoda Connected-component-labeling je hojně využívána při zpracování obrazu a slouží k označování objektů. Základ pro tuto metodu vychází z teorie grafů, kde si lze každý objekt představit jako souvislou komponentu grafu. Labeling pracuje s binárním obrazem, kde jsou všechny objekty odlišeny od pozadí, ale mají stejnou hodnotu. Po provedení labelingu jsou tyto objekty označovány unikátní značkou. Algoritmus byl implementován v práci [5], proto jsou zde popsány jen jeho základy. Základem pro nalezení sousednosti objektů při průběžném zpracování obrazu je prohledávání asymetrického 4-okolí. Viz obr. 2.7. Zpracovávanému pixelu E je přiřazena minimální ze značek okolních pixelů A-D nebo nová značka jde-li o nový objekt.



Obr. 2.7 – Prohledávané asymetrické 4-okolí [5]

Pro rychlou implementaci byl vytvořen rozhodovací strom (viz obr. 2.8), s jehož pomocí je možné rychle rozhodnout o příslušnosti daného pixelu k některému z objektů v okolí. V některých případech může dojít ke sloučení objektů. Pro tento účel algoritmus navíc obsahuje tabulku ekvivalentních značek uchávající informaci o tom, které objekty se mají sloučit do jednoho. Pro udržení konzistence a zároveň jednoduchosti je tabulka ekvivalentních značek implementována pomocí struktury union-find, též podrobně popsané v práci [5].



Obr. 2.8 – Rozhodovací strom labelingu [5]

### 2.1.6 Získané parametry objektů

Již v průběhu labelingu je nutné o objektech uchovávat potřebné informace, aby bylo možné určit jejich parametry. K základním parametrům, které lze určit přímo na základě labelingu, je plocha objektu  $S$  udávaná v pixelech a ohraničující

obdélník, ze kterého dále dá určit velikost objektu. Další parametry se dají vypočítat pomocí statistických metod momentů a centrálních momentů, které jsou podrobněji popsány v práci [6]. My se budeme věnovat pouze prvním momentům, mezi něž patří například jakýsi ekvivalent hmotnosti objektu  $M$ , který se dá vypočítat podle:

$$M = \sum_x \sum_y I(x, y) \quad (3)$$

Kde  $x$  a  $y$  jsou souřadnice v obrazu a  $I(x, y)$  je intenzita daného pixelu.

Dalším parametrem je poloha těžiště, která se dá vypočítat podle vzorce:

$$X_T = \frac{1}{M} \sum_x \sum_y x I(x, y) \quad (4)$$

Kde  $X_T$  je poloha těžiště v ose  $X$ . Ekvivalentně lze vypočítat polohu těžiště i v ose  $Y$ .

Alternativně lze vypočítat polohu středu neboli těžiště homogenního tělesa, tedy bez uvažování jasových hodnot. Pro tento případ platí:

$$X_C = \frac{1}{S} \sum_x \sum_y x \quad (5)$$

Kde  $X_C$  je poloha středu a  $S$  je plocha objektu v pixelech.

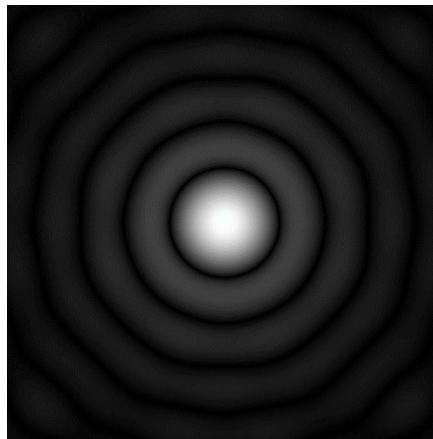
Při výpočtu těžiště po předchozím prahování se též dá využít výpočet těžiště s odseknutím minimální hodnoty. Pro výpočet těžiště pak platí:

$$X_{Tc} = \frac{1}{\sum_x \sum_y [I(x, y) - \min(I(x, y))]} \sum_x \sum_y x [I(x, y) - \min(I(x, y))] \quad (6)$$



## 2.2 Určení polohy jasových značek

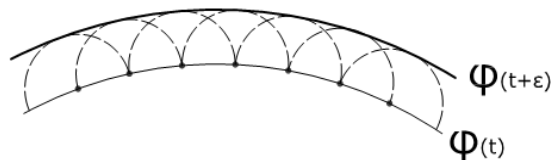
Dalším typem obrazu, se kterým se lze setkat je obraz s jasovými značkami, jejichž polohu chceme určit. Jasovou značkou v obrazu se myslí místo s výrazně vyšším jasnem. Tyto značky obvykle mívají malé rozměry a mohou vzniknout posvícením laserem na plochu nebo promítnutím laserového paprsku skrze štěrbinu. Promítání paprsku skrz štěrbinu je z hlediska fyzikálního popisu a vzniklého obrazce (obr. 2.9) mnohem zajímavější, proto se mu budeme dále věnovat.



Obr. 2.9 – Příklad difrakčního obrazce

### 2.2.1 Difrakční jevy na štěrbině

Difrakce neboli ohyb v našem případě světla vzniká na jakékoliv překážce. Je tomu tak díky Huygens–Fresnelově principu [7], který zjednodušeně říká, že každá část vlnoplochy se stává samostatným všesměrovým zdrojem záření, jak je naznačeno na obr. 2.10. Každý bod původní neboli primární vlnoplochy vyzáří novou sekundární vlnoplochu.



Obr. 2.10 – Znázornění Huygens–Fresnelova principu

Díky tomuto jevu se záření dostává i do míst tzv. geometrického stínu za překážkou. Navíc za každou překážkou spolu sekundární vlnoplochy interferují

a vznikají tak difrakční obrazce. Prochází-li světlo štěrbínou, lze rozdělit vznikající difrakční jevy do dvou kategorií na tzv. Fresnelovu a Fraunhoferovu difrakci [8]. Ve které ze dvou oblastí se pohybujeme, je vyjádřeno pomocí Fresnelova čísla  $F$  [7]:

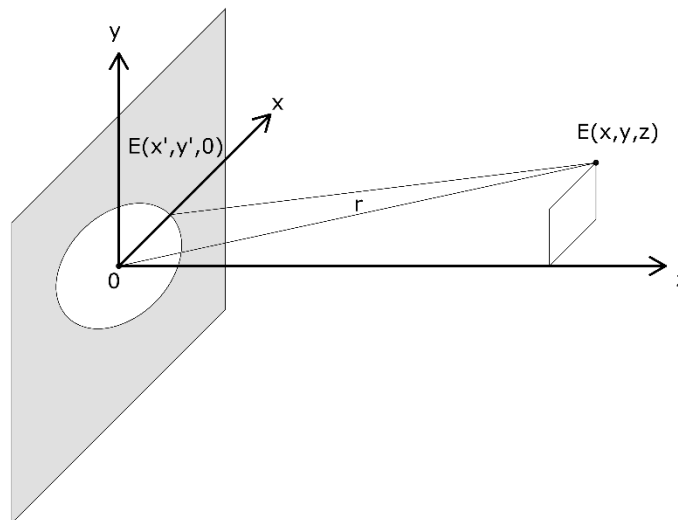
$$F = \frac{a^2}{L\lambda} \quad (7)$$

Kde  $a$  je charakteristický rozměr štěrbiny,  $\lambda$  je vlnová délka záření a  $L$  je vzdálenost stínítka od štěrbiny.

Pro velké vzdálenosti je  $F < 1$  a nacházíme se tak ve vzdálené zóně, kde hovoříme o Fraunhoferově difrakci. Ta je obecně známější, avšak jedná se o speciální případ Fresnelovy difrakce vznikající v blízké zóně kdy je  $F \gg 1$ . V našem případě bude velikost štěrbiny  $a = 0,35 \text{ mm}$ , vlnová délka záření  $\lambda = 650 \text{ nm}$  a vzdálenost od štěrbiny budou jednotky centimetrů. Fresnelovo číslo v tomto případě vychází přibližně  $F \sim 1$  a budeme se tak nacházet na rozhraní obou difrakcí. Musíme tedy uvažovat obecnější z nich, a sice Fresnelovu. Intenzita záření v prostoru  $E(x, y, z)$  je dána vztahem [8]:

$$E(x, y, z) = -\frac{ik}{2\pi} \iint_{-\infty}^{+\infty} E(x', y', 0) \frac{e^{ikr}}{r} dx' dy' \quad (8)$$

Kde  $E(x', y', 0)$  je funkce intenzity záření procházející stínítkem,  $e^{ikr}/r$  je kulová plocha ve vzdálenosti  $r = \sqrt{(x - x')^2 + (y - y')^2 + z^2}$  od štěrbiny a  $k = 2\pi/\lambda$ . Prostorové uspořádání je naznačeno na obr. 2.11



Obr. 2.11 – Znázornění geometrického uspořádání

### 2.2.1.1 Fresnelova Aproximace

Výše uvedený integrál je jen velmi těžko řešitelný analyticky, proto již sám Fresnel provedl aproximaci kulové plochy prvními dvěma členy mocninné řady [8].

$$r = \sqrt{(x - x')^2 + (y - y')^2 + z^2} \approx z \left( 1 + \frac{(x - x')^2 + (y - y')^2}{2z^2} \right) \quad (9)$$

Kulovou plochu z původního integrálu tedy převedeme do tvaru:

$$\frac{e^{ikr}}{r} \approx \frac{e^{ikz}}{z} e^{\left( \frac{ik}{2z} ((x-x')^2 + (y-y')^2) \right)} \quad (10)$$

Po dosazení do původního integrálu dostaneme:

$$E(x, y, z) = -\frac{ik}{2\pi} \frac{e^{ikz}}{z} \iint_{-\infty}^{+\infty} E(x', y', 0) e^{\left( \frac{ik}{2z} ((x-x')^2 + (y-y')^2) \right)} dx' dy' \quad (11)$$

Druhé mocniny exponentu můžeme rozvést a část vytknout před integrál:

$$\begin{aligned} E(x, y, z) &= \\ &= -\frac{ik}{2\pi} \frac{e^{ikz}}{z} e^{\left( \frac{ik}{2z} (x^2 + y^2) \right)} \iint_{-\infty}^{+\infty} E(x', y', 0) e^{\left( \frac{ik}{2z} (x'^2 + y'^2) \right)} e^{\left( -\frac{ik}{z} ((xx') + (yy')) \right)} dx' dy' \end{aligned} \quad (12)$$

Uvnitř integrálu nám nyní vzniklo jádro Fourierovy transformace, které nám dále usnadní výpočet. Pro  $z \geq 0 = konst.$  lze celý integrál chápat jako konvoluci intenzity záření  $E_0(x, y)$  procházející stínítkem a druhou částí, která nám vznikla aproximací kulové vlnoplochy (9). Označme si ji  $h(x, y)$ . Integrál můžeme tedy přepsat do tvaru konvoluce:

$$E(x, y, z) = -\frac{ik}{2\pi} \frac{e^{ikz}}{z} e^{\left( \frac{ik}{2z} (x^2 + y^2) \right)} (E_0(x, y) * h(x, y)) \quad (13)$$

Přejdeme-li do frekvenční oblasti pomocí 2D Fourierovy transformace, dostaneme:

$$E(x, y, z) = -\frac{ik}{2\pi} \frac{e^{ikz}}{z} e^{\left( \frac{ik}{2z} (x^2 + y^2) \right)} \mathcal{F}^{-1} \{ \mathcal{F} \{ E_0(x, y) \} \cdot H(x, y) \} \quad (14)$$

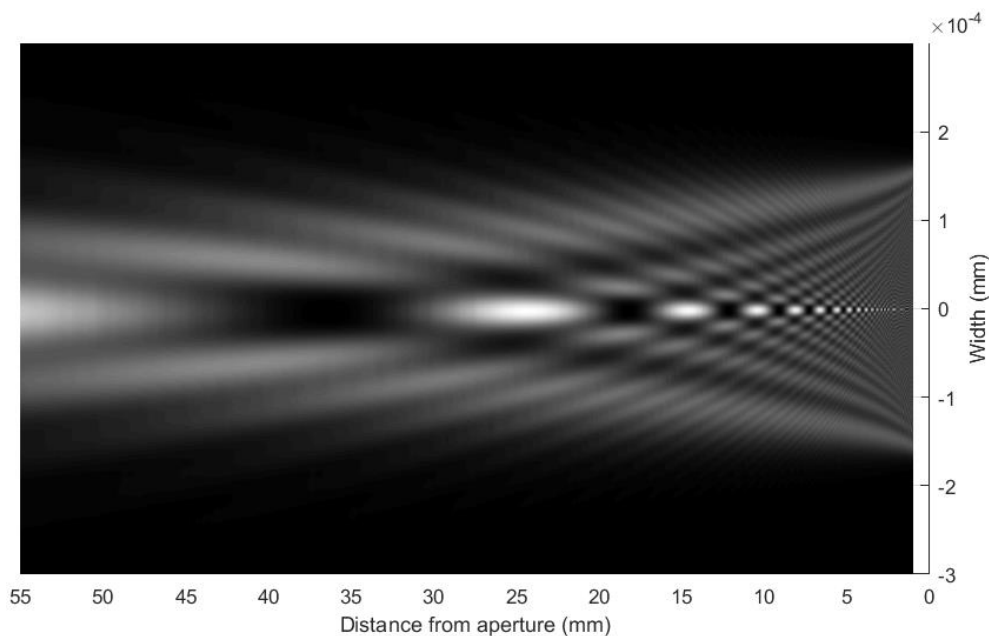
Kde operátor  $\mathcal{F}$  značí 2D Fourierovu transformaci a  $H(x, y) = \mathcal{F} \{ h(x, y) \}$  Díky jádru Fourierovy transformace vzniklému v (11) je  $H(x, y)$ :

$$H(x, y) = e^{\left(\frac{ik}{2z}(x^2+y^2)\right)} \quad (15)$$

Fyzikálně lze tuto rovnici chápat jako vyjádření fáze, se kterou záření dospěje do roviny stínítka. Pro Fraunhoferovu difrakci, kde platí  $z \gg x, y$ , se fázor (14) blíží hodnotě  $H(x, y) \cong 1$  a v bodě (11) pak přímo vidíme, že obrazec Fraunhoferovy difrakce je roven Fourierově transformaci intenzity záření  $E_0(x, y)$  procházející stínítkem. Pro obecná stínítka (kruh, čtverec) s propustností 0 nebo 1 se pak v různých směrech nachází různě široký obraz Fourierovy transformace obdélníka tedy funkce  $\sin(x)/x$ . Toto však pro náš případ, kdy se pohybujeme v oblasti Fresnelovy difrakce, neplatí, nicméně vzniklé obrazce fyzikálně též vycházejí z tvaru funkce  $\sin(x)/x$ .

### 2.2.1.2 Simulace difrakčních jevů v prostředí Matlab

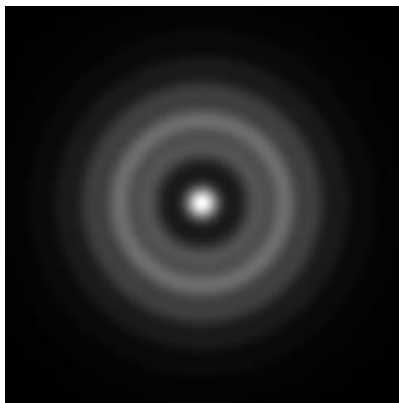
Takto upravený integrál (13) již lze numericky vypočítat pomocí programu Matlab [9]. Difrakční obrazec pro šěrbinu s kruhovým otvorem o průměru  $r = 0,35 \text{ mm}$  a vlnovou délkou světla  $\lambda = 650 \text{ nm}$  je na obr. 2.12. Obrázek znázorňuje intenzitu ozáření  $I(x, y, z) = E(x, y, z)^2$  v řezu obrazcem v rovině  $I(x, y, z)|_{x=0}$



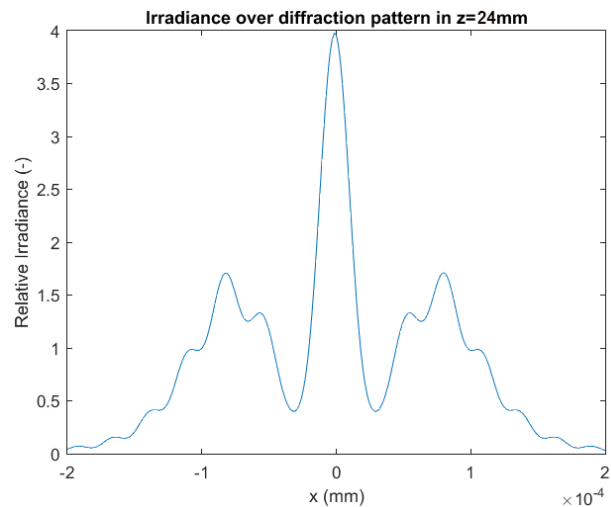
Obr. 2.12 – Průřez difrakčním obrazcem

Z obr. 2.12 je patrné, že na ose  $z$  vznikají maxima a minima. Intenzita záření na ose  $E(x, y, z)|_{x=0, y=0}$  má průběh ve tvaru funkce  $E(z) \sim \sin(1/z)$ . Jako nejvhodnější z hlediska dalšího zpracování se jeví vybrat si za bod zájmu některé z maxim. Pro

vzdálenosti  $z > 55 \text{ mm}$  je sice obrazec relativně jednoduchý a má přibližně tvar funkce  $\sin(x)/x$ , ale je příliš daleko. Blízká maxima jsou naopak citlivá na přesné nastavení polohy v ose  $Z$ , proto se jako nelepší jeví zvolit si za bod zájmu maximum vznikající ve vzdálenosti  $z = 24 \text{ mm}$ . Celý obrazec v této vzdálenosti bude mít následující tvar (obr. 2.14) a průřez jasových hodnot bude odpovídat obr. 2.13, kde též vidíme, že při jednotkové intenzitě záření procházející šterbinou je intenzita ozáření středu difrakčního obrazce 4x vyšší než v rovině šterbiny.



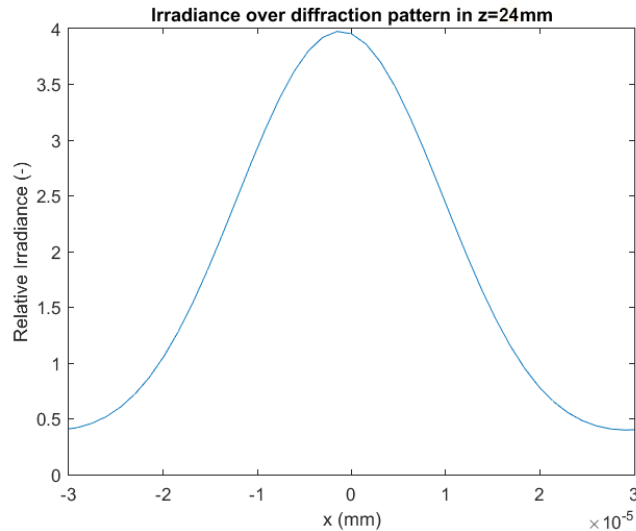
Obr. 2.14 – Difrakční obrazec



Obr. 2.13 – Jasový průřez středem difrakčního obrazce

### 2.2.2 Aproximace difrakčních obrazců

Difrakční obrazec znázorněný na obr. 2.14 a obr. 2.13 má šířku prostředního maxima cca  $40 \mu\text{m}$ . CMOS senzor použitý k určení polohy obrazce má velikost jednoho pixelu  $6 \mu\text{m}$ . Difrakční obrazec se tedy zobrazí pouze na několika pixelech a je dobré jej, pro dosažení sub-pixelového rozlišení, aproximovat vhodnou matematickou funkcí. Podíváme-li se na střed obrazce, viz obr. 2.15 vidíme, že obrazec by se dal aproximovat například Gaussovou funkcí.



Obr. 2.15 – Střed difrakčního obrazce

Výpočet této aproximace však není triviální, proto se nejdříve podíváme na polynomiální aproximace prováděné metodou nejmenších čtverců [10].

### 2.2.2.1 Lineární metoda nejmenších čtverců

Metoda nejmenších čtverců se snaží najít takovou aproximaci  $\hat{f}(x, \mathbf{p})$  signálu  $f(x)$ , aby chyba  $e(x) = f(x) - \hat{f}(x, \mathbf{p})$  byla co nejmenší. V maticovém zápisu se jedná o hledání řešení přeuročené soustavy rovnic  $\mathbf{Q}\mathbf{p} = \mathbf{f}$ . Metoda proto zavádí optimalizační kritérium:

$$\mathbf{p} = \arg \min(S(\mathbf{p})) = \arg \min \left( \sum_i (f(x_i) - \hat{f}(x_i, \mathbf{p}))^2 \right) \quad (16)$$

Kde  $i$  jsou jednotlivá pozorování, které máme k dispozici a aproximace  $\hat{f}(x_i, \mathbf{p})$  s parametry  $\mathbf{p}$  má tvar:

$$\hat{f}(x_i, \mathbf{p}) = \sum_k Q_{ik} p_k \quad (17)$$

Kde  $k$  je počet parametrů a  $Q_{ik}$  vyjadřuje funkci jednotlivých parametrů.

Pro nalezení minima optimalizačního kritéria  $S$  položíme jeho první derivace podle jednotlivých parametrů rovny nule, tedy:

$$\frac{\partial S(\mathbf{p})}{\partial p_j} = -2 \sum_i \frac{\partial \hat{f}(x_i, \mathbf{p})}{\partial p_j} (f(x_i) - \hat{f}(x_i, \mathbf{p})) = 0 \quad (18)$$

Derivujeme-li funkci  $\hat{f}(x_i, \mathbf{p})$  podle parametru  $p_j$  dostaneme:

$$\frac{\partial \hat{f}(x_i, \mathbf{p})}{\partial p_j} = Q_{ij} \quad (19)$$

Po dosazení aproximace  $\hat{f}(x_i, \mathbf{p})$  a její derivace do (17) získáme:

$$\sum_i \sum_k Q_{ij} Q_{ik} p_k = \sum_i Q_{ij} f(x_i) \quad (20)$$

Přejitím zpět k maticovému zápisu vznikne:

$$(\mathbf{Q}^T \mathbf{Q}) \mathbf{p} = \mathbf{Q}^T \mathbf{f} \quad (21)$$

Řešení této rovnice poskytuje nejlepší aproximaci funkce  $f(x)$  ve smyslu nejmenších čtverců.

### 2.2.2.2 Aproximace paraboloidem

Nejjednodušší je aproximovat střed difrakčního obrazce z obr. 2.14 polynomem 2. řádu o dvou proměnných neboli paraboloidem.

$$f(x, y) = p_1 x^2 + p_2 y^2 + p_3 xy + p_4 x + p_5 y + p_6 \quad (22)$$

Střed takto nalezeného paraboloidu lze najít pomocí první derivace funkce:

$$\frac{\partial f}{\partial x} = 2p_1 x + p_3 y + p_4 = 0 \quad \frac{\partial f}{\partial y} = 2p_2 y + p_3 x + p_5 = 0 \quad (23)$$

Tato soustava rovnic má právě jedno řešení, které odpovídá globálnímu extrému paraboloidu, což je hledaný střed difrakčního obrazce.

### 2.2.2.3 Aproximace polynomem 4. řádu

Přesnější aproximací středu difrakčního obrazce by měl být polynom 4. řádu o dvou proměnných ve tvaru:

$$f(x, y) = p_1 x^4 + p_2 y^4 + p_3 x^3 y + p_4 x y^3 + p_5 x^2 y^2 + p_6 x^3 + p_7 y^3 + p_8 x^2 y + p_9 x y^2 + p_{10} x^2 + p_{11} y^2 + p_{12} x y + p_{13} x + p_{14} y + p_{15} \quad (24)$$

Takto vzniklý útvar však již má několik bodů podezřelých z extrému a řešení pomocí první derivace vyžaduje iterační metody. V tomto případě však lze využít známého tvaru objektu. Při předchozím zpracování lze zhruba určit polohu středu. Dosazením

takto získaného středu do první derivace získáme směr, kterým se nachází lokální maximum. Iteračně tedy lze dospět do lokálního maxima, aniž by bylo nutné prohledávat velkou část definičního oboru funkce.

#### 2.2.2.4 Nelineární metoda nejmenších čtverců

Pro získání fyzikálně nejpodobnější aproximace středu difrakčního obrazce pomocí Gaussovy funkce je potřeba využít nelineárních aproximačních metod jako je například nelineární metoda nejmenších čtverců [11]. Její základ je definován stejně jako pro lineární případ:

$$S(\mathbf{p}) = \sum_i \left( f(x_i) - \hat{f}(x_i, \mathbf{p}) \right)^2 \quad (25)$$

Stejně jako v předchozím případě řešíme soustavu rovnic skládající se z prvních derivací:

$$\frac{\partial S(\mathbf{p})}{\partial p_j} = -2 \sum_i \frac{\partial \hat{f}(x_i, \mathbf{p})}{\partial p_j} \left( f(x_i) - \hat{f}(x_i, \mathbf{p}) \right) = 0 \quad (26)$$

Avšak derivace funkce  $\hat{f}(x_i, \mathbf{p})$  podle parametru  $p_j$  není v případě Gaussovy funkce lineárně nezávislá na dalších parametrech, proto zde neplatí vztah (19) a tuto soustavu lze řešit pouze iteračně od odhadu řešení  $\mathbf{p}_0$ , které na základě znalosti tvaru funkce můžeme poměrně dobře určit.

Hodnotu funkce  $\hat{f}(x_i, \mathbf{p})$  a její derivaci nahradíme Taylorovým rozvojem v okolí  $\mathbf{p}_0$ :

$$\hat{f}(x_i, \mathbf{p}) = \hat{f}(x_i, \mathbf{p}_0) + \sum_k \frac{\partial \hat{f}(x_i, \mathbf{p}_0)}{\partial p_k} \Delta p_k \quad (27)$$

$$\frac{\partial \hat{f}(x_i, \mathbf{p})}{\partial p_j} = \frac{\partial \hat{f}(x_i, \mathbf{p}_0)}{\partial p_j} + \sum_k \frac{\partial^2 \hat{f}(x_i, \mathbf{p}_0)}{\partial p_j \partial p_k} \Delta p_k \quad (28)$$

Dosazením do rovnice (26) získáme novou soustavu rovnic:

$$\sum_k \left( \sum_j \frac{\partial \hat{f}(x_i, \mathbf{p}_0)}{\partial p_k} \frac{\partial \hat{f}(x_i, \mathbf{p}_0)}{\partial p_j} \right) \Delta p_k = \sum_i \frac{\partial \hat{f}(x_i, \mathbf{p}_0)}{\partial p_j} \left( f(x_i) - \hat{f}(x_i, \mathbf{p}) \right) \quad (29)$$



Soustavu rovnic (29) lze přepsat do maticového zápisu:

$$(\mathbf{J}^T \mathbf{J}) \Delta \mathbf{p} = \mathbf{J}^T \Delta \mathbf{f} \quad (30)$$

Kde  $\mathbf{J}$  je matice parciální derivací tzv. Jakobiho matice,  $\Delta \mathbf{p}$  je posun vektoru řešení  $\mathbf{p}$  pro který platí  $\mathbf{p} = \mathbf{p}_0 + \Delta \mathbf{p}$  a  $\Delta \mathbf{f}$  je vektor odchylek  $\Delta f(x_i) = f(x_i) - \hat{f}(x_i, \mathbf{p})$ .

Soustava rovnic (30) má podobný tvar jako u lineární metody nejmenších čtverců, až na to, že zde je místo matice  $\mathbf{Q}$  použita Jakobiho matice  $\mathbf{J}$  s parciálními derivacemi. Iterační postup je zde podobný jako u aproximace polynomem 4. řádu, kdy se každá další iterace pohybuje ve směru derivace k lokálnímu maximu resp. minimu optimalizační funkce.

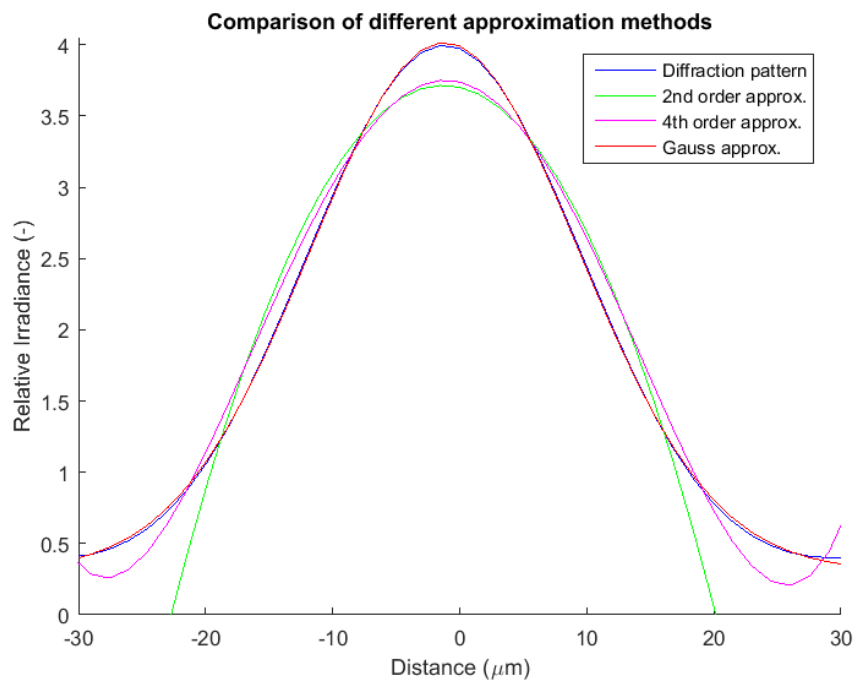
#### 2.2.2.5 Aproximace Gaussovou funkcí dvou proměnných

Gaussova funkce dvou proměnných, jež dobře vystihuje difrakční obrazec, má tvar:

$$f(x, y) = ae^{((x-x_0)^2/q^2 - (y-y_0)^2/q^2)} + c \quad (31)$$

Kde je použit vektor parametrů  $\mathbf{p} = (a, x_0, y_0, q, c)$ . Při použití této aproximace není nutné po jejím provedení hledat střed, protože souřadnice středu  $(x_0, y_0)$  jsou již zahrnuty v parametrech. Oproti klasické definici Gaussovy funkce je zde pro jednoduchost zavedena konstanta  $a$ , díky které nemusí být objem pod křivkou jednotkový a přidána konstanta  $c$ , která zajišťuje posun celé funkce, protože v uvažovaném případě minimum funkce nemusí být 0.

Porovnání přesností různých aproximací lze vidět na obr. 2.16. Jedná se o aproximace teoretického tvaru středu difrakčního obrazce. Na reálných datech, kdy bude obrazec navzorkován pouze přes několik pixelů, se dají očekávat horší výsledky. Všechny aproximace budou v kapitole 5.2.3 porovnány na reálných datech.

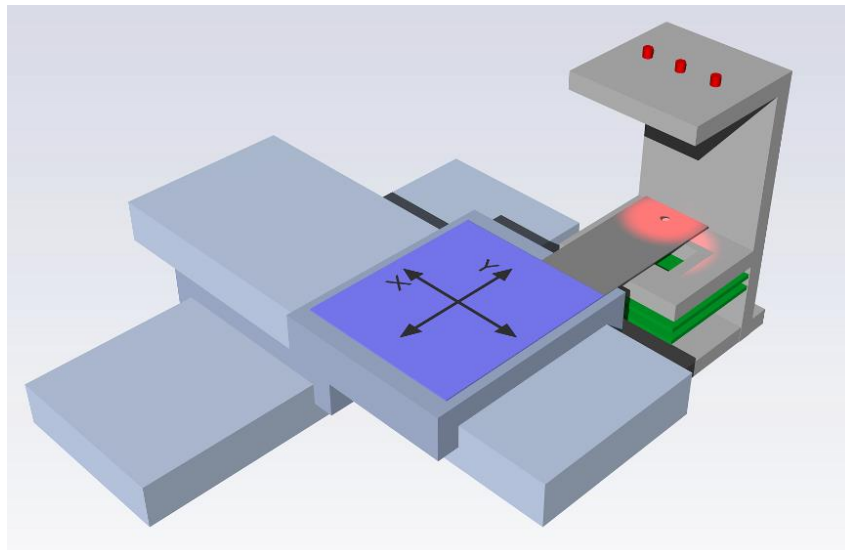


Obr. 2.16 – Porovnání různých aproximací

### 3 Návrh polohovacího systému a jeho součástí

Z rozboru na začátku práce vyplývá, že se vyhodnocení jasové značky dá použít k měření polohy. Za účelem ověření této metody je však potřeba navrhnout a realizovat experiment, proto se v této kapitole pokusíme získat podrobnou představu jak má polohovací systém vypadat a co vše je nutné vytvořit a dořešit.

Ideový náčrt polohovacího systému je k vidění na obr. 3.1. Odstíny modré barvy znázorňují polohovací mechanismus umožňující posuv v osách X a Y. Tento polohovací mechanismus se pohybuje nad CMOS senzorem. Z vrchu je polohovací mechanismus osvětlen laserovými diodami. Polohu systému pak lze určit přímo z polohy jasové značky.



Obr. 3.1 – Ideový náčrt systému

### 3.1 Procesorový a obrazový modul

Pro rychlé zpracování obrazu a případně výpočet matematických aproximací je potřeba poměrně velký výpočetní výkon. Za tímto účelem byl v rámci práce [12] vytvořen procesorový modul se signálovým procesorem ADSP-BF533 Blackfin jež lze použít. Více informací o použitém procesoru se dá najít například v [13] a [14]. Zde jsou uvedeny pouze periferie, které budou a úloze potřeba:

- HS USB radič FT2232H [15] – Komunikace s PC
- PPI – Paralelní interface pro načítání dat obrazového modulu
- I2C – Komunikace s obrazovým modulem
- UART – Pro ovládání pojezdů
- GPIO – Ovládání laserů

Uvedený signálový procesor disponuje čtyřmi 8-bitovými ALU jednotkami a využívá metod pipeliningu. Díky více ALU jednotkám, které jsou optimalizovány pro operace sčítání a posouvání, může procesor zpracovávat některé operace paralelně. Díky pipeliningu procesor navíc zpracovává více instrukcí najednou v sérii, čímž mu bohužel trvá déle vykonání podmíněného skoku. Z těchto důvodů je při jeho programování nutné věnovat pozornost implementovaným algoritmům. Programátor by se měl snažit minimalizovat počet podmíněných skoků v časově kritických částech kódu a co nejméně používat operace typu násobení, dělení a modulo. Dále je vhodné používat pouze cykly s inkrementací proměnné o 1, jinak procesor nemůže použít vnitřní hardwarovou podporu smyček. Více se o specifikách při programování uvedeného procesoru lze dozvědět v [16].

Firma Analog Devices ke svým procesorům dodává balíček knihoven ve kterém kromě základních matematických funkcí jako je sinus, exponenciální funkce, mocniny atd. lze najít i operace s maticemi, které budou velmi nápomocné při výpočtech aproximací. Funkce implementované v originální knihovně vždy vychází rychlejší než naivní nebo mírně pokročilá implementace.

Pro danou úlohu je nejdůležitějším prvkem připojeným k procesorovému modulu obrazový modul, který je též standardně používán v laboratoři videometrie.

V tomto případě je modul osazen CMOS senzorem Aptina MT9V032. Podrobné parametry použitého senzoru lze nalézt v [17]. Zde jsou zmíněny pouze základní parametry důležité pro danou úlohu:

- Rozlišení: WVGA – 752 x 480 *pix*
- Velikost: 1/3-inch – 4,51 x 2,88 *mm*
- Velikost pixelu: 6 x 6  $\mu m$
- Typ barev: Černobílý
- Rychlost čtený dat: Maximálně 27 *MHz*
- Závěrka: Global shutter

Jedním z důležitých parametrů, který bude ovlivňovat přesnost měření je velikost aktivní plochy senzoru neboli fill-factor. Z hlediska ideálního navzorkování pro další matematické zpracování by měla být aktivní plocha jednoho pixelu infinitezimálně malá. Výrobci se však snaží dělat aktivní část pixelů co největší, aby zachytili více světla. Díky tomu není hodnota jednoho pixelu přesným vyjádřením intenzity v daném bodě, ale průměrnou hodnotou z části celého pixelu. Tím může vzniknout periodická chyba v rámci jednoho pixelu a sníží se tak subpixelové rozlišení.

### 3.2 Pojezdy pro polohování

Z výše uvedeného nákresu systému vidíme, že změna polohy systému přímo úměrně odpovídá změně polohy stopy na CMOS senzoru. Z toho důvodu musí mít pojezdy použité pro polohování řádově větší přesnost, než je velikost jednoho pixelu. V rámci dřívějších projektů byly v laboratoři videometrie pořízeny dva precizní pojezdy od firmy Physik Instrumente M-511.DD [18] jejichž parametry jsou následující:

- Pohon: DC motor řízen PWM
- Regulátor polohy: PID (anti-windup)
- Délka pojezdu: 100 *mm*
- Měření polohy: Lineární inkrementální snímač s interpolací

- Vzdálenost stop:  $20 \mu m$
- Opakovatelnost:  $\pm 1 \mu m$
- Rozlišení:  $0,1 \mu m$
- Mechanická vůle:  $\pm 0,2 \mu m$

Pojezdy jsou řízeny pomocí driverů Physik Instrumente C-863 [19], které se dají ovládat přes sériovou linku RS-232. Drivery navíc mají konektory pro daisy-chain propojení, takže je lze ovládat pomocí jedné sériové linky a určitý pojezd aktivovat softwarově.

### 3.3 Řízení laserů

Úloha vyžaduje pokud možno homogenní osvětlení celé plochy senzoru monochromatickým koherentním zářením. Ideální řešení poskytují laserové diody, které mají dostatečné úzké spektrum ve kterém září. Použije-li se laserová dioda bez kolimační optiky poskytuje z určité vzdálenosti dostatečné homogenní osvětlení. Koherenční délka levných laserových LED diod se pohybuje v řádech desítek  $\mu m$ , což je pro danou úlohu postačující.

Systém lze obohatit o měření vzdálenosti od senzoru a určovat polohu ve všech 3 osách. Za tímto účelem bude použito více laserových diod v přesné vzdálenosti od sebe. Podle vzdálenosti laserových stop vzniklých na CMOS senzoru lze ze znalosti polohy laserových diod určit vzdálenost. Systém tak bude schopen v omezeném prostoru určit polohu ve všech třech osách. Podrobněji je geometrie měření popsána v kapitole 4.3.

Zmíněné laserové diody však potřebují buzení. Pro zajištění konstantního jasu více diod je nutné monitorovat jejich světelný tok. Za tímto účelem bude muset být vyvinut budič ovládaný z procesoru a umožňující nastavení jasu jednotlivých diod. Návrh budiče je podrobně popsán v kapitole 4.4.

### **3.4 Mechanická konstrukce**

Z nákresu systému je evidentní, že je nutné vytvořit dostatečně tuhou mechanickou konstrukci nesoucí nejen moduly s procesorem, CMOS senzorem a budič laserů, ale i pojezdů. Některé části mechanické konstrukce bude nutné navrhnout a vyrobit, čemuž se věnuje kapitola 4.1.

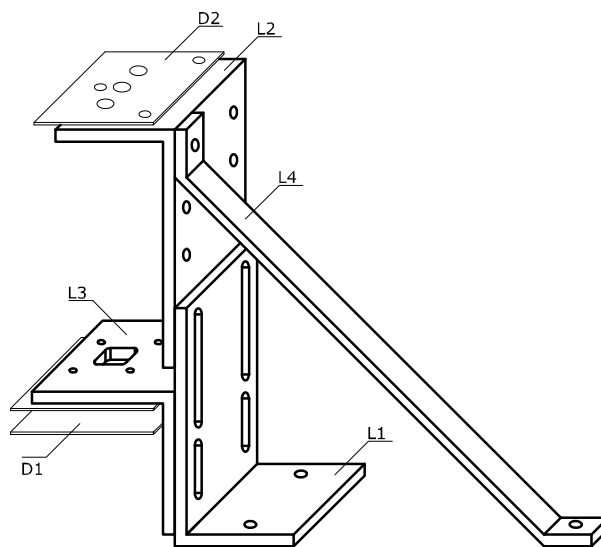
Provedení konstrukce bude velmi ovlivňovat přesnost měření. Kromě dostatečné tuhosti, je nutné dosáhnout co nejlepší kolmosti souřadnic pojezdů a CMOS senzoru. Dalším parametrem, který ovlivní přesnost je teplotní roztažnost mechanické konstrukce.

## 4 Hardwarové řešení

Celý systém popsaný výše je potřeba postavit a vytvořit chybějící elektrické a i mechanické součásti. Za účelem vytváření pevných mechanických konstrukcí jsou v laboratoři videometrie k dispozici hliníkové profily MayTec [20], které poslouží jako základna pro pojezdy a konstrukci držící procesorový a obrazový modul.

### 4.1 Vytvoření mechanické konstrukce

Návrh výše zmíněné konstrukce držící procesorový modul, obrazový modul a lasery je znázorněn na obr. 4.1.



Obr. 4.1 – Náčrves konstrukce držící CMOS senzor a lasery

Uvedená konstrukce je vytvořena z hliníkových L profilů o rozměrech  $100 \times 50 \text{ mm}$  a  $60 \times 60 \text{ mm}$ , které byly obráběny na CNC frézce, díky čemuž by měla být zajištěna přesnost. Pro zvýšení tuhosti byla přidána vzpěra L4. Přesný výkres konstrukce je k nalezení v příloze.

Takto vytvořená konstrukce je společně s pojezdy připevněna na profilech MayTec. Fotografie celého systému viz fotopříloha C. Souřadnice pojezdů se podařilo



srovnat s rastrem CMOS senzoru s odchylkou 1 *pix* v obou směrech, což odpovídá úhlové chybě  $0^{\circ} 05'$  v delším směru resp.  $0^{\circ} 07'$  v kratším směru senzoru. V úzkém rozsahu pojezdu se tedy dají souřadné systémy považovat za shodné.

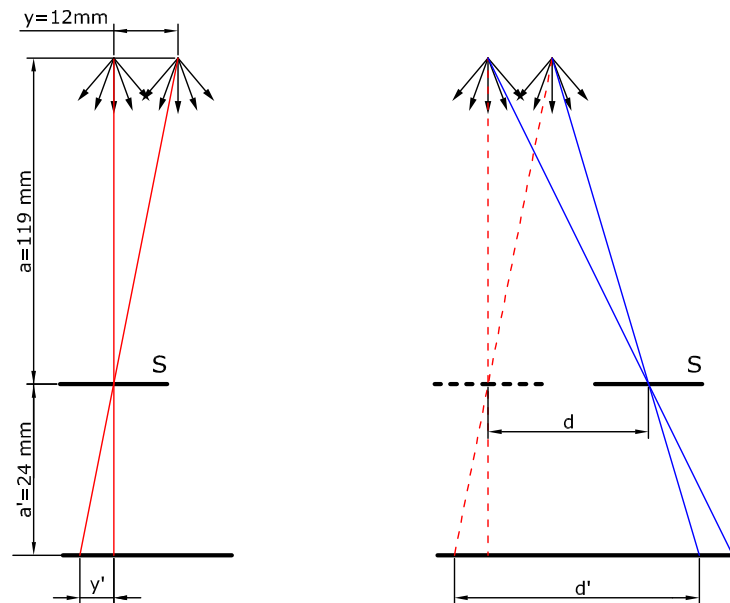
## 4.2 Nepřesnosti mechanické konstrukce

Kromě výše popsaných nepřesností sestavení konstrukce je nutné počítat i s teplotní roztažností zmíněnou v teoretickém rozboru 3.4. Základna vytvořená z hliníkových profilů MayTec je provázaná ve více bodech. Teplotní roztažnosti základny tudíž zanedbáme.

Důležitější pro přesnost systému bude například teplotní roztažnost kuličkových šroubů použitých v pojezdech. Součinitel teplotní roztažnosti železa je  $12 \cdot 10^{-6} K^{-1}$  což při délce šroubů 100 *mm* znamená 1,2  $\mu m$  na  $1^{\circ}C$ . Tato změna nastává v obou osách a při subpixelové přesnosti určitě ovlivní měření. Další komponentou, jejíž délka se bude s teplotou hodně měnit, je hliníkový profil držící clonku. Teplotní roztažnost hliníku je  $23,8 \cdot 10^{-6} K^{-1}$ . Při délce cca 80 *mm* činí změna délky 1,9  $\mu m$  na  $1^{\circ}C$ . Tato změna by opět měla být měřitelná. Poslední částí, kterou je nutné uvažovat je roztažnost konstrukce laserů vyrobené též z hliníku. Při výšce 150 *mm* činí změna délky na  $1^{\circ}C$  3,6  $\mu m$ . Tato změna způsobí změnu zvětšení promítané stopy, ale při celkové délce 150 *mm* by měla mít zanedbatelný vliv.

### 4.3 Geometrie soustavy

Geometrický popis soustavy, důležitý pro odvození vztahů pohybu obrazce v závislosti na pohybu systému, je na obr. 4.2. Jsou zde též znázorněny charakteristické rozměry systému důležité pro získání správného tvaru jasové značky. Na situaci vlevo jsou získány dvě polohy jasové značky, každá pro jeden zdroj světla. V pravé části je systém posunut o doprava.



Obr. 4.2 – Znáznornění systému pohledem geometrické optiky

Z obrázku obr. 4.2 vyplývá, že pro poměr posunu obrazce na senzoru  $d'$  vůči posunu  $d$  platí vztah:

$$\frac{d'}{d} = \frac{a + a'}{a} = k \quad (32)$$

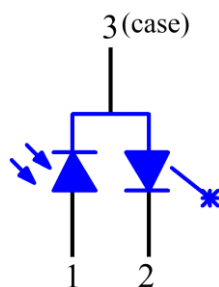
Kde  $k$  má pro konkrétní případ hodnotu  $k \cong 1,201$  a lze díky němu dopočítat posun systému ze znalosti posunu obrazce na senzoru.

Dalším parametrem, který lze zjistit díky více laserům vzdálených od sebe  $y = 12 \text{ mm}$ , je změna vzdálenosti od senzoru. Vztah pro změnu vzdálenosti  $\Delta a'$ , který lze odvodit z podobnosti trojúhelníků má tvar:

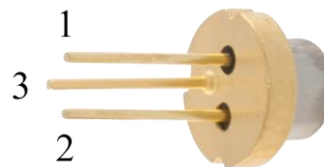
$$\Delta a' = \frac{ay' - a_0y}{y' + y} \quad (33)$$

## 4.4 Návrh budiče laserů

Úloha vyžaduje osvětlení s konstantní intenzitou v čase. Z toho důvodu je nutné světelný tok jednotlivých zdrojů záření monitorovat a stabilizovat pomocí zde navrhovaného regulátoru. Jako zdroj světla byly vybrány laserové diody L-SLD6505 [21] obsahující fotodiodu, která umožňuje snímání světelného toku. Z vnitřního schéma diod (viz obr. 4.4 a obr. 4.3) vidíme, že na pouzdro součástky je přivedena katoda fotodiody a anoda laserové diody, což lehce komplikuje návrh, protože pro zachování GND na pouzdře součástky by bylo potřeba symetrické napájení.



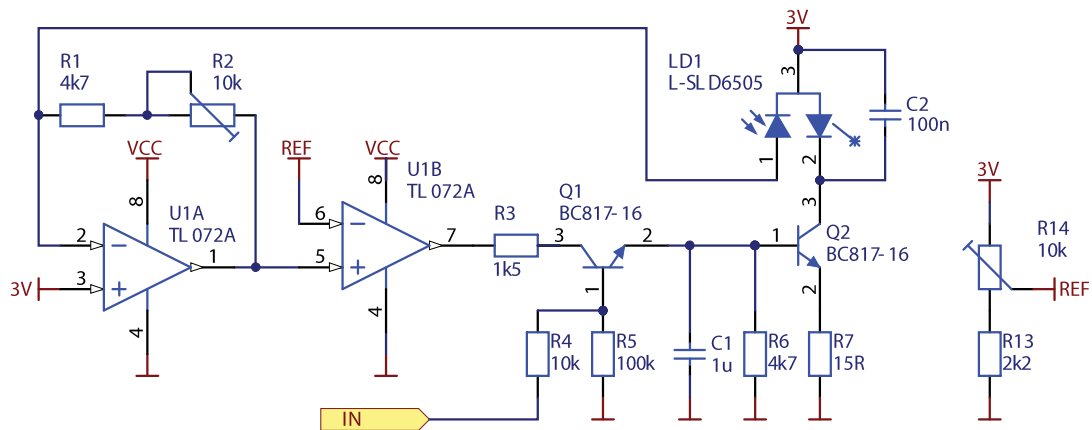
Obr. 4.3 – Schéma laserové diody



Obr. 4.4 – Pouzdro laserové diody

V tomto případě však nevadí, pokud na pouzdře bude jiné napětí než GND proto je pro napájení diod použito stabilizované napětí 3V a pro napájení operačních zesilovačů o něco vyšší napětí cca 5V, aby byla zajištěna jejich správná funkce.

Požadavkem na driver laserů bylo, aby se dal řídit jas jednotlivých diod jak samostatně, tak i všech současně. Navíc mají být diody ovládány z procesoru. Na základě těchto požadavků bylo navrženo schéma regulátoru znázorněné na obr. 4.5.



Obr. 4.5 – Schéma budiče laserových diod

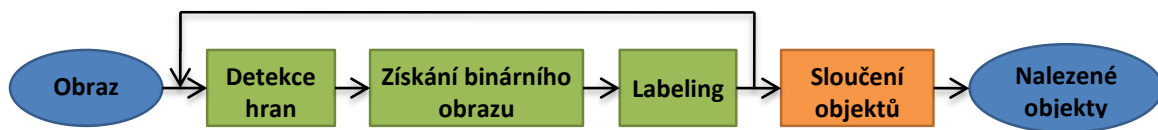
Pro měření světelného toku diod slouží operační zesilovač U1A zapojený jako U-I převodník s proměnným zesílením nastavitelným pomocí R2. Tím lze regulovat jas jednotlivých diod. Výstup z U-I převodníku je společně s REF signálem, z nastavitelného odporového děliče, přiveden do U1B zapojeného jako servo, které ovládá tranzistor Q2 spínající laserovou diodu. REF signál je společný pro všechny 3 diody a nastavuje se pomocí něj jas všech diod najednou. Ovládání diody z procesoru obstarává tranzistor Q1. Navržená regulační smyčka má snahu kmitat, proto je spínání laserové diody blokováno kondenzátorem C1.

Výše uvedené schéma je zopakováno 3x pro všechny laserové diody a doplněno o další části jako stabilizátor napětí a indikační led. Celé schéma včetně návrhu plošného spoje se nachází v příloze B.

## 5 Implementace algoritmů do DSP

### 5.1 Zpracování stínových obrazů

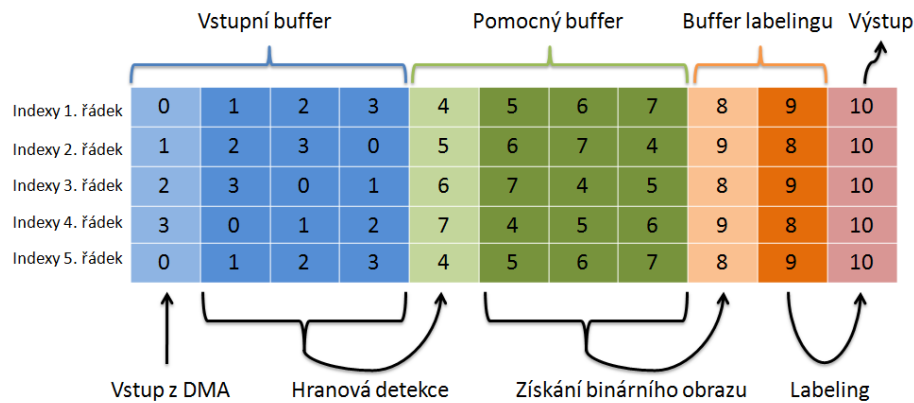
Hlavním úkolem algoritmu pro zpracování stínových obrazů je najít v obraze všechny objekty a následně určit jejich parametry. Z metod pro zpracování stínových obrazů popsaných v teoretické části byl navržen algoritmus, který je znázorněn na obr. 5.1. Na začátku vstupují data postupně po řádcích, tak jak jsou čtena ze senzoru. V hlavní smyčce algoritmu, naznačené zelenými obdélníky, je obraz zpracován a po dokončení celého snímku jsou nalezené objekty sloučeny, vyhodnoceny jejich parametry a odeslány.



Obr. 5.1 – Znázornění algoritmu pro zpracování stínových obrazů

#### 5.1.1 Rozložení paměti

Všechny 3 hlavní části zpracování obrazu uvedené výše potřebují svou vlastní část paměti pro uložení dat, nad kterými pracují. Nejvíce místa, 3 řádky, potřebuje detekce hran pro provedení konvoluce. Navíc musí být vyhrazen další řádek pro DMA kanál, který načítá nová data. Z tohoto důvodu byla paměť rozdělena do několika kruhových bufferů, jejichž řazení je na obr. 5.2



Obr. 5.2 – Znázornění rozdělení paměti v průběhu zpracování obrazu

### 5.1.2 Detekce hran

Objekty stínových obrazů se vyznačují výraznými hranami, proto jako první krok při jejich zpracování byla zvolena hranová detekce. Za účelem dosažení co nejvyšší rychlosti byla zvolena nejjednodušší konvoluční maska a sice  $[1, 0, -1]$ . Tato maska je na data aplikována ve dvou směrech a je vložena přímo do kódu. Tím se výpočet konvoluce, potřebné pro hranovou detekci, zjednodušil na několik rozdílů resp. součtů. Pro výpočet absolutní hodnoty je použita vnitřní funkce procesoru. Ukázka kódu výpočtu konvoluce pro detekci hran ve dvou směrech:

```
EdgeImage[x][y]=Abs(Image[x-1][y]-Image[x+1][y])
                +Abs(Image[x][y-1]-Image[x][y+1]);
```

### 5.1.3 Získání binárního obrazu

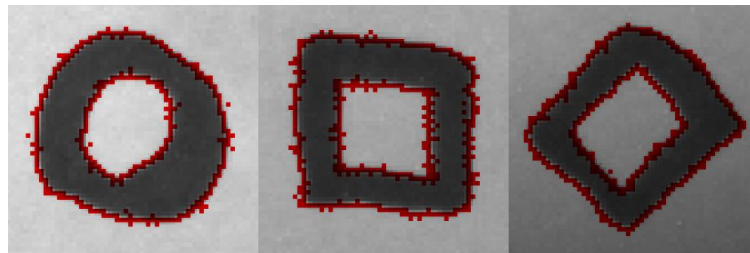
K dalšímu zpracování je nutné nalezené objekty s výraznými hranami jednoznačně odlišit od pozadí. Toho lze dosáhnout vytvořením binárního obrazu. Pro získání binárního obrazu s tenkými a dobře lokalizovanými hranami je použita metoda non-maxima-suppression [4], která z obrazu se získanými hranami vybere pouze tu část hrany, která je lokálním maximem. Hledání lokálního maxima probíhá ve 2 hlavních směrech a ve 2 diagonálních směrech. Kód pro nalezení maxima hrany v jednom směru:

```

if EdgeImage[x][y]>Threshold &&
    EdgeImage[x][y]>=EdgeImage[x+1][y] &&
    EdgeImage[x][y]>=EdgeImage[x-1][y] then
        BinImage[x][y]=255;
    else
        BinImage[x][y]=0;
    end if

```

Výstup z algoritmu detekce hran a následné lokalizace hran pomocí non-maxima-suppression je k vidění na obr. 5.3

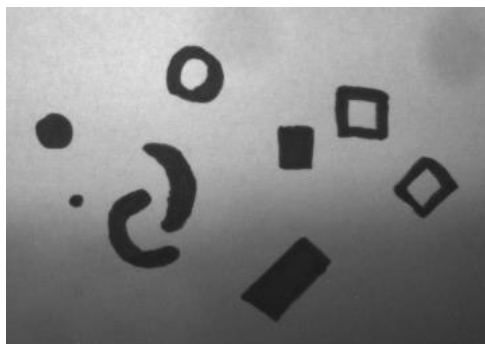


Obr. 5.3 – Promítnutí detekovaných hran do původního obrazu

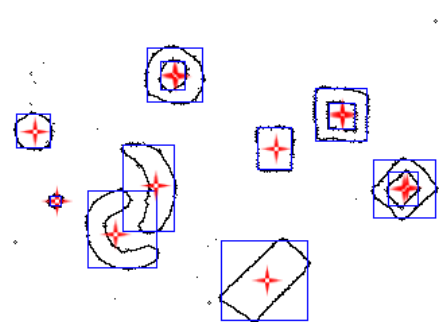
#### 5.1.4 Labeling

Základní algoritmus labelingu popsany v teoretické části neobsahuje metody určování parametrů objektů. K uchování parametrů jednotlivých objektů je vytvořena datová struktura uchovávající plochu, souřadnice těžiště a souřadnice ohraničujícího obdélníku. Tyto parametry se plní v průběhu přiřazování značek k pixelům společně s vytvářením tabulky ekvivalentních značek. Na konci celého procesu, je tabulka ekvivalentních značek zpracována a provedeno sloučení objektů. Přesný výpočet plochy a těžiště je proveden až poté.

Výše popsany algoritmus byl vyzkoušen na reálných datech. Na obr. 5.5 a obr. 5.4 je k vidění vstup a výstup z algoritmu.



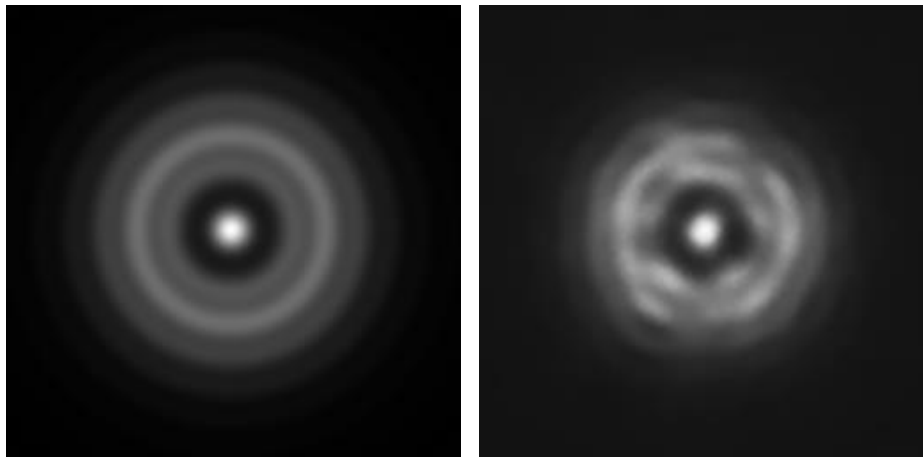
Obr. 5.5 – Vstupní data



Obr. 5.4 – Výstup z algoritmu

## 5.2 Zpracování jasových značek

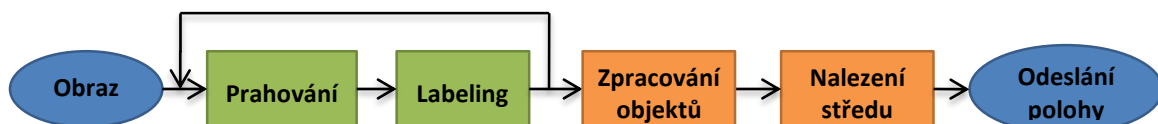
Reálný difrakční obrazec promítnutý na senzor a teoreticky vypočítaný difrakční obrazec je vidět na obrázku obr. 5.6 resp. obr. 5.7. Podrobnější měření difrakčních obrazců potvrzující teoretický výpočet v kapitole 2.2.1.2 je k vidění ve fotopříloze A.



Obr. 5.6 – Teoreticky vypočítaný obrazec

Obr. 5.7 – Reálný difrakční obrazec

Pro úlohu určení polohy jasových značek byly všechny algoritmy nejdříve vyzkoušeny v Matlabu způsobem co nejvíce podobnému následnému programování v jazyce C, použitým v procesoru, a pak teprve implementovány do procesoru. Navržený algoritmus pro určení polohy difrakčního obrazce je k vidění na obr. 5.8.



Obr. 5.8 – Algoritmus zpracování jasových značek

### 5.2.1 Expoziční doba a nastavení prahování

Pro uvažovaný systém platí konstantní expoziční podmínky nezávislé na okolním osvětlení. Navíc intenzitu všech laserů je možné nastavit na stejnou úroveň. CMOS senzor tedy může mít nastavenou konstantní dobu expozice. Nesmí ale docházet k přeexpozici, která by zapříčinila špatné výsledky aproximací. Díky homogenním světelným podmínkám je též možné nastavit konstantní prahovací úroveň, která by měla být co nejnižší, aby střed difrakčního obrazce měl co nejvíce



bodů a dal se tak lépe aproximovat, ale na druhou stranu musí být dost vysoká, aby střed obrazce byl jednoznačně oddělen od kruhu kolem něj.

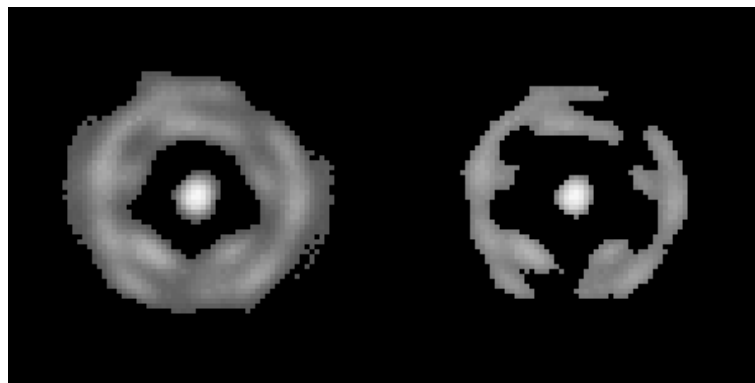
Za účelem dosažení co nejvyšší rychlosti zpracování byla metoda prahování implementována do procesoru pomocí look-up tabulky. Pseudokód pro prahování vypadá následovně:

```
ThresholdedImage[x][y]=Threshold[Image[x][y]];
```

### 5.2.2 Labeling a zpracování objektů

Oproti předchozí implementaci labelingu, kdy stačovalo pamatovat si pouze několik integrálních veličin, je v tomto případě potřeba pamatovat si z obrazu více informací. O každém pixelu, který patří objektu, je potřeba uchovat informaci o jas, poloze a přiřazené značce, což zabírá více paměti.

Na základě takto provedeného labelingu jsou sloučeny objekty s ekvivalentními značkami a pro každý objekt je určena plocha a vzdálenost od středu celého obrazce. Objekt, který má požadovanou plochu a je nejbližší středu, je prohlášen za objekt zájmu a bere se jako vstupní data pro aproximace. Toto určení prostředního objektu pracuje i v případě, kdy je nastaven příliš vysoký práh a obrazec se rozpadne na více částí. Tento případ je na obr. 5.9 vpravo a správný výstup vlevo.

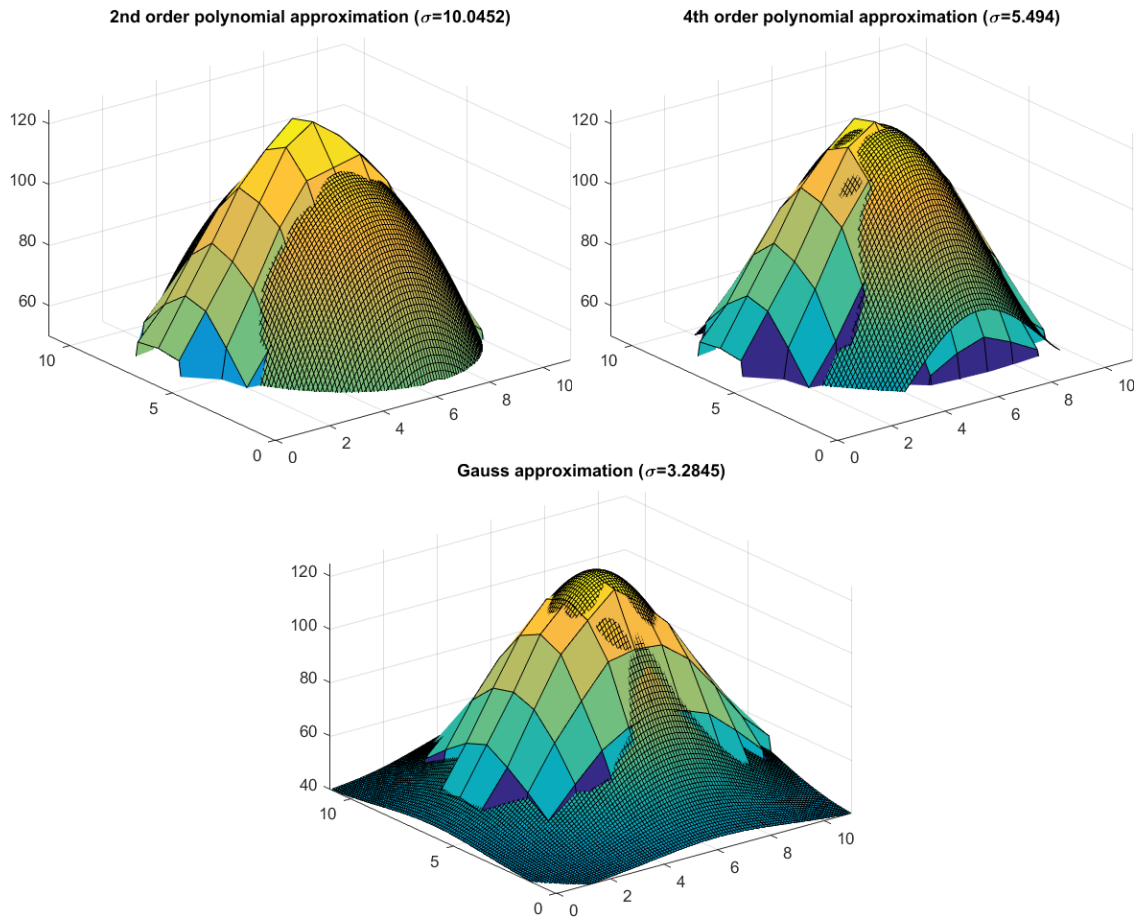


Obr. 5.9 – Ukázka dat pro následné určení středu

### 5.2.3 Aproximace obrazce a nalezení středu

Z teoretického rozboru vyplývá, že difrakční obrazce vycházejí z funkce  $\sin(x)/x$  avšak střed zvoleného obrazce je podobný Gaussově funkci. S výhodou tedy lze použít aproximační metody, které jsou schopny určit střed obrazce přesněji než jednoduché hledání maxima. Reálně navíc nasnímaný střed difrakčního obrazce obsahuje velmi málo pixelů, je tedy vhodné implementovat algoritmy aproximací pro co nejpřesnější nalezení jeho středu.

Do procesoru byly implementovány metody aproximace nejen pomocí Gaussovy křivky, ale také pomocí polynomů prováděné metodou nejmenších čtverců. Pro výpočet této metody je nutné použít maticový počet. Maticové násobení a transpozici matic má již procesor implementovány v knihovnách, takže jediné, co bylo implementováno, je metoda pro výpočet soustavy rovnic. Řešení soustavy rovnic je prováděno Gaussovou eliminační metodou, po jejímž provedení zůstane matice v horním trojúhelníkovém tvaru, kde na diagonále jsou hodnoty 1. Poté již je možné lehce dopočítat řešení pravé strany. Výsledné řešení se shoduje s výsledky v Matlabu na 5 desetinných míst i přes to, že procesor má přesnost *double* pouze  $2 \times 16$  – bit. Pro názornost jsou na obr. 5.10 vidět simulace aproximovaných funkcí spolu s reálnými obrazovými daty a směrodatnou odchylkou od reálných dat.



Obr. 5.10 – Porovnání aproximací (jemné struktury) a reálných dat

Jak již bylo zmíněno výše, při výpočtu aproximací je použit maticový počet. Pro získání  $n$  parametrů aproximace, musí mít střed obrazce plochu alespoň  $n$  pixelů, jinak jsou získané matice singulární. Pro aproximaci pomocí polynomu 4. řádu je počet parametrů  $n = 15$ . Pro výpočet aproximace pomocí Gaussovy křivky je potřebný počet parametrů  $n = 5$ , avšak kvůli metodě výpočtu pomocí iterací je potřebná plocha středu cca 50 *pix* v závislosti na datech. Výpočet aproximací tedy občas může selhat, proto bylo implementováno i nalezení těžiště s odseknutím minimální hodnoty, které bylo popsáno v teoretické části 2.1.6 a které funguje vždy.

### 5.2.4 Rychlost implementovaných algoritmů

Všechny uvedené algoritmy byly implementovány s ohledem na rychlost zpracování. Z hlediska metodiky porovnání rychlostí v jednotkách počtu operací na pixel (*op/pix*) zavedené v práci [5] vychází má implementace labelingu s rychlostí

40 *op/pix* pomalejší než implementace v rámci práce [5], kde bylo dosaženo rychlosti 28 *op/pix*. Obě implementace labelingu by měly být shodné, co se týká rozhodovacího stromu a struktury union-find, takže rozdíl bude vznikat zřejmě kvůli rozdílným datovým typům a v počtu informací, které se v průběhu labelingu uchovávají. Pro analýzu rozdílů mezi implementacemi bych musel mít přístup k originálním zdrojovým kódům z práce [5]. I přes pomalejší implementaci labelingu bylo dosaženo čtecí rychlosti z CMOS senzoru 15 *MHz*, což při rozlišení 752 x 480 *pix* odpovídá snímkové frekvenci asi 40 snímků za sekundu.

Dalším na co byla zaměřena pozornost, je implementace matematických aproximací. V laboratoři videometrie se zatím bohužel touto úlohou nikdo nezabýval, proto nelze provést srovnání s různými implementacemi. Pro zajímavost uvádím tabulku porovnávající rychlost výpočtu v programu Matlab běžícím na procesoru s dvěma jádry a frekvencí 2,4 *GHz* s rychlostí výpočtu v procesoru Blackfin s frekvencí hodin 600 *MHz*.

	<i>Polynom 2. řádu (ms)</i>	<i>Polynom 4. řádu (ms)</i>	<i>Gaussova funkce (ms)</i>	<i>Výpočet těžiště (ms)</i>
<i>Matlab</i>	1,8	18,9	14,8	0,3
<i>Blackfin</i>	3,9	21,0	52,2	0,1

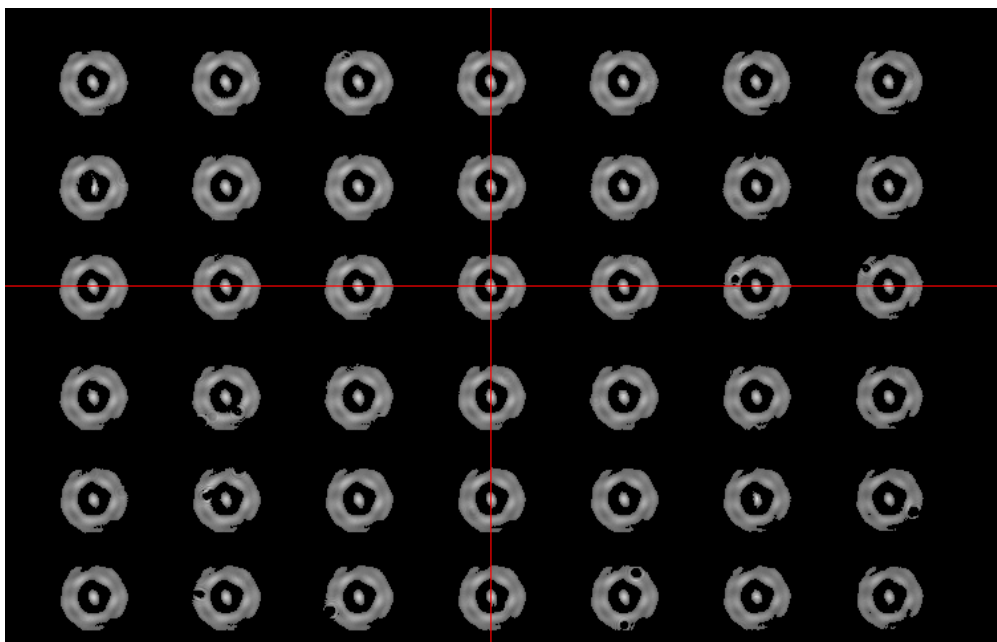
Tab. 5.1 – Porovnání rychlostí aproximací

## 6 Zhodnocení dosažených výsledků

Pro zjištění dosažené přesnosti bylo na polohovacím systému provedeno několik měření popsaných níže. Všechny rozměry jsou uváděny v pixelech místo v mikrometrech, aby bylo zřejmé, zdali naměřené nepřesnosti odpovídají periodě pixelů nebo ne. Jeden pixel odpovídá vzdálenosti  $5 \mu\text{m}$  v souřadnicích pojezdů.

### 6.1 Měření homogenity osvětlení

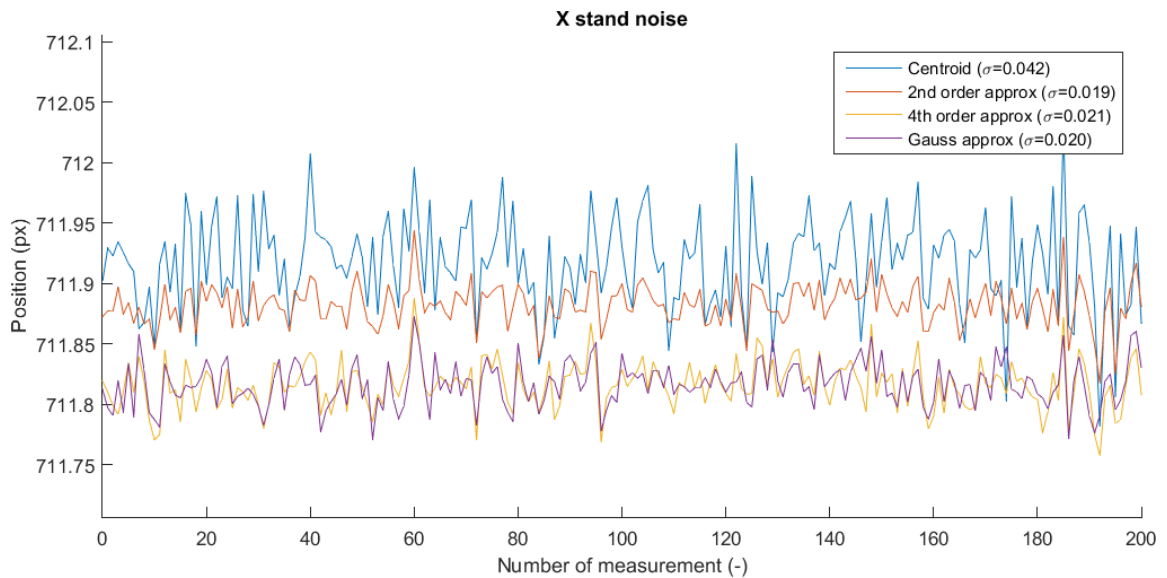
Aby všechny aproximace poskytovaly dobré výsledky, musí být CMOS senzor homogenně nasvícen po celé ploše. Složením snímků vždy s jedním difrakčním obrazcem v různých polohách byla vytvořena kompozice znázorněná na obr. 6.1, na kterém je patrné, že ve všech bodech vzniká na senzoru stejný obrazec se stejnou intenzitou. Metody aproximací tedy budou fungovat správně ve všech bodech senzoru. Na obrázku jsou též vidět nečistoty na senzoru způsobující černá místa v obrazcích, kolem kterých vznikají další nežádoucí difrakční obrazce.



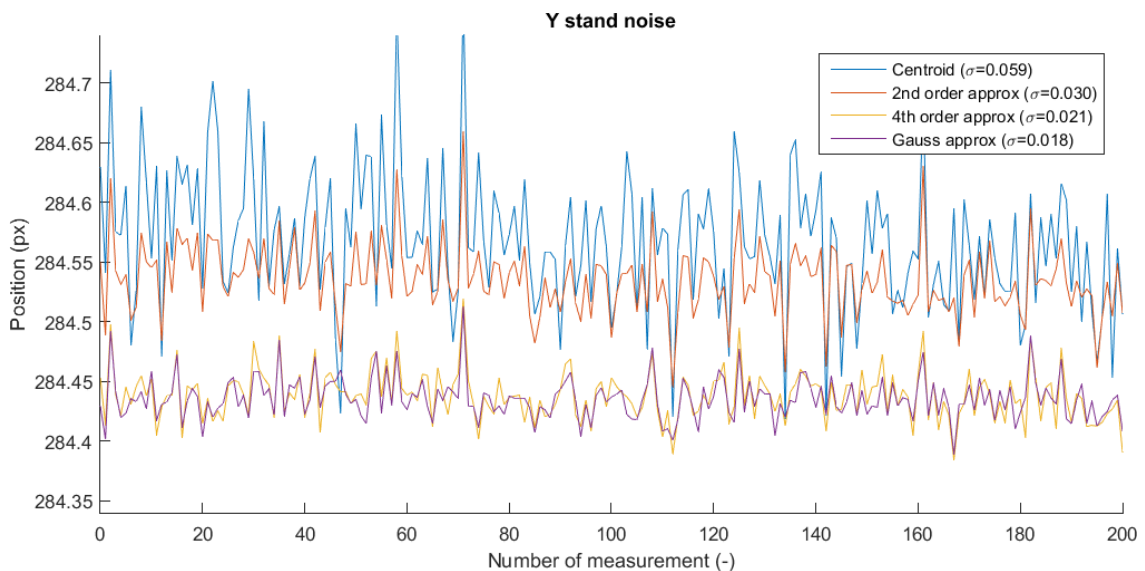
Obr. 6.1 – Měření homogenity osvětlení a znázornění kolmosti

## 6.2 Měření opakovatelnosti

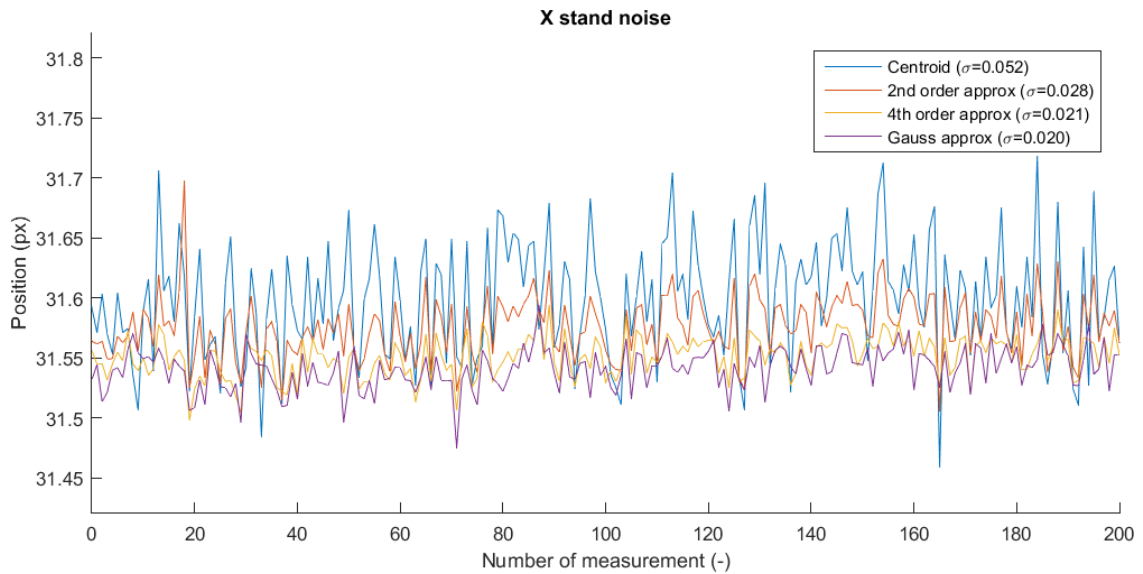
K porovnání šumů u jednotlivých aproximací byla provedena dvě měření na opačných koncích senzoru. Při obou těchto měřeních byly pojezdy v klidu. Výsledky jsou znázorněny na obr. 6.2 až obr. 6.5. Pro porovnání je u jednotlivých aproximací uvedena i směrodatná odchylka při daném měření.



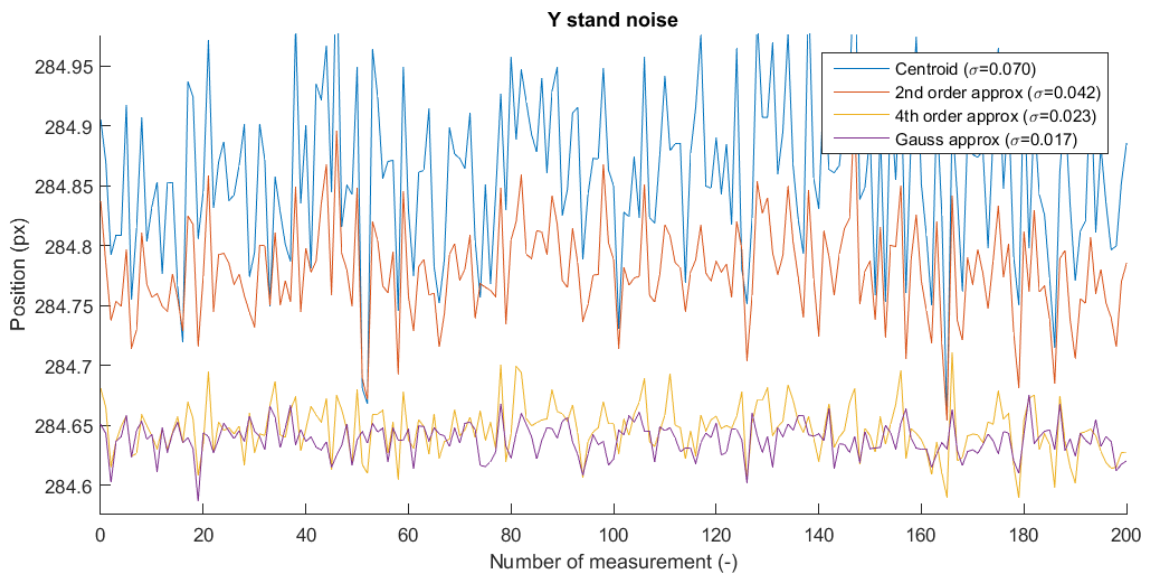
Obr. 6.2 – Měření opakovatelnosti (měření 1, osa X)



Obr. 6.3 – Měření opakovatelnosti (měření 1, osa Y)



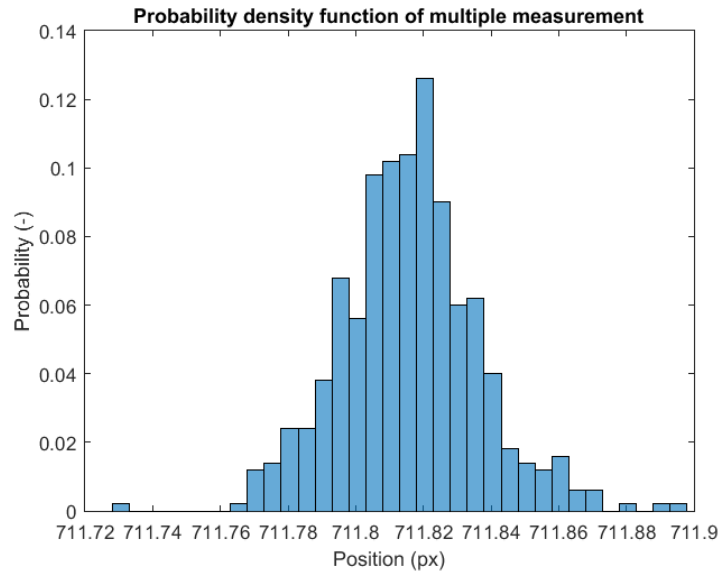
Obr. 6.4 – Měření opakovatelnosti (měření 2, osa X)



Obr. 6.5 – Měření opakovatelnosti (měření 2, osa Y)

Z naměřených průběhů je patrné, že aproximace Gaussovou funkcí má nejmenší směrodatnou odchylku  $\sigma = 0,02$ . Tento výsledek byl očekáván, a proto bude další měření prováděno pouze s touto aproximací. Z rozdílných středních hodnot jednotlivých aproximací lze usuzovat, že alespoň některé z nich jsou náchylné k chybám způsobeným polohou jasové značky v rámci periodicity pixelů.

Pro úplnost je na obr. 6.6 zobrazeno rozdělení pravděpodobnosti pro aproximaci Gaussovou funkcí, na kterém je vidět, že téměř všechna měření spadají do oblasti široké 0,1 *pix*.



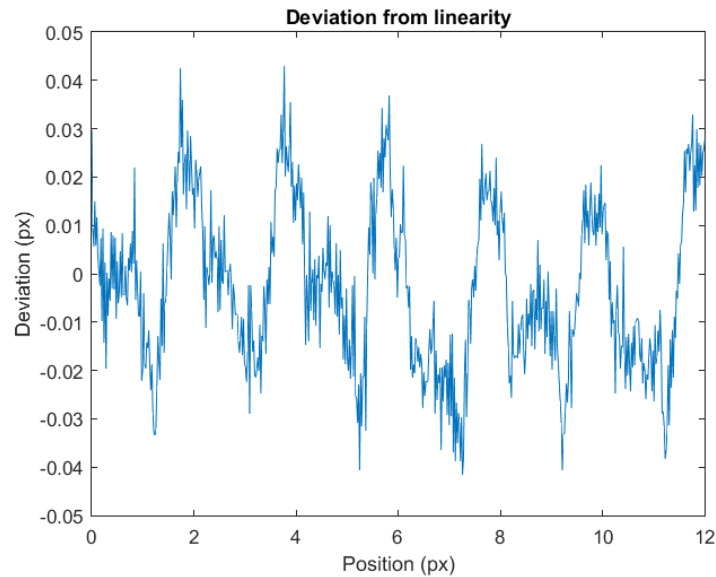
Obr. 6.6 – Rozdělení pravděpodobnosti pro Gaussovou aproximaci

### 6.3 Odchylka od linearity

Měření odchylky od linearity bylo provedeno pouze v jednom směru, protože pojezd použitý na polohování v druhém směru byl v době měření nefunkční a nepodařilo se jej opravit.

I přes to, že měření probíhalo v jedné ose, bylo dosti problematické, protože dosažená přesnost určení polohy je srovnatelná s přesností pojezdu. Pojezd navíc občas najel na hodnotu jinou než zadanou. Z tohoto důvodu bylo měření odchylky od linearity provedeno několikrát s tím, že před každým měřením byl vypnut motor pojezdu i regulace polohy, aby nedocházelo k dalším zásahům. Poloha vyčtená z pojezdu v takto ustáleném stavu byla považována za přesnou. Po několikerém měření byly získány všechny hodnoty polohy, jež je pojezd schopen rozlišit. Pro zpřesnění byl při každém měření vypočítán průměr z 10 odběrů. Výslednou odchylku od linearity složenou ze všech možných poloh pojezdu lze vidět na obr. 6.7.



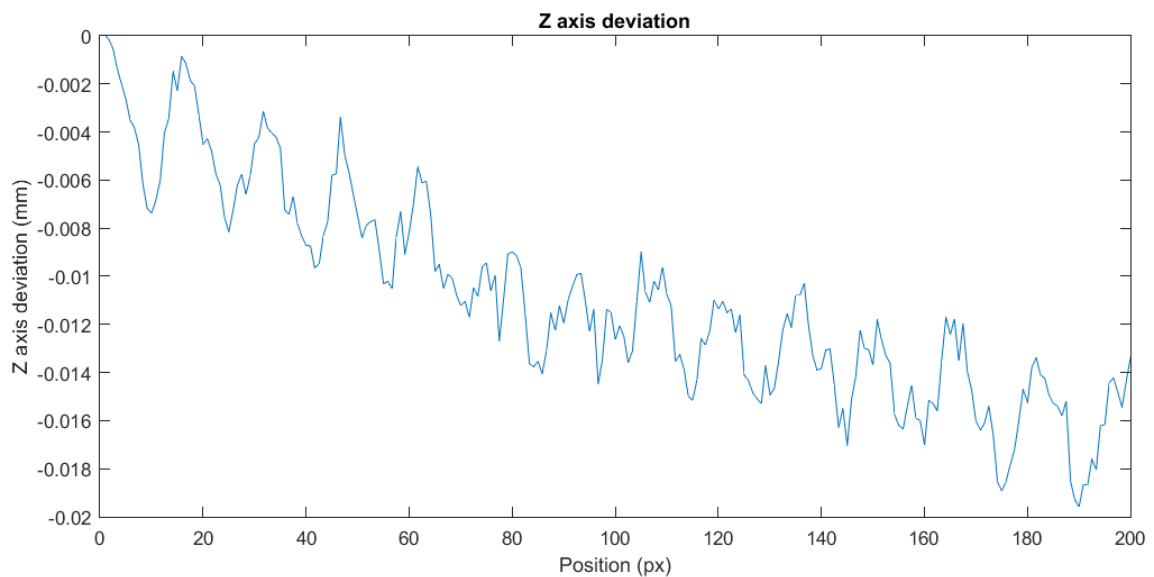


Obr. 6.7 – Změřená odchylka od linearity

Z průběhu je patrné, že měření je zatíženo chybou, která se opakuje s periodou 2 pixely. Přesná periodicitu naznačuje, že chyba vzniká při určení polohy jasové značky na CMOS senzoru a ne chybou pojezdu. Jedním z možných vysvětlení je, že použitý CMOS sensor sám o sobě obsahuje tuto periodickou chybu. Zvolený sensor existuje i v barevné variantě, ve které jsou před pixely vloženy barevné filtry podle Bayerova vzoru, který obsahuje právě periodu 2 *pix*. Lze předpokládat, že výrobce nebude pro barevný typ senzoru vyrábět rozdílný křemíkový čip, takže se případné rozdílnosti v pixelech mohou dostat i do černobílé varianty. Bohužel nejsou k dispozici žádné informace týkající se přesné struktury pixelů, proto je velmi těžké určit, kde chyba vzniká. Za účelem nalezení těchto nepřesností byl sensor osvětlen různě barevným světlem s nadějí, že každý pixel bude mít lehce jinou spektrální charakteristiku. To se však neprokázalo. Též byla naměřena šumová charakteristika jednotlivých pixelů, která také neukázala žádnou periodicitu. Uvedenou chybu se tedy zatím nepodařilo přesně analyzovat.

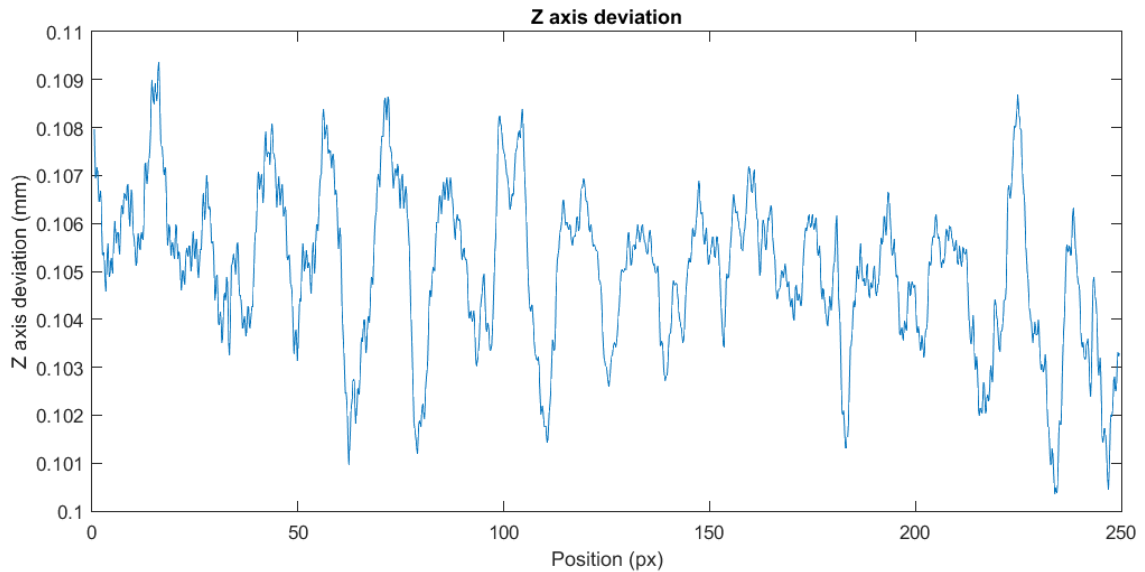
## 6.4 Měření ve směru normály

Díky více laserovým diodám lze měřit změnu zvětšení optické soustavy. Ze změny tohoto zvětšení je dále možné určit změnu vzdálenosti od senzoru. Vzorec pro výpočet vzdálenosti od senzoru, ze znalosti polohy difrakčních obrazců, je uveden v kapitole 4.3. Z téhož vzorce vyplývá přesnost určení polohy. Při dané přesnosti určení polohy obrazů s rozlišením  $0,5 \mu\text{m}$  je výsledná přesnost měření v ose Z  $5,9 \mu\text{m}$ . První měření odchylky v ose Z při pohybu v ose X je vidět na obr. 6.8.



Obr. 6.8 – Měření změny vzdálenosti clonky od senzoru

Měření znázorněné na obr. 6.8 ukazuje, že při pohybu v ose X o  $200 \text{ pix}$  ( $1 \text{ mm}$ ) došlo ke změně vzdálenosti o  $16 \mu\text{m}$  z čehož vyplývá, že rovina pojezdů není rovnoběžná s rovinou CMOS senzoru. Úhlová odchylka činíc cca  $0^{\circ}55'$  vznikla nepozorností při sestavování mechanické konstrukce. Nechtěně se tím podařilo ověřit citlivost systému na vzdálenost clonky od senzoru. Díky této vlastnosti lze do měřicího systému implementovat autodiagnostiku, která sama pozná, kdy systém měří ve správném pracovním rozsahu vzdálenosti clonky od senzoru. Opakované měření po odstranění chyby je na obr. 6.9



Obr. 6.9 – Měření změny vzdálenosti clonky od senzoru po opravě

Na naměřených datech po opravě je patrné zlepšení odchylky v ose Z, ale stále se objevuje periodická chyba s periodou asi 15 *pix*. Tato chyba zřejmě vzniká periodickou chybou zjištěnou při měření linearity, je však výraznější a nemá konstantní amplitudu, zřejmě proto, že vzniká navzorkováním dvou periodických signálů s mírně odlišnou periodou. Odlišnost těchto signálů je způsobena právě pohybem v ose Z.

## 6.5 Návrhy na další vývoj

Mikropolohovací systém popsany v práci splnil očekávání přesnosti, avšak je zde několik faktorů, které nepříznivě ovlivňují výslednou přesnost měření. Jedná se zejména o mechaniku a přesnost referenčního polohování. Pro podrobnější analýzu chyb a případný další vývoj by bylo vhodné vytvořit preciznější mechanickou část a referenční odměřování provádět například pomocí interferometrů.

Dále by bylo vhodné se znovu zaměřit na zpracování difrakčního obrazce a najít lepší popis jeho středu než pomocí Gaussovy křivky. Aproximace obrazce pomocí Gaussovy křivky je navíc prováděna pouze z několika bodů, což též omezuje přesnost. Při dostatku paměti pro uchování celého obrazce by však bylo možné jej aproximovat v širším okolí a určit tak jeho polohu přesněji.

## 7 Závěr

Hlavním cílem práce bylo ověřit metodu přesného mikropolohování navrženého systému pomocí určení polohy jasové značky v obraze. Tato jasová značka byla získávána pomocí nekolimovaného laserového paprsku svítícího přímo na CMOS senzor. Díky koherenci laserových paprsků vznikl na senzoru difrakční obrazec, ze kterého lze určit polohu v rovině rovnoběžné s plošným CMOS senzorem. Pomocí více laserových diod v přesné vzdálenosti od sebe byla určována i změna vzdálenosti polohovacího systému od senzoru.

Pro získání představy o tvaru difrakčních obrazců byl na začátku práce proveden rozbor difrakčních jevů vznikajících na šěrbině. Tyto jevy jsou popsány pomocí Fresnelova integrálu, který se podařilo pomocí numerických metod vypočítat a určit tak vzdálenost, do které má být umístěn senzor.

Na základě znalosti tvaru difrakčního obrazce vznikajícího na senzoru byly analyzovány metody průběžného zpracování obrazu vhodné pro zpracování jasových značek. Pro úplnost byly prozkoumány i metody určení parametrů objektů vznikajících promítáním stínových obrazů. Z tohoto rozboru se ukázalo jako výhodné použít metodu connected-component-labeling i pro zpracování obrazu s jasovými značkami. Za účelem získání co nejlepšího určení polohy jasové značky byl diskutován tvar středu difrakčního obrazce a provedeny aproximace pomocí polynomů 2. a 4. řádu a pomocí Gaussovy funkce dvou proměnných. Pro výpočet aproximací byla použita lineární i nelineární metoda nejmenších čtverců.

Metody zpracování obrazu byly pro ověření jejich funkčnosti nejprve simulovány v prostředí Matlab a poté implementovány do signálového procesoru. Všechny metody kromě aproximací pracují na principu rychlého průběžného zpracování. Díky tomuto principu a dalším optimalizacím bylo dosaženo snímkové frekvence určení polohy jasové značky 6 snímků/sekundu.

Dále byl navržen systém mikropolohování, který měl za úkol ověření funkčnosti metody polohování pomocí promítnuté jasové značky. Výsledný systém je schopen určit absolutní polohu v rozsahu jednotek milimetrů s přesností na  $500\text{ nm}$  v rovině senzoru a  $6\text{ }\mu\text{m}$  ve směru normály k senzoru. Dosažená přesnost je na hranici rozlišitelnosti referenčního odměřování polohy. Systém tak splnil očekávání a dá se předpokládat, že s přesnějším referenčním odměřováním by se dalo dosáhnout vyšších přesností.

## 8 Seznam Použitých zkratek

ADC	Analog to digital converter
ALU	Arithmetic logic unit
CCD	Coupled charge device
CMOS	Complementary metal oxide semiconductor
CNC	Computer numerical control
DC	Direct current
DMA	Direct memory access
GPIO	General-purpose input/output
HS USB	High speed - Universal serial bus
I2C	Inter-Integrated Circuit
LED	Light emitting diode
LoG	Laplacian of Gaussian
PID	Proportional-integral-derivative
PPI	Parallel peripheral interface
PSD	Position-sensitive diode
PWM	Pulse width modulation
UART	Universal asynchronous receiver/transmitter
WVGA	Wide video Graphics Array

## 9 Seznam použitých symbolů

$a$	Charakteristický rozměr štěrbiny
$a$	Vzdálenost předmětu
$a'$	Vzdálenost obrazu
$d$	Posun stínítka
$d'$	Posun obrazce na senzoru
$E$	Intenzita záření
$e$	odchylka
$\mathcal{F}$	Fourierova transformace
$F$	Fresnelovo číslo
$f$	Aproximovaná funkce
$\hat{f}$	Aproximace
$h$	Konvoluční jádro
$H$	Obraz konvolučního jádra ve Fourierově transformaci
$I$	Intenzita ozáření
$I$	Hodnota obrazové funkce
$J$	Jakobiho matice
$k$	Poměr pro převod souřadnic
$\lambda$	Vlnová délka
$L$	Vzdálenost štěrbiny os stínítka
$M$	Hmotnost obrazové stopy

$\varphi$	Vlnoplocha
$\mathbf{p}$	Vektor parametrů aproximací
$p_n$	Parametry aproximací
$\mathbf{Q}$	Matice pro výpočet nejmenších čtverců
$r$	Poloměr vlnoplochy
$\sigma$	Směrodatná odchylka
$S$	Minimalizační kritérium
$S$	Plocha objektu
$\tau$	Integrační proměnná času
$t$	Čas
$x, y$	Souřadnice v rovině senzoru
$X_C, Y_C$	Souřadnice těžiště obrazové stopy
$X_T, Y_T$	Souřadnice středu obrazové stopy
$X_{TC}, Y_{TC}$	Souřadnice těžiště obrazové stopy s odečtením minima
$y$	Velikost předmětu
$y'$	Velikost obrazu
$z$	Směr normály k senzoru



## 10 Seznam obrázků a tabulek

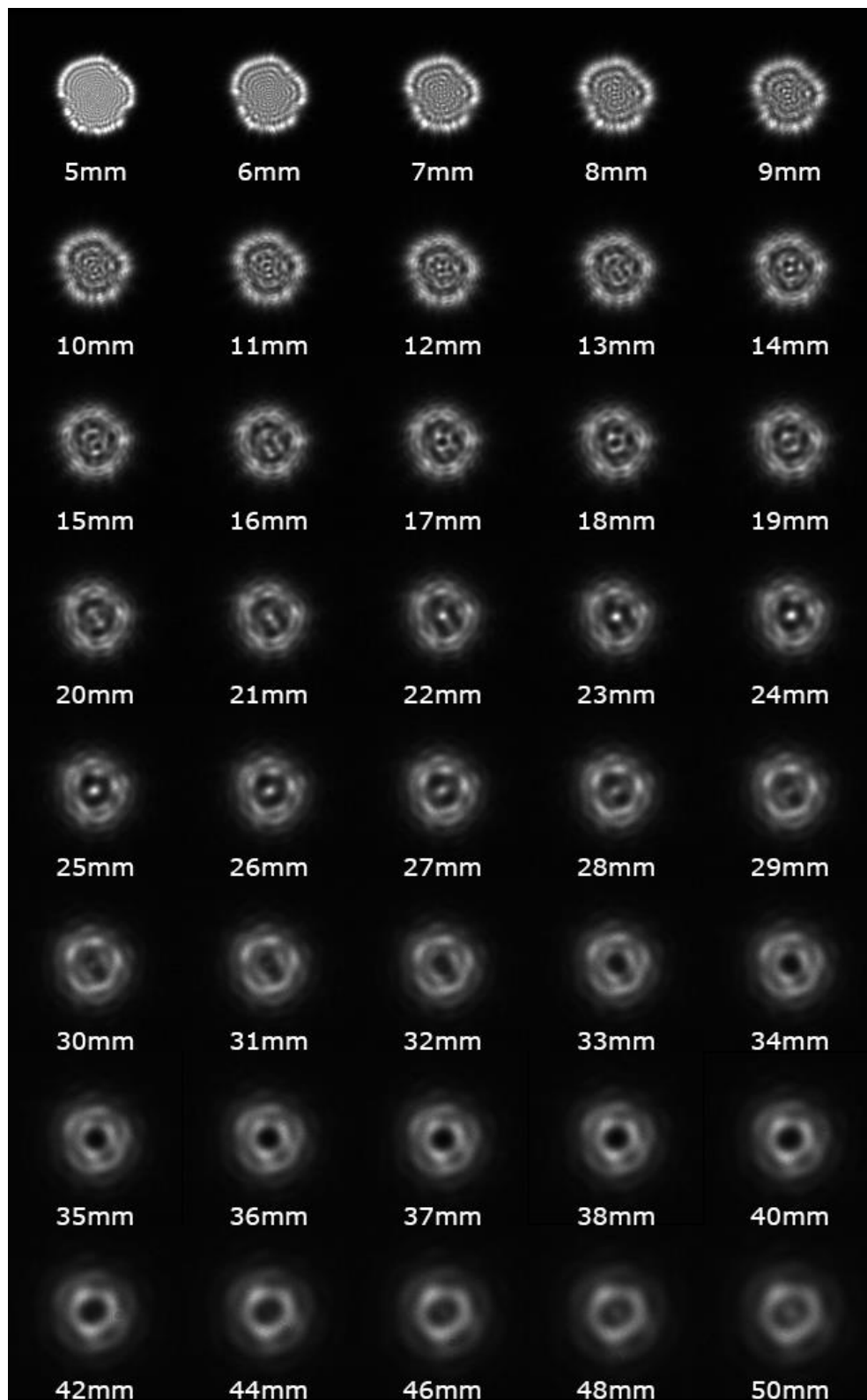
Obr. 2.1 – Příklad obrazu se stínovými objekty.....	3
Obr. 2.2 – Zobrazení hran různými metodami.....	5
Obr. 2.3 – Detekované hrany.....	5
Obr. 2.4 – Vstupní data .....	5
Obr. 2.5 – vlevo: vstup, uprostřed: prahování hran, vpravo: nalezení maxima.....	6
Obr. 2.6 – Znázornění problémových situací při segmentaci.....	7
Obr. 2.7 – Prohledávané asymetrické 4-okolí [5].....	8
Obr. 2.8 – Rozhodovací strom labelingu [5].....	8
Obr. 2.9 – Příklad difrakčního obrazce .....	10
Obr. 2.10 – Znázornění Huygens–Fresnelova principu .....	10
Obr. 2.11 – Znázornění geometrického uspořádání .....	11
Obr. 2.12 – Průřez difrakčním obrazcem.....	13
Obr. 2.13 – Jasový průřez středem difrakčního obrazce .....	14
Obr. 2.14 – Difrakční obrazec .....	14
Obr. 2.15 – Střed difrakčního obrazce.....	15
Obr. 2.16 – Porovnání různých aproximací .....	19
Obr. 3.1 – Ideový nákres systému .....	20
Obr. 4.1 – Nákres konstrukce držící CMOS senzor a lasery.....	25
Obr. 4.2 – Znázornění systému pohledem geometrické optiky .....	27
Obr. 4.3 – Schéma laserové diody.....	28
Obr. 4.4 – Pouzdro laserové diody .....	28
Obr. 4.5 – Schéma budiče laserových diod .....	29
Obr. 5.1 – Znázornění algoritmu pro zpracování stínových obrazů.....	30
Obr. 5.2 – Znázornění rozdělení paměti v průběhu zpracování obrazu .....	31
Obr. 5.3 – Promítnutí detekovaných hran do původního obrazu .....	32

Obr. 5.4 – Výstup z algoritmu .....	32
Obr. 5.5 – Vstupní data .....	32
Obr. 5.6 – Teoreticky vypočítaný obrazec .....	33
Obr. 5.7 – Reálný difrakční obrazec .....	33
Obr. 5.8 – Algoritmus zpracování jasových značek.....	33
Obr. 5.9 – Ukázka dat pro následné určení středu.....	34
Obr. 5.10 – Porovnání aproximací (jemné struktury) a reálných dat .....	36
Obr. 6.1 – Měření homogenity osvětlení a znázornění kolmosti .....	38
Obr. 6.2 – Měření opakovatelnosti (měření 1, osa X).....	39
Obr. 6.3 – Měření opakovatelnosti (měření 1, osa Y).....	39
Obr. 6.4 – Měření opakovatelnosti (měření 2, osa X).....	40
Obr. 6.5 – Měření opakovatelnosti (měření 2, osa Y).....	40
Obr. 6.6 – Rozdělení pravděpodobnosti pro Gaussovu aproximaci .....	41
Obr. 6.7 – Změřená odchylka od linearity.....	42
Obr. 6.8 – Měření změny vzdálenosti clonky od senzoru.....	43
Obr. 6.9 – Měření změny vzdálenosti clonky od senzoru po opravě.....	44
Obr. A.1 – Reálné tvary difrakčních obrazců v různých vzdálenostech .....	54
Obr. B.2 – Výkres DPS pro budič laserů - vrstva BOT .....	55
Obr. B.3 – Osazovací výkres pro budič laserů.....	58
Obr. C.4 – Celkový pohled na mikropolohovací systém .....	60
Obr. C.5 – Pohled na polohovací systém z boku.....	60
Obr. C.6 – Pohled na systém z vrchu .....	61
Obr. C.7 – Detail odměřovací soustavy .....	61
Obr. C.8 – Detail uložení CMOS senzoru.....	62
Obr. C.9 – Detail umístění laserů.....	62
Obr. C.10 – Detail clonky s prosvítajícím světlem bílé LED.....	62
Tab. 5.1 – Porovnání rychlostí aproximací.....	37
Tab. B.1 – Soupis materiálu pro budič laserů .....	58
Tab. D.2 – Obsah příloženého CD .....	63

## 11 Bibliografie

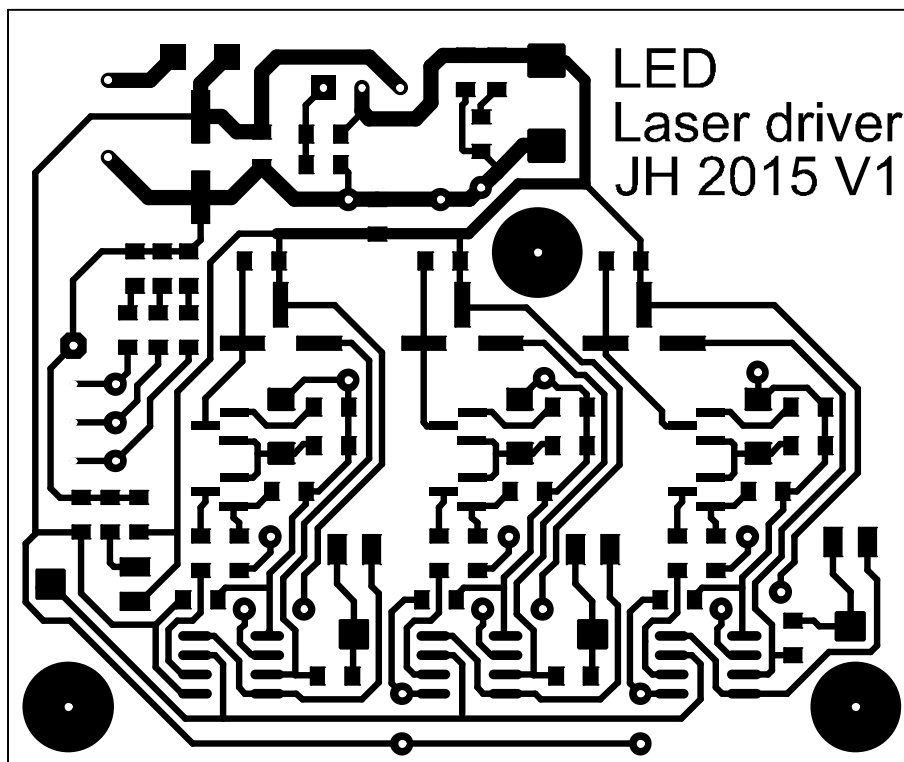
- [1] N.B., Watson, J.D. Jones, *Digital signal processing : Principles, devices and applications*. London: Peter Peregrinus Ltd, 1990.
- [2] R. G. Keys, *Cubic convolution interpolation for digital image processing.*: IEEE, 1981.
- [3] Kasturi R., Schunck B.G. Jain R., *Machine Vision.*: McGraw Hill, 1995.
- [4] Gool L. V. Neubeck A., *Efficient Non-Maximum Suppression*. Zurich: IEEE, 2006.
- [5] A. Michálek, *Počítání a třídění objektů metodou průběžného zpracování obrazu.*: Bakalářská práce ČVUT, 2011.
- [6] M. Kavková, *Měřicí kamera založená na Raspberry Pi.*: Diplomová práce ČVUT, 2014.
- [7] M. a kol. Bass, *Handbook of optics*. San Francisco: McGraw Hill, 1995.
- [8] J. W. Goodman, *Introduction to Fourier Optics*. Stanford: McGraw Hill, 1996.
- [9] Mathworks. (2015) Matlab 2015a. [Online]. <http://www.mathworks.com/>
- [10] Strang G., *Introduction to Linear Algebra*. Cambridge: Wellesley-Cambridge Press, 2009.
- [11] H.B. Nielsen, O. Tingleff K. Madsen, *Methods for non-linear least squares problems*. Copenhagen: Technical University of Denmark, 2004.
- [12] P. Bělík, *Diplomová práce - Smínač polohy využívající obrazové informace.*: Diplomová práce ČVUT, 2011.

- [13] Analog Devices. Blackfin Processors ADSP-BF5xx. [Online]. <http://www.analog.com/en/processors-dsp/blackfin/products/index.html>
- [14] Analog Devices, *ADSP-BF533 Hardware Reference.*, 2009. [Online]. [http://www.analog.com/static/imported-files/processor\\_manuals/bf533\\_hwr\\_Rev3.4.pdf](http://www.analog.com/static/imported-files/processor_manuals/bf533_hwr_Rev3.4.pdf)
- [15] FTDI chip, *FT2232H Dual High Speed USB to Multipurpose UART/FIFO IC - Hardware reference.*, 2010.
- [16] Analog Devices, *C/C++ Compiler and Library Manual for Blackfin® Processors.*, 2014.
- [17] Aptina, *MT9V032: 1/3-Inch Wide-VGA Digital Image Sensor - Reference manual.*, 2006.
- [18] Physik Instrumente. (2015) M-511 High-Precision Linear Translation Stage. [Online]. <http://www.physikinstrumente.com/product-detail-page/m-511-m-521-m-531-702100.html>
- [19] Physik Instrumente. (2015) C-863 Mercury Servo Controller. [Online]. <http://www.physikinstrumente.com/product-detail-page/c-863-900606.html>
- [20] MayTec. AL profil 40x40, 3E, heavy specifikace. [Online]. <http://www.maytec.org/detail/40403e.htm>
- [21] Union Optronics, *U-LD-650543A - 650nm Red Laser Diode.* Taiwan, 2008.

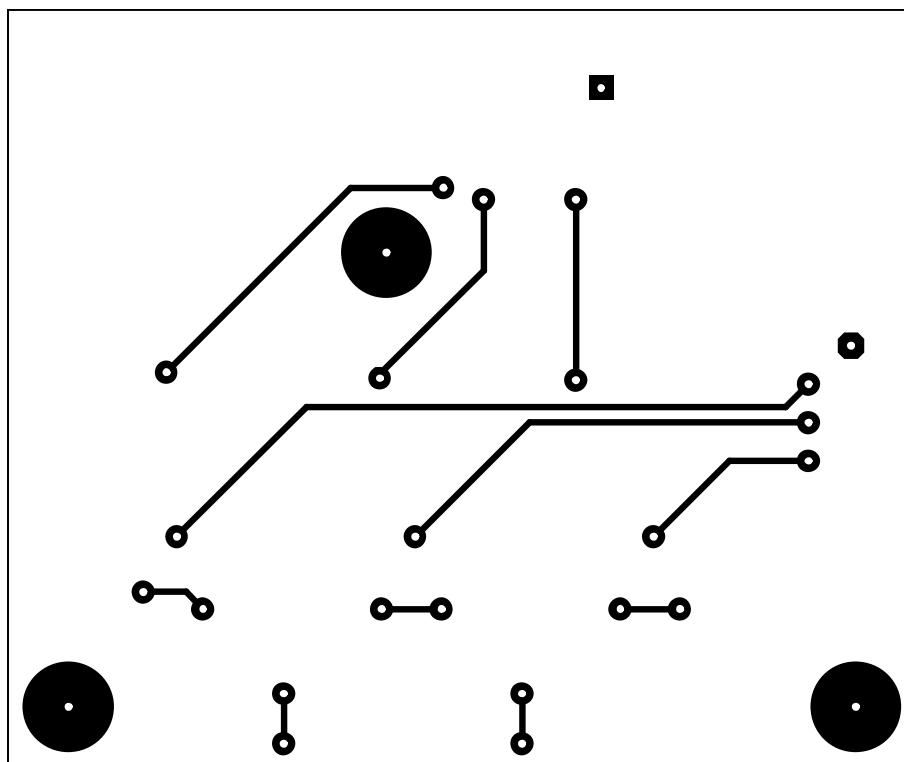
**Příloha A: Měření difrakčních obrazců**

Obr. A.1 – Reálné tvary difrakčních obrazců v různých vzdálenostech

## Příloha B: Návrh DPS budiče laserů



Obr. B.1 – Výkres DPS pro budič laserů - vrstva TOP



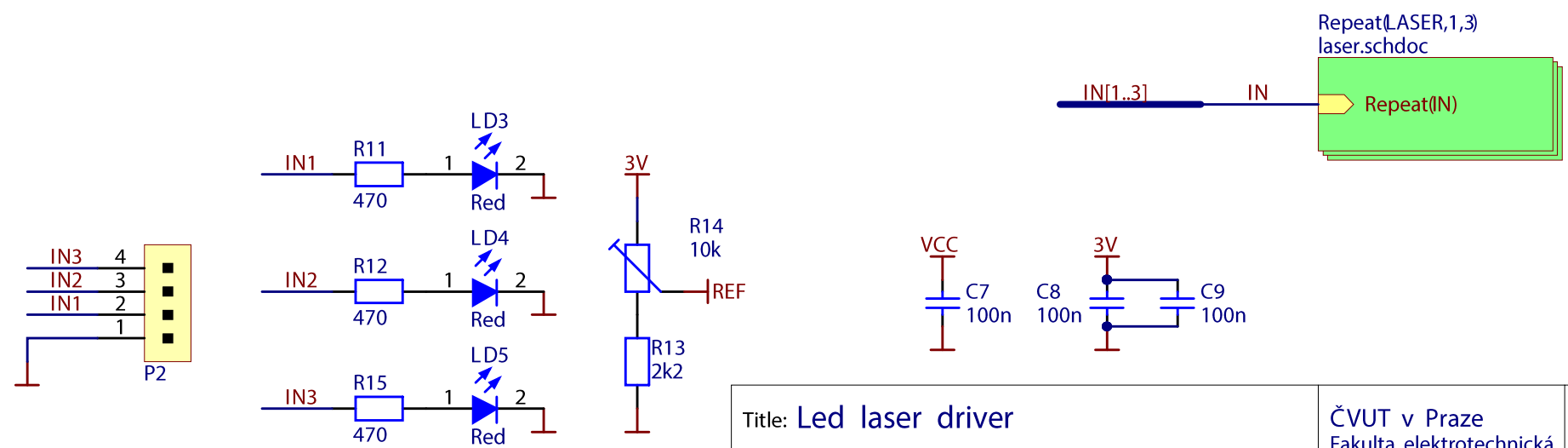
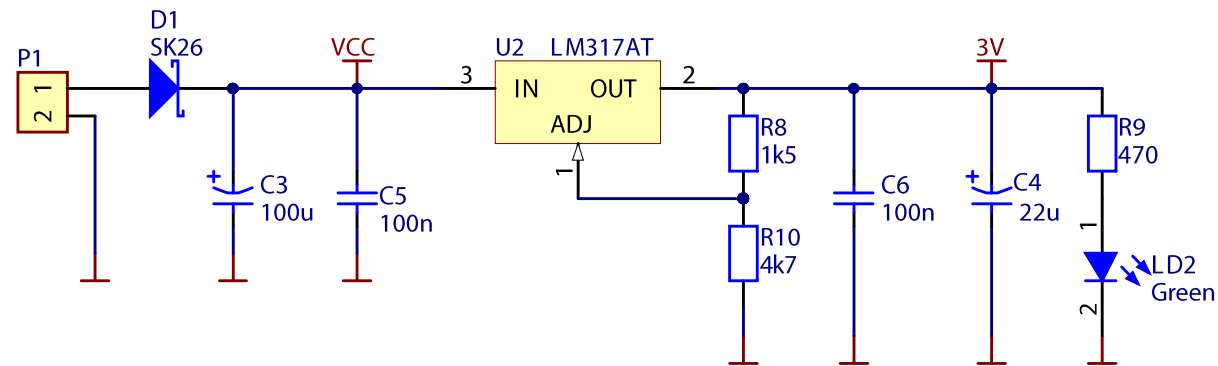
Obr. B.2 – Výkres DPS pro budič laserů - vrstva BOT

1

2

3

4



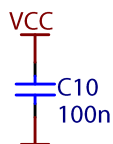
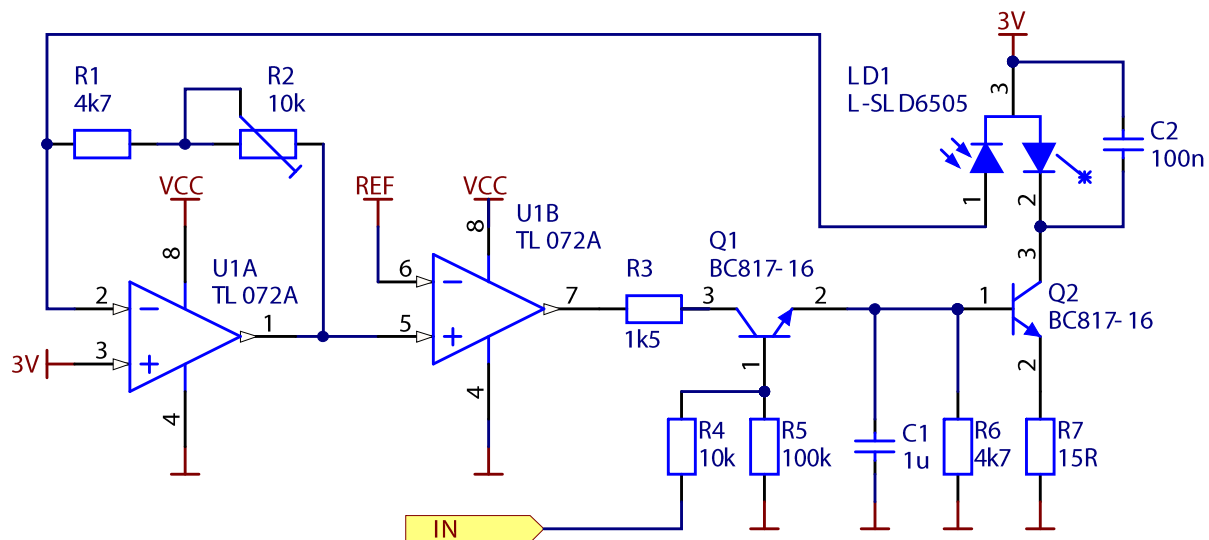
Title: Led laser driver		ČVUT v Praze Fakulta elektrotechnická Katedra měření Laboratoř videometrie	
Author: Jiří Hladík	Revision: 1		
Date: 5.5.2015	Sheet1 of 2		

1

2

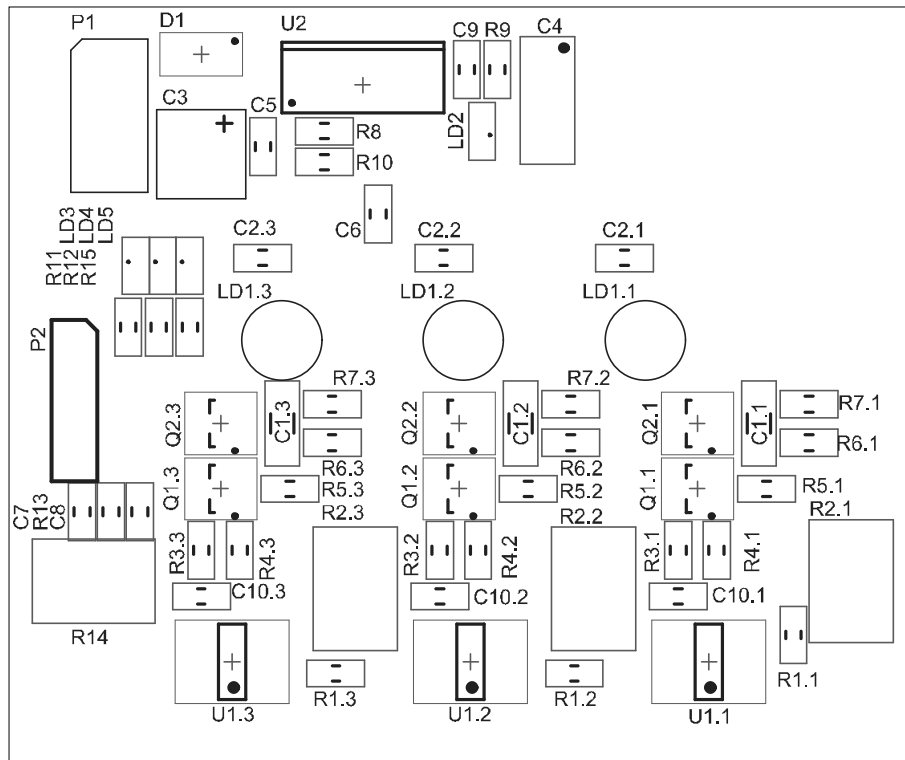
3

4



Title: <b>Led laser driver</b>		ČVUT v Praze Fakulta elektrotechnická Katedra měření Laboratoř videometrie	
Author: Jiří Hladík	Revision: 1		
Date: 5.5.2015	Sheet 2 of 2		

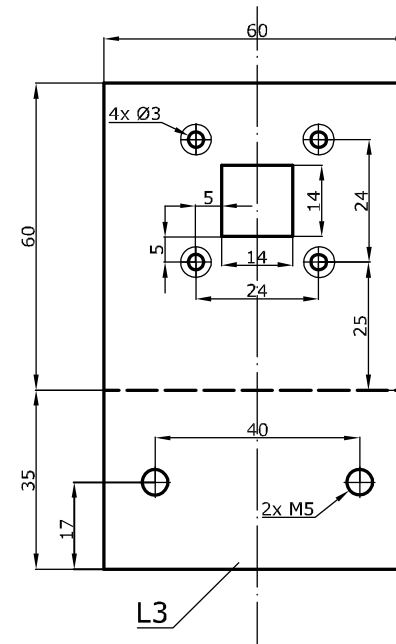
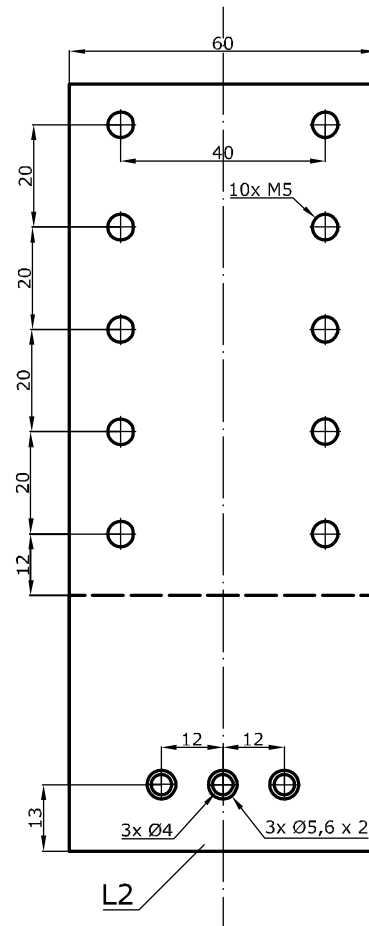
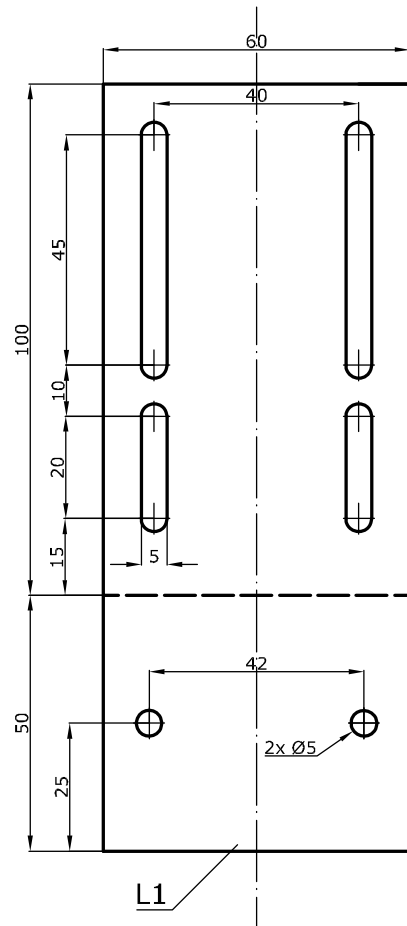
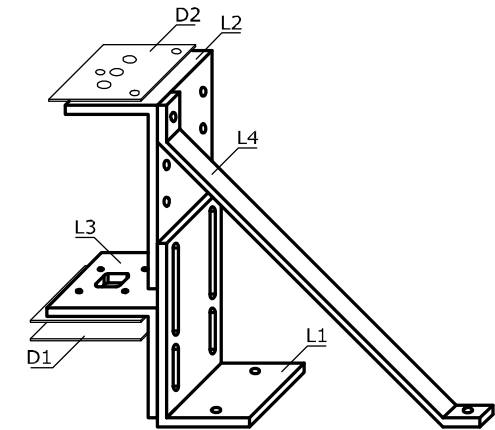
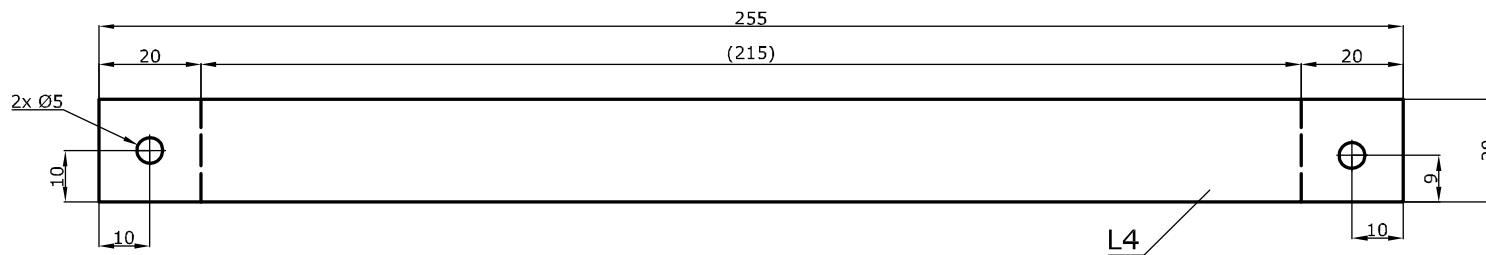




Obr. B.3 – Osazovací výkres pro budič laserů

Hodnota	Počet	Pozice	Pouzdro
1u	3	C1.1, C1.2, C1.3	CAP1206
100n	11	C2.1, C2.2, C2.3, C5, C6, C7, C8, C9, C10.1, C10.2, C10.3	CAP0805
100u	1	C3	CAP_ELYT_6,3mm
22u	1	C4	CAP_TANTAL_C
SK26	1	D1	DO-214AC
L-SLD6505	3	LD1.1, LD1.2, LD1.3	Laser
Green	1	LD2	LED_GREEN_0805
Red	3	LD3, LD4, LD5	LED_RED_0805
	1	P1	Konektor 5_08
CON 4	1	P2	HEADER 1x4
BC817-16	6	Q1.1, Q1.2, Q1.3, Q2.1, Q2.2, Q2.3	SOT-23
4k7	7	R1.1, R1.2, R1.3, R6.1, R6.2, R6.3, R10	RES0805
10k	4	R2.1, R2.2, R2.3, R14	TRIM_5x5mm
1k5	4	R3.1, R3.2, R3.3, R8	RES0805
10k	3	R4.1, R4.2, R4.3	RES0805
100k	3	R5.1, R5.2, R5.3	RES0805
15R	3	R7.1, R7.2, R7.3	RES0805
470	4	R9, R11, R12, R15	RES0805
2k2	1	R13	RES0805
TL072A	3	U1.1, U1.2, U1.3	SOIC-8
LM317AT	1	U2	TO220

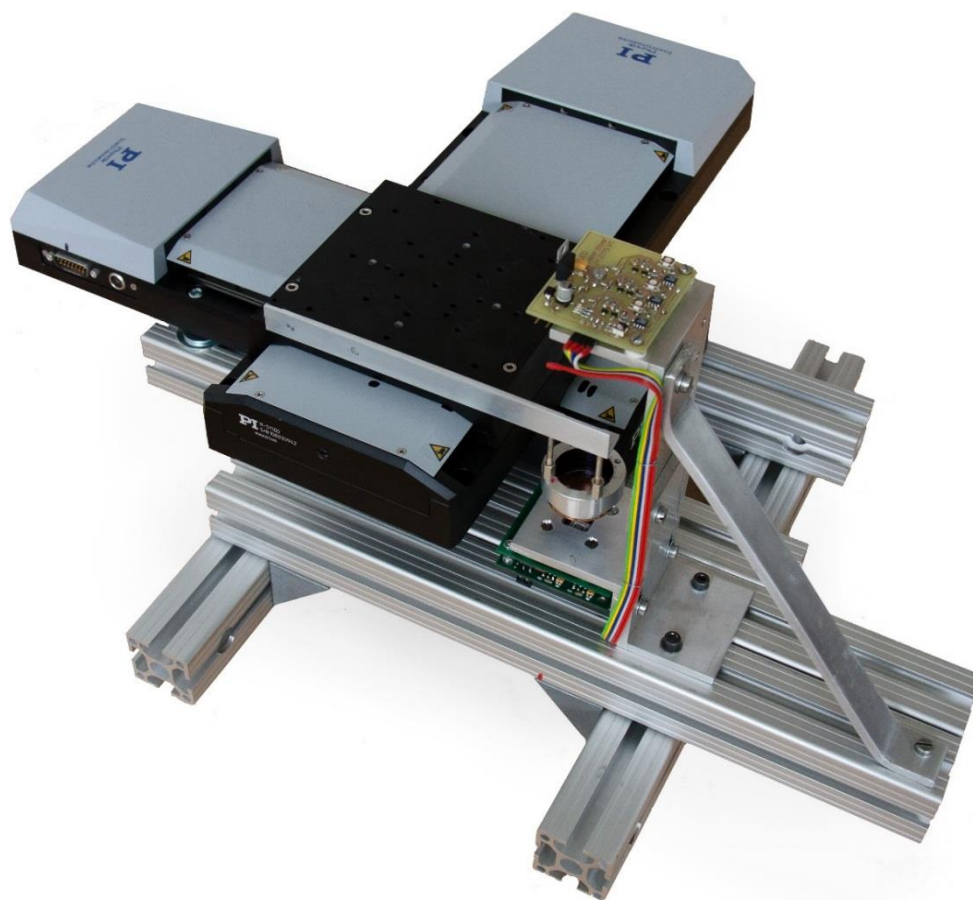
Tab. B.1 – Soupis materiálu pro budič laserů



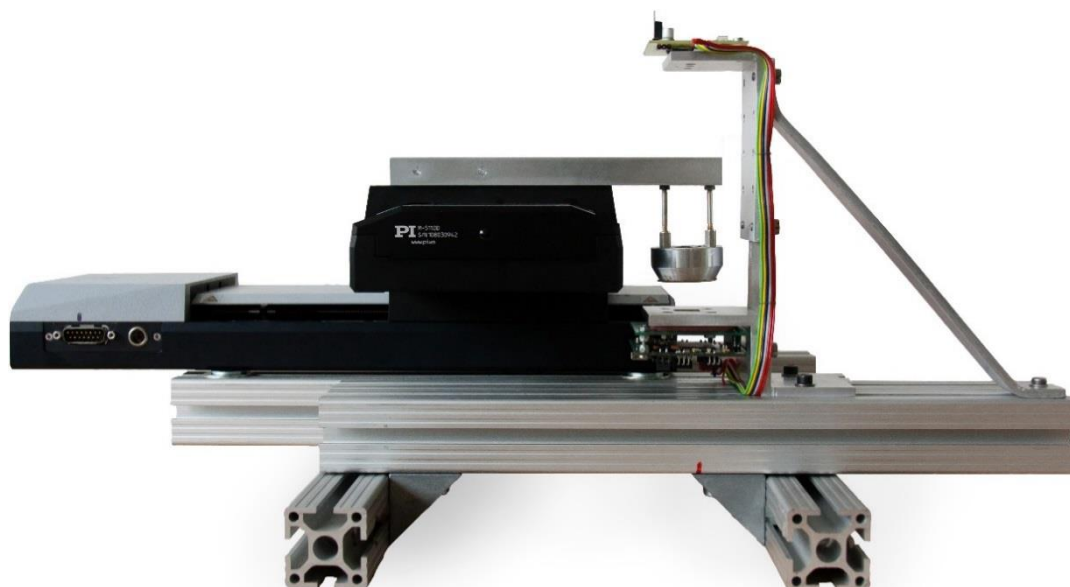
L1	AL profil L 100x50mm tl. 5mm
L2	AL profil L 100x50mm tl. 5mm
L3	AL profil L 60x60mm tl. 5mm
L4	AL pásek 20x5mm

<b>Mechanická konstrukce modulů</b>			<b>ČVUT v Praze</b> Fakulta elektrotechnická Katedra měření Laboratoř videometrie	
Autor: Jiří Hladík		Datum: 6.5.2015		
Měřítko: 1:1	Verze: 1	Formát: A3		

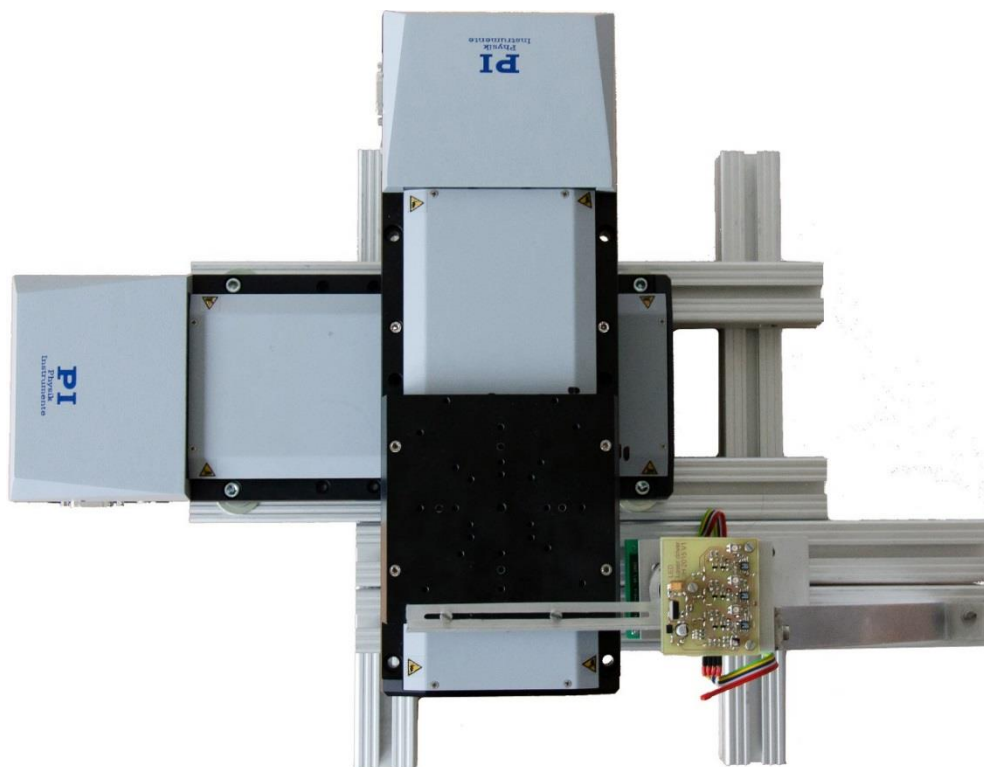
## Příloha C: Fotodokumentace



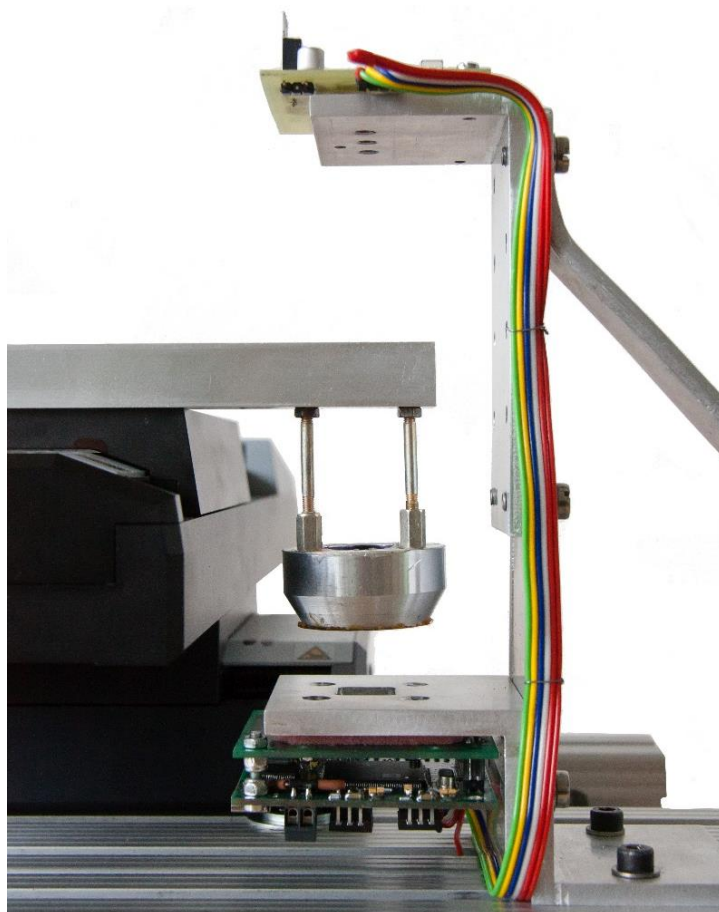
Obr. C.4 – Celkový pohled na mikropoložovací systém



Obr. C.5 – Pohled na polohovací systém z boku



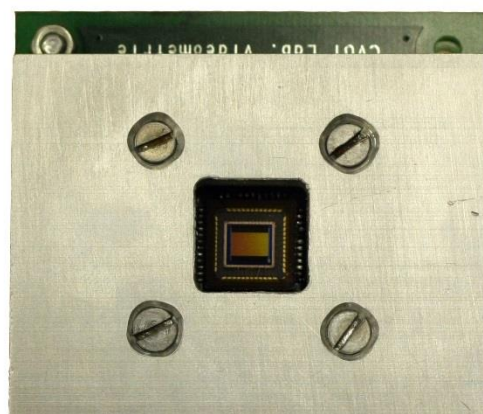
Obr. C.6 – Pohled na systém z vrchu



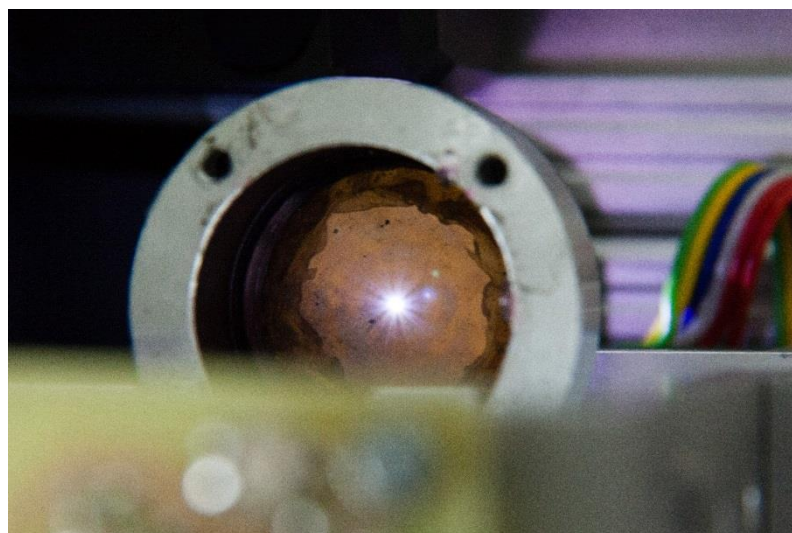
Obr. C.7 – Detail odměřovací soustavy



Obr. C.9 – Detail umístění laserů



Obr. C.8 – Detail uložení CMOS senzoru



Obr. C.10 – Detail clonky s prosvítajícím světlem bílé LED

## Příloha D: Obsah přiloženého CD

<i>Obsah</i>	<i>Cesta k souboru</i>
Elektronický text práce	Hladik_prubezne_zpracovani_obrazu_v_DSP_pro_sledovani_objektu.pdf
Skripty v programu Matlab	Matlab/
Zdrojové kódy pro zpracování stínových obrazů	Blackfin_SRC/Shadowed_images/
Zdrojové kódy pro zpracování jasových značek	Blackfin_SRC/Bright_signs/
PC zobrazovač	PC_SRC
Dokumentace k budiči laserů	Laser_driver/
Dokumentace k mechanické konstrukci	Mechanic/
Fotodokumentace	Images/Photo/
Obrázky použité v práci	Images/Other

Tab. D.2 – Obsah přiloženého CD