

# ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

Přesné měření polohy kamerami CCD a CMOS

Vedoucí práce : Ing. Jan Fischer CSc.

Autor : Bc. Vladimír Coufal

**2011**



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Vladimír Coufal**

Obor: **Kybernetika a měření – blok Měřicí a přístrojové systémy**

Název tématu česky: **Přesné měření polohy kamerami CCD a CMOS**

Název tématu anglicky: **Precise Position Measurement Using CCD and CMOS Camera**

### Zásady pro vypracování:

Analýzujte meze použitelnosti kamer CCD a CMOS (s projekčním objektivem i bez něj) pro přesné měření polohy. Navrhněte a realizujte experimentální pracoviště s polohovacím mechanismem, kterým se bude pohybovat snímanými objekty, případně snímanými zdroji záření. Vytvořte potřebné programové vybavení, které zajistí ovládání experimentu, ukládání snímků a jejich zpracování na PC. Navrhněte a realizujte metody měření určující přesnou polohu středu objektu, polohu hran, čar a dalších použitých značek. Ověřte možnost zpětnovazebního řízení polohy na základě informace zjištěné vyhodnocením obrazu.


### Seznam odborné literatury:

- [1] Fischer, J.: Optoelektronické senzory a videometrie. Skripta ČVUT, Praha 2002.
- [2] Heijden F.: Image Based Measurement Systems. Hoboken, NJ, Wiley, 1994.
- [3] Kučera, J.: Systém pro automatické ověřování videometrických metod měření rozměru a polohy. Diplomová práce ČVUT-FEL, Praha 2005.

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

Datum zadání diplomové práce: **26. listopadu 2009**

Platnost zadání do<sup>1</sup>: **10. června 2011**

  
Prof. Ing. Pavel Ripka, CSc.  
vedoucí katedry



  
Doc. Ing. Boris Šimák, CSc.  
děkan

V Praze dne 23.11.2009

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

## **ABSTRAKT**

Diplomová práce analyzuje problematiku přesného měření pomocí průmyslově vyráběných kamer a „smart kamer“. Zabývá se návrhem speciálního pracoviště, které slouží k pořizování velmi přesných subpixelových náměrů. Vytváří a implementuje softwarové vybavení pro pořizování těchto náměrů i jejich zpracování. V poslední části vyhodnocuje vlastnosti chování hran a světelných objektů při subpixelovém měření při použití lineární a fotometrické interpolace.

## **ABSTRACT**

This diploma thesis analyses the problem of precise position measurement using mass-produced cameras or "smart cameras". It deals with a design of special workplace, which is used for obtaining very precision sub-pixel position measurement. It creates and implements a software for getting and processing of these measurements. The last part of this thesis evaluates characteristics of contour and light objects behavior during sub-pixel measurement in using of linear and photometric interpolation.

## **PROHLÁŠENÍ**

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

V Praze dne .....

.....

**Bc. Vladimír Coufal**

## **PODĚKOVÁNÍ**

Děkuji vedoucímu diplomové práce Ing. Janu Fischerovi, CSc. a doktorandskému pracovníkovi Ing. Ondřeji Pribulovi za pomoc při objasňování některých problematik v oblasti videometrického měření a poskytnutí dalších cenných rad při zpracování této diplomové práce. Poděkování určitě patří i mým rodičům, kteří mě po celou dobu studia velmi podporovali.

V Praze dne .....

.....

Bc. Vladimír Coufal

# OBSAH

<b>Seznam obrázků</b>	<b>viii</b>
<b>Seznam tabulek</b>	<b>xi</b>
<b>Úvod</b>	<b>1</b>
<b>1 Rozbor problematiky</b>	<b>2</b>
1.1 Senzory v kamerových systémech .....	2
1.1.1 Kamery se senzory CCD.....	2
1.1.2 Kamery se senzory CMOS .....	4
1.1.3 Velikosti senzorů a jejich rozměry .....	5
1.2 Výstupní rozhraní kamer .....	6
1.2.1 Analogové rozhraní (kompozitní).....	6
1.2.2 Digitální rozhraní .....	8
1.2.3 Kompresní algoritmy a prezentace dat obrázků .....	9
1.3 Objektiv, struktura a jeho vliv na zobrazení obrazu .....	10
1.3.1 Další zdroje chyb ovlivňující přesnost měření .....	12
1.4 Metody optického měření rozměru objektu.....	13
1.4.1 Měření bez projekčního objektivu .....	13
1.4.2 Výpočet rozměrů objektu s projekční soustavou .....	14
1.5 Metody detekce polohy hrany objektu.....	15
1.5.1 Určování polohy hrany lineární interpolací .....	15
1.5.2 Určování polohy hrany fotometrickou interpolací.....	16
1.6 Určování pozice objektů a jejich vzdálenosti s využitím polohy těžiště	18
<b>2 Návrh a realizace měřicího pracoviště</b>	<b>19</b>
2.1 Konstrukce měřicí soustavy .....	19
2.2 Blokové schéma zapojení měřicí soustavy .....	21
2.3 Softwarové vybavení pracoviště.....	21
2.3.1 KSD - Software pro sběr dat z kamery .....	22
2.3.2 Software pro řízení pojezdu .....	26
2.3.3 Software pro zpracování dat .....	28

2.4	Formáty ukládaných dat.....	31
2.5	Zásady pro pořízení a vyhodnocení náměrů .....	33
<b>3</b>	<b>Algoritmy měření a vyhodnocení dat</b>	<b>34</b>
3.1	Software pro náměr dat.....	34
3.2	Software pro zpracování dat .....	35
3.2.1	Formát CSV .....	36
3.2.2	Třída DBImg.....	36
3.2.3	Algoritmus vykreslení přechodu hrany.....	39
3.2.4	Algoritmus výpočtu lineární interpolace hrany .....	39
3.2.5	Algoritmus výpočtu fotometrické interpolace hrany .....	40
3.2.6	Algoritmus výpočtu souřadnic těžiště.....	41
3.3	Software pro řízení pojezdu.....	41
3.3.1	Řídicí jednotka MARSII.....	42
3.3.2	Řídicí jednotka MERCURY PI.....	43
<b>4</b>	<b>Zpracování náměrů a výsledky měření</b>	<b>46</b>
4.1	Určení hranice počtu vzorků pro průměrování.....	46
4.2	Měření šumu za tmy a jeho vliv na měření.....	47
4.3	Měření přechodu ostré hrany .....	51
4.3.1	Zpřesnění interpolační metody, fázový posun hrany .....	57
4.4	Měření přechodu rozostřené hrany .....	58
4.5	Hledání pozice středu objektu ohraničeném dvěma hranami .....	60
4.6	Měření vzdálenosti posunu světelného bodu .....	63
<b>5</b>	<b>Závěr</b>	<b>65</b>
	<b>Reference</b>	<b>67</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>68</b>
	<b>Seznam příloh</b>	<b>69</b>

## SEZNAM OBRÁZKŮ

Obr. 1.1 Blokové schéma zapojení a principu fungování CCD snímače (zdroj [1]).....	2
Obr. 1.2 Princip interpolace barev z rozložení Bayerovského filtru na ploše senzoru (zdroj [2]).....	3
Obr. 1.3 Citlivost CCD senzoru v závislosti na vlnové délce záření světla (zdroj [3])....	4
Obr. 1.4 Schéma uspořádání buňky CMOS senzoru (převzato z [4]) .....	5
Obr. 1.5 Citlivost CMOS senzoru v závislosti na vlnové délce záření světla (zdroj [5])	5
Obr. 1.6 Struktura monochromatického kompozitního videosignálu (zdroj [6]) .....	7
Obr. 1.7 Struktura barevného kompozitního videosignálu (zdroj [6]) .....	7
Obr. 1.8 Náhled konstrukce objektivu. $F$ a $F'$ ohniska, $f$ a $f'$ ohniskové vzdálenosti, „ $a$ “ vzdálenost předmětu od hlavní roviny, „ $l$ “ zaostřovací vzdálenost (zdroj [8]) .....	11
Obr. 1.9 Blokové schéma principu měření s bodovým zdrojem záření nebo s kolimátorem (zdroj [9]).....	13
Obr. 1.10 Blokové schéma měřící soustavy s projekčním objektivem (zdroj [8]).....	14
Obr. 1.11 Graf přechodu hrany jednoho řádku.....	15
Obr. 1.12 Lineární interpolace.....	16
Obr. 1.13 Fotometrická interpolace .....	17
Obr. 2.1 Návrh konstrukce měřící soustavy .....	19
Obr. 2.2 Fotodokumentace konstrukce měřící soustavy.....	20
Obr. 2.3 Blokové schéma měřící soustavy .....	21
Obr. 2.4 Screen shot základního zobrazení aplikace KSD pro náměr obrazových dat ..	22
Obr. 2.5 Screen shot – výběr video zařízení – seznam připojených video zařízení k PC .....	23
Obr. 2.6 Screen shot – sekce Video zařízení – výběr vstupního zařízení.....	23
Obr. 2.7 Screen shot – sekce Parametry výřezu – nastavení pozice a velikosti výřezové oblasti .....	23
Obr. 2.8 Screen shot – sekce Nastavení pojezdu – možnosti ovládání.....	24
Obr. 2.9 Screen shot – sekce Nastavení měření – možnosti nastavení měřícího cyklu..	25
Obr. 2.10 Screen shot – sekce Pracovní adresář – volba umístění naměřených dat.....	25
Obr. 2.11 Screen shot – sekce Souřadnice těžiště – vizualizace polohy těžiště a maximální jasové hodnoty .....	25
Obr. 2.12 Screen shot – aplikace pro Zpracování měření v2.1.....	29



Obr. 2.13	Struktura databázového formátu ukládání dat .....	32
Obr. 3.1	Sekvenční diagram – aplikace pro náměr dat .....	34
Obr. 3.2	Screen Excel – rozparsování dat CSV formátu .....	36
Obr. 3.3	Třída DBimg .....	36
Obr. 3.4	Sekvenční diagram – aplikace pro řízení pojezdu .....	42
Obr. 4.1	Graf střední hodnoty signálu v závislosti na počtu vzorků .....	47
Obr. 4.2	Graf plošného jasového rozložení signálu na senzoru CMOS 1,2MPix – bez průměrování. Závěrka 0x419hex (defaultní hodnota) .....	48
Obr. 4.3	Graf plošného jasové rozložení signálu na senzoru CMOS 1,2MPix – s průměrováním 50 snímků. Závěrka 0x419hex (defaultní hodnota) .....	49
Obr. 4.4	Graf plošného jasové rozložení signálu na senzoru CMOS 1,2MPix – s průměrováním 50 snímků. Závěrka 0x100hex .....	50
Obr. 4.5	Graf závislosti proudu poškozeného CMOS pixelu na nastavení elektronické závěrky (shutter delay) .....	50
Obr. 4.6	Graf přechodu hrany – kamera Visor, pojezd ESSA, krok 10um, průměrováno z 9 snímků .....	51
Obr. 4.7	Graf odchylky od linearity lineární oblasti přechodu z grafu 4.6 .....	52
Obr. 4.8	Graf lineární interpolace hrany, kamera Visor, pojezd ESSA, krok 10um .....	52
Obr. 4.9	Graf odchylky od linearity interpolačního grafu na obrázku 4.8 .....	53
Obr. 4.10	Graf přechodu hrany – kamera Visor, pojezd PI instruments, krok 10um, průměrováno z 25 snímků .....	54
Obr. 4.11	Graf odchylky od linearity lineární oblasti přechodu z grafu 4.10 .....	54
Obr. 4.12	Graf lineární interpolace hrany, kamera Visor, pojezd PI instrument, krok 1um .....	55
Obr. 4.13	Graf odchylky od linearity interpolačního grafu na obrázku 4.12 .....	55
Obr. 4.14	Graf fotometrické interpolace hrany, kamera Visor, pojezd PI instrument, krok 1um .....	56
Obr. 4.15	Graf odchylky od linearity grafu na obrázku 4.14 .....	56
Obr. 4.16	Graf fázově posunutých interpolovaných hran .....	57
Obr. 4.17	Graf součtu dvou fázově posunutých hran .....	58
Obr. 4.18	obrázky vlevo zaostřená hrana, vpravo rozostřená hrana .....	59
Obr. 4.19	Graf odchylky od linearity přechodu rozostřené hrany .....	59
Obr. 4.20	Objekt pro měření pozice ohraničené dvěma hranami. ....	60
Obr. 4.21	Graf střední hodnoty lineární interpolace dvou hran .....	61
Obr. 4.22	Graf odchylky od linearity z grafu na obrázku 4.21 .....	61
Obr. 4.23	Graf střední hodnoty fotometrické interpolace dvou hran .....	62

Obr. 4.24 Graf odchylky od linearity z grafu na obrázku 4.23.....	62
Obr. 4.25 Vlevo snímek z kamery světelného bodu, vpravo snímek upevnění diody na pojezd.....	63
Obr. 4.26 Graf posunu těžiště LED diody v závislosti na posunu pojezdu .....	64
Obr. 4.27 Graf odchylka od linearity posunu těžiště LED diody .....	64

## SEZNAM TABULEK

Tab. 1.1 Rozměry a velikosti běžných CCD a CMOS senzorů.....	6
Tab. 1.2 Struktura BMP souboru reprezentující obrazová data.....	10
Tab. 3.1 Nastavení parametrů sériového portu .....	42
Tab. 3.2 Tabulka použitých příkazů .....	43
Tab. 3.3 Nastavení parametrů sériového portu .....	44
Tab. 3.4 Tabulka identifikovaných příkazů .....	44
Tab. 4.1 Střední hodnoty signálu pixelu v závislosti na počtu snímků v průměrování ..	47

# ÚVOD

O měření pomocí kamer lze s menší nadsázkou prohlásit, že měřit objekt, nalézt hranu, spočítat těžiště lze jakoukoliv průmyslovou nebo low-end kamerou dostupnou na trhu. Bohužel už nelze říci, že tyto kamery budou měřit se stejnou přesností nebo budou pracovat za stejných podmínek. Základní a zásadní rozdíl v kvalitě pořízeného náměru u těchto kamer je připisován použitému senzoru, jeho citlivosti, kvalitě objektivu, algoritmu zpracování výsledného obrazu a rozhraní, kterým lze data z kamery získat.

Cílem této práce je vyhodnotit možnosti měření pomocí těchto kamer se zaměřením na jednotlivé části měřicího ústrojí ovlivňující výslednou kvalitu pořízeného snímku a poukázat na problematiku reálného nasazení v provozu. Dále diagnostikovat reálné schopnosti měření pomocí velmi rozšířeného senzoru typu CMOS a CCD a vytvořit pracoviště, kde bude možné ověřit metody pro detekci pozice hran a posunu světelných objektů. Např. lineární a fotometrická interpolace, výpočet těžiště, zobrazení histogramu, stanovení chyb linearity atd.

V první fázi je třeba udělat rozbor vlastností dostupných kamer, jejich možnosti připojení, zhodnotit očekávané výstupní parametry, definovat si parametry vnášející chyby do měření a způsob jak lze tyto chyby potlačit nebo korigovat. Jako měřicí objekty jsou k dispozici monochromatická kamera Digital KC-30H2C s 1/3" CCD snímačem a smart kamera Visor s 1,3MPx CMOS senzorem MT9M001. Tato analýza nám poskytne souhrn požadavků potřebných pro návrh ovládacího a vyhodnocovacího softwaru.

V druhé fázi se budeme zabývat návrhem bytelné a co nejvíce ořesům stabilní konstrukce, na kterou lze umístit pojezdový osový kříž spolu se snímacím prvkem, v našem případě kamerou. Tato konstrukce musí umožňovat pohodlnou změnu pozice kamery od snímané plochy případně výměnu pojezdového osového kříže.

Pro jednoduchou a pohodlnou obsluhu je nutné vytvořit softwarové rozhraní umožňující ovládání pojezdů a ukládání naměřených dat. Pro polohování objektů jsou k dispozici dva typy pojezdů a to pojezd značky ESSA s řídicí jednotkou MarsII a velmi přesný pojezd od firmy Physikinstrumente s řídicí jednotkou Mercury PI. Software by dále měl umět zpracovávat data z připojeného frame-graber (ADC), IP kamery a USB rozhraní kamery Visor. Data by měla mít jednoduchý formát pro budoucí zpracování. Tj. ukládání jednotlivých snímků ve formátu bitmapy, nebo použít jiný dobře zpracovatelný formát.

Zpracování měření bude možné provádět ve zcela libovolném softwaru pracujícím s obrázky. Pro zjednodušení však bude vytvořena aplikace vlastní, která bude mít za účel data předzpracovat nebo provádět rutinní automatické úpravy.

Pro vytvoření programového vybavení ovládání pojezdů, sběr dat a vyhodnocování naměřených dat, jsem zvolil programovací jazyk C# od společnosti Microsoft spolu s vývojovým prostředím Visual Studio 2008 Express edice.

# 1 ROZBOR PROBLEMATIKY

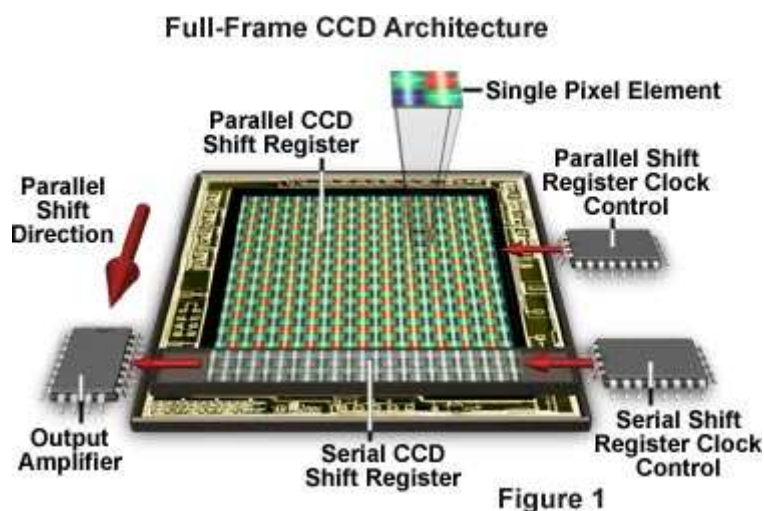
Při využití kamer jako měřicího zařízení je přesnost a rychlost pořízeného náměru závislá především na použitém senzoru, jeho citlivosti, kvalitě objektivu, algoritmu zpracování výsledného obrazu a v neposlední řadě rozhraní, po kterém jsou data přenášena ke zpracování.

## 1.1 Senzory v kamerových systémech

Nejrozšířenějším snímacím prvkem používaném v dnešní průmyslové technice je čip typu CMOS a CCD. CMOS je masově nasazován pro svou jednoduchou a levnou výrobu kompenzovanou horšími vlastnostmi za ztížených světelných podmínek. CCD čip si vybudoval svoje postavení díky kvalitě barevného podání a citlivostí především u profesionálních zařízení.

### 1.1.1 Kamery se senzory CCD

Kamery se senzory CCD mají nejširší uplatnění v profesionální nebo speciální technice (zrcadlové fotoaparáty, astronomické dalekohledy). Hlavní důraz je kladen na kvalitu obrazu a v případě barevných kamer i na věrnost podání barev.



Obr. 1.1 Blokové schéma zapojení a principu fungování CCD snímače (zdroj [1])

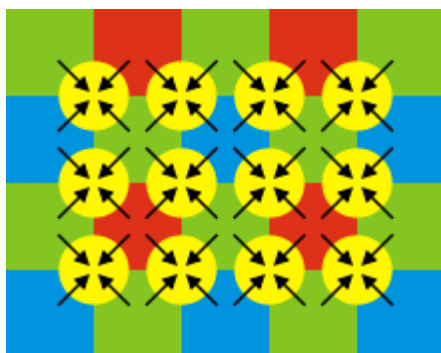
Senzor CCD, zobrazen na obrázku 1.1, je elektronická součástka určená ke snímání obrazové informace. Pracuje na principu posuvného paměťového registru. Využívá fyzikálního jevu zvaného fotoelektrický jev. Při tomto jevu jsou elektrony uvolňovány ze speciální látky v důsledku absorpce elektromagnetického záření. Uvolněný elektron se poté může podílet na elektrické vodivosti v polovodiči.

Senzory lze rozdělit na základní kategorie lineární (jednořádkové snímače) a plošné (víceřádkové senzory). Lineární funguje na principu postupného vyčítání

jasových hodnot jednotlivých pixelů. Tyto senzory jsou především známé jako čtečky čárového kódu nebo řádkové čtečky textu (fax). Plošné senzory fungují na obdobném principu.

V případě barevného snímače se užívají v zásadě dvě konstrukce. První je použití 3 senzorů R,G,B (red, green, black), kde před každým senzorem umístíme barevný filtr propouštějící příslušnou barvu.

Druhý způsob využívá pro snímání barevného obrazu jediný senzor CCD. Na jednotlivých pixelech jsou naneseny filtry o různé propustnosti světla. Takovýto filtr se nazývá Bayerovský filtr. Důležité je barevné rozložení na celé ploše senzoru. Typické rozložení a výpočet interpolace barev ukazuje obrázek 1.2.



Obr. 1.2 Princip interpolace barev z rozložení Bayerovského filtru na ploše senzoru (zdroj [2])

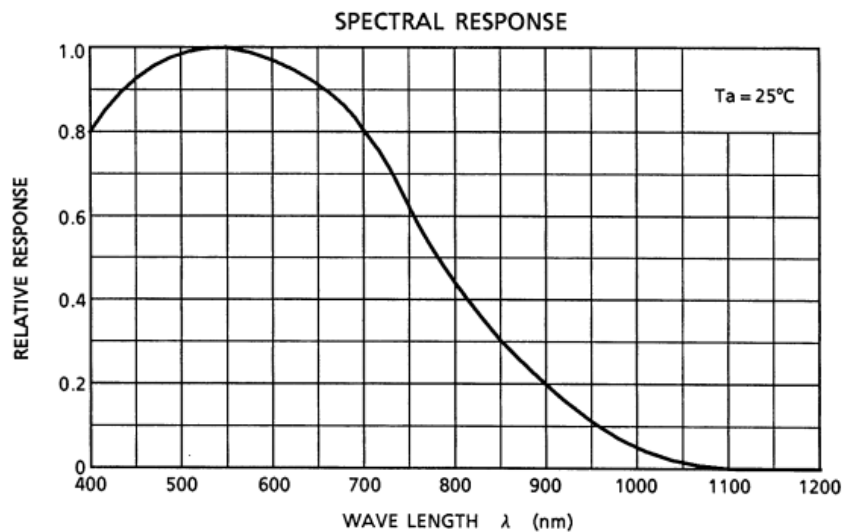
Na první pohled by se mohlo zdát, že celá plocha pixelu je zabrána velikostí fotocitlivého prvku. Avšak vlivem výrobního procesu a konstrukce samotné plochy tomu tak ve skutečnosti není. To má za následek, že plocha pixelu není konstantně citlivá. Tento efekt se snaží výrobci odstranit vytvořením mikroskopické čočky. Ta má za úkol zvyšovat efektivní plochu pixelu i jeho citlivost.

*Pozn: Diplomová práce by měla do jisté míry objasnit tuto problematiku pomocí reálných náměrů dat.*

Jednou z hlavních nevýhod CCD senzoru je analogový výstup. Po vyčtení hodnoty z pixelu je třeba signál převést A/D převodníkem, upravit a poté dále distribuovat.

Základními vlastnostmi CCD snímače jsou dynamický rozsah a citlivost. Dynamický rozsah nám udává schopnost snímače rozlišit barevný přechod od „nejčernější černé“ k „nejbělejší bílé“. To je dáno mírou interakce fotonů, které je schopen snímač pojmut, a vlivem šumu generovaného samotnými buňkami.

Citlivost udává schopnost přepočítat jasový rozsah fotometrického členu vůči intenzitě vstupního signálu. Vyjadřujeme ji formou tzv. ISO hodnot. Tato stupnice byla přejata z filmového materiálu. Čím vyšší citlivost máme, tím vyšší je vliv šumu na pořízený obraz. Obrázek 1.3 ukazuje citlivost monochromatického CCD senzoru na různé vlnové délky záření. Z grafu lze odvodit, že nejcitlivější oblast se nachází v pásmu 550nm, která připadá žlutému světlu.



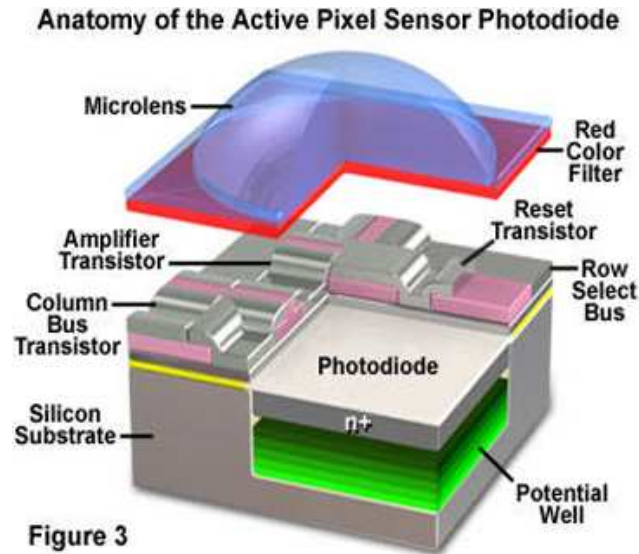
Obr. 1.3 Citlivost CCD senzoru v závislosti na vlnové délce záření světla (zdroj [3])

### 1.1.2 Kamery se senzory CMOS

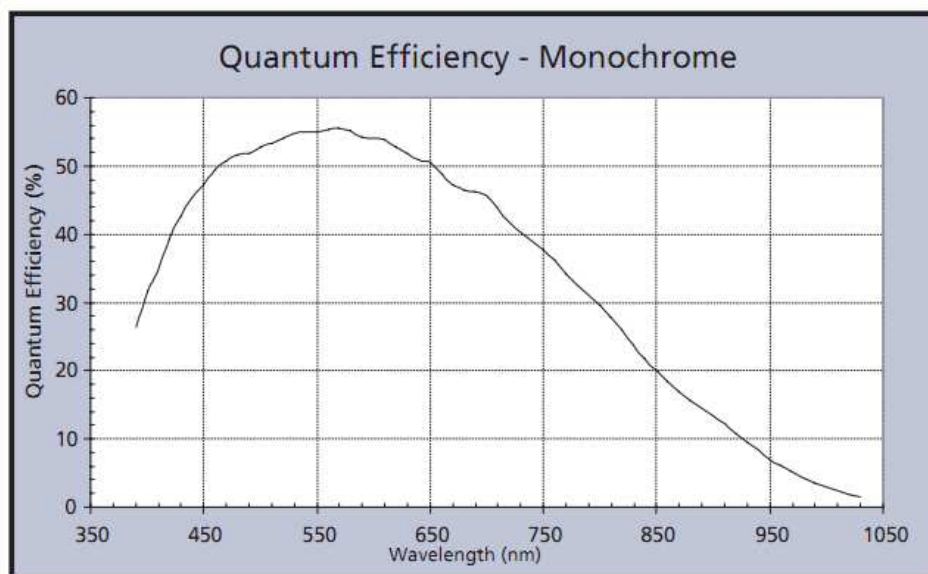
Kamery se senzorem CMOS jsou díky dobře zvládnuté výrobě a velikosti hojně rozšiřovány nejen do domácí elektroniky ale i do profesionálních zařízení. Nevýhodou u těchto senzorů je horší citlivost na světlo, nehomogenita fotocitlivosti a větší proudy za tmy, což přináší jistá omezení použití oproti snímačům CCD. Naopak mají vysokou odolnost proti rušení a nízkou spotřebu. Oproti CCD snímačům nepotřebují převádět analogový signál, jejich výstup je rovnou v digitální podobě. Každá buňka má svůj výstup a proto může být digitalizace dat provedena v jednom cyklu. Ovšem tato skutečnost závisí na dostatečně velké paměti pro uložení a zpracování dat.

Na obrázku 1.4 je znázorněna struktura jedné buňky CMOS senzoru. V křemíkovém substrátu, polovodiče typu P, je vytvořena oblast s polovodičem typu N+, která vytváří prvek fotodiody. Tato fotodioda má kolem sebe umístěné další součástky, zaručující zpracování signálu. Je to zesilovací tranzistor, resetovací tranzistor a obvody pro obsluhu sběrnice vyčítání dat. Nad těmito součástkami je umístěn, v případě barevného snímače, filtr propouštějící záření určité vlnové délky např. R, G nebo B. Ve stejném případě jako u CCD snímačů je použita mikročočka, která směřuje koncentraci záření přímo na fotodiodu a tím zvyšuje její citlivost.

Díky výrobním tolerancím dochází k nejednotnosti každého pixelu a to má za následek větší velikost šumu a interpolaci mezi sousedními pixely. Dochází tím ke ztrátě kontrastu a ostrosti obrazu. Obraz se pak jeví jako rozmazaný a bledý.



Obr. 1.4 Schéma uspořádání buňky CMOS senzoru (převzato z [4])



Obr. 1.5 Citlivost CMOS senzoru v závislosti na vlnové délce záření světla (zdroj [5])

Obrázek 1.5 zobrazuje citlivost senzoru v závislosti na vlnové délce záření. Senzor je opět nejcitlivější ve spektru žluté barvy, stejně jak CCD snímač.

### 1.1.3 Velikosti senzorů a jejich rozměry

Ve většině případů se rozměr senzoru udává v palcích. Tato konvence byla převzata od používání snímacích elektronek v historických dobách. Tyto elektrony se nazývaly „vidikony“ a vnější rozměr skleněné baňky určoval její rozměr.

Základním měřítkem ve videotechnice, ale i v běžné fotografii, se stal formát kinofilmu. Ten činí 36x24mm, tedy poměr stran 3:2. Běžný rozměr televizní obrazovky



činí 4:3 a dnes velmi rozšiřovaný formát 16:9, který je mnohem přirozenější pro sledování lidským okem. To lze ovšem tvrdit pouze v případě sledování pohyblivých scén. V případě statických scén je vhodnější formát 4:3 např. u bezpečnostních kamer.

Rozlišení samotného senzoru se udává v megapixelech. Tato jednotka udává počet obrazových bodů na senzoru. Obecně však nelze říci, že čím větší rozlišení, tím kvalitnější obraz.

Nejběžnější velikosti a rozměry jsou uvedeny v tabulce 1.1

Označení	Šířka (mm)	Výška (mm)	poměr
<b>1/4</b>	<b>3,2</b>	<b>2,4</b>	<b>4:3</b>
1/3,6	4	3	4:3
1/3,2	4,5	3,4	
<b>1/3</b>	<b>4,8</b>	<b>3,6</b>	<b>4:3</b>
1/2,7	5,3	4	4:3
1/2,5	5,76	4,29	4:3
1/2,35	5,9	4,4	
1/2,3	6	4,5	4:3
<b>1/2</b>	<b>6,4</b>	<b>4,8</b>	<b>4:3</b>
1/1,8	7,18	5,32	4:3
1/1,76	7,3	5,4	4:3
1/1,7	7,6	5,7	4:3
<b>1/1,65</b>	<b>8,7</b>	<b>5</b>	<b>16:9</b>
1/1,6	8	6	4:3
2/3	8,8	6,6	4:3
1	12,8	9,6	4:3

Tab. 1.1 Rozměry a velikosti běžných CCD a CMOS senzorů

## 1.2 Výstupní rozhraní kamer

### 1.2.1 Analogové rozhraní (kompozitní)

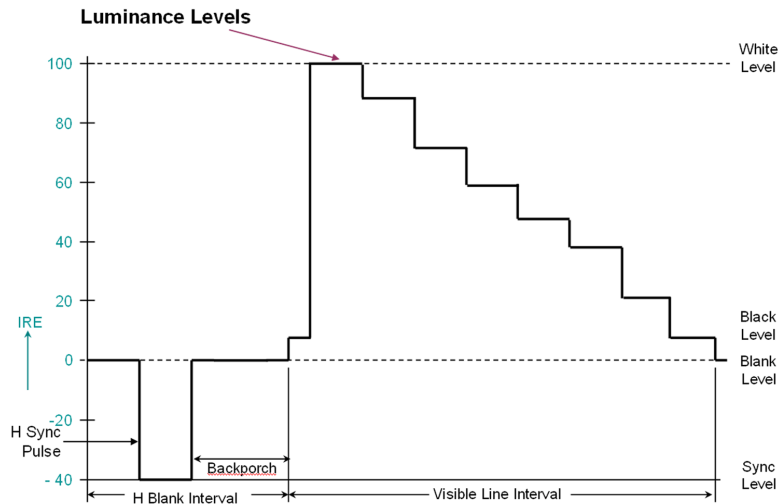
Analogové rozhraní není příliš vhodné pro velmi přesná měření. Při zpracování dochází v mnoha případech k „úpravě obrazu“ např. zvýraznění hran, úpravy jasu atd. Výrobce však neuvádí jaké algoritmy pro vylepšení obrazu používá a též nelze nastavit vlastnosti samotného zpracování.

K pozitivním stránkám analogových kamer lze zmínit, že pro zobrazení snímané scény není třeba využívat žádného specializovaného zařízení. Posloužit k tomu může zcela běžný barevný televizor. Problém ovšem nastane, pokud chceme s daným signálem dále pracovat, případně podle něho řídit nějaký výrobní proces. Je nutné tento signál převést do digitální podoby. Procesu převodu do digitální podoby se říká „grabování“. Slouží k tomu převodník zvaný „frame grabber“. Ten obsahuje velmi rychlý A/D převodník, DSP procesor a vyrovnávací paměť. Při samotném převodu

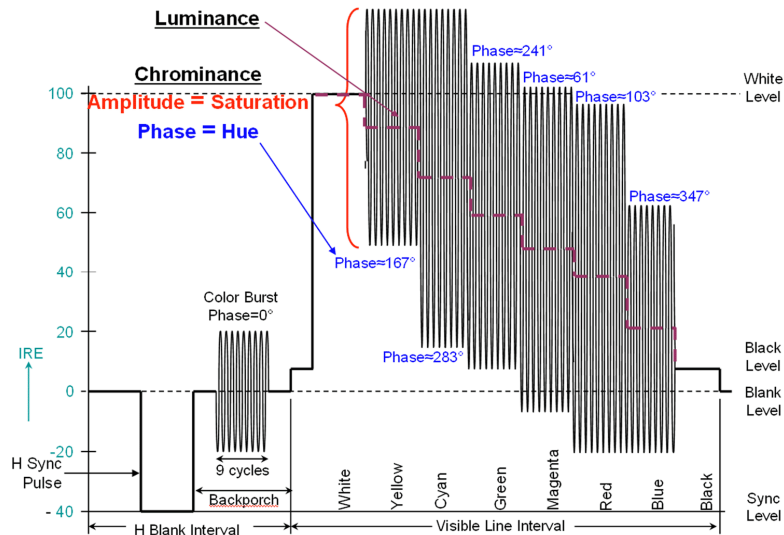
signálu ovšem může docházet k deformaci obrazové informace. Jedná se především o přepočít formátu sloupců a řádků. Vstupní signál, který má rozlišení 768x576 je transformován na formát 720x576. Takto přepočtený formát se pak projeví obdélníkovým tvarem pixelu.

Kompozitní video signál dosahuje šířky pásma kolem 5-6MHz. Jeho podobu pro monochromatický obraz vidíme na obrázku 1.6.

Případ barevného video signálu znázorňuje obrázek 1.7. Barva se zakóduje do fázově posunutého signálu a tato fáze nám určuje odstín barvy.



Obr. 1.6 Struktura monochromatického kompozitního videosignálu (zdroj [6])



Obr. 1.7 Struktura barevného kompozitního videosignálu (zdroj [6])

Vlivem velké šířky pásma kompozitního signálu, se stává tento signál velmi choulostivý na okolní rušení, které může způsobit pruhy v obraze, zvýšení šumu nebo pokles jasu barev. Těmto problémům lze předcházet správně impedančně přizpůsobeným stíněným vedením. Standardní nominální impedance je 75ohm.

K vedení signálu se nejčastěji používají koaxiální kabely označené zkratkou RG59. Jádru kabelu je z pevného drátu a opletení (stínění) dosahuje procentuálního pokrytí 93%. Praktické zkušenosti však ukázaly, že na menší vzdálenosti (cca do 10m) lze možné použít i koaxiální kabel s impedanční hodnotou 50ohm.

## 1.2.2 Digitální rozhraní

Druhou možností pro přenos dat z kamer je digitální rozhraní. Data se přenášejí pomocí sběrnice USB, Firewire nebo ethernet.

Při připojení ke sběrnici USB dojde v mnoha většinou k automatickému nainstalování ovladačů a poté k němu lze přistupovat jako ke standardnímu „Video Device“. Bohužel systém Windows nedovolí používat USB zařízení na přímo a je vždy nutné použít nějakou komunikační mezivrstvu. např. DirectX a jeho část určenou pro práci s videem „DirectShow“. Výrobci tak nemusí řešit některé problémy s přístupem např. ke grafickému akceleratoru nebo implementaci komprimačních algoritmů. Z pohledu výrobců je to úžasná věc, ale pro účely měření a zkoumání fyzikálních vlastností senzorů věc zcela nevhodná. Nelze s určitou jistotou říci, co se s daným nevzorkovaným signálem děje, případně jaké filtry byly použity pro jeho zpracování. Tato situace je řešitelná pouze napsáním vlastního ovladače pro převodník, který si sami navrheme. Pro přenos dat mohou využít mnoho typů tzv. stream protokolů nebo nejjednodušší formu přenosu a to pomocí obrázků. Jeden frame, představuje jeden obrázek. Vhodnost a výběr protokolu závisí na okolnostech použití videokamery.

### Web kamery

Nejrozšířenějším typem zařízení využívající sběrnici USB bych označil názvem „web kamery“. Operační systém windows ji interpretuje jako již výše uvedený „video device“. Jelikož tato zařízení nejsou primárně určená k přenosu co nejlepší kvality obrazu a musí minimálně zatěžovat operační systém, využívají velké komprese dat a to většinou algoritmem H264, nebo starším MPEG-1 či MPEG2.

Web kamery mívají poměrně kvalitní senzor typu CMOS. Jejich použitelnost pro přesná měření, ale nejvíce znehodnocuje optická soustava a algoritmy komprese. Optická soustava je tvořena jednoduchým plastovým výliskem s jednou čočkou reprezentující objektiv. Aperturová korekce, zvýraznění hran obrazu, je též problém, který znemožňuje využít tuto levnou kameru k přesnému měření.

### IP kamery

IP kamery jsou typem kamer využívající se převážně v zabezpečovací technice. Mohou být vybaveny jak senzory CCD tak CMOS. Lze je vzdáleně ovládat, napájet po ethernetu a v případě připojení do sítě internet sledovat jejich výstup kdekoliv na světě. Přestože datová propustnost např. GSM modemů, nebo v dnešní době hojně rozmáhané sítě 3G je poměrně vysoká, nelze načtená data přenášet v plné kvalitě.

U přenosu dat tohoto typu kamer hrají především roli trochu jiné vlastnosti než u usb nebo firewire. První z nich je doručitelnost kompletních dat, opoždění vůči realitě a zmiňovaná šířka pásma pro přenos. Vyžaduje-li situace kontinuální přenos s rychlostí 25fps, v plném rozlišení, lze vhodně využít např. http stream, nebo udp stream.

Http streaming funguje na principu přenosu dat po blocích. Blok dat je vždy přenesen najednou a lze nastavit velikost až v řádu sekund. Jelikož tato forma využívá

standard TCP/IP je zaručena doručitelnost dat. Další nespornou výhodou bývá možnost ovlivňovat některé parametry během přenosu. Například možnost plynule změnit kvalitu videa. Zpoždění vysílání ovšem dosahuje cca 20-60s oproti reálnému času.

UDP stream se vyznačuje svojí minimální latencí vůči snímané scéně. V praxi dosahuje kolem 1-6 sekund. Data se posílají po paketech a to bez zpětné kontroly doručení. Mnohdy tento fakt je vidět např. u broadcastově vysílaných televizí, kdy při ztrátě dat dojde k „rozkostičkování“ obrazu. Paket může mít formát dat RTP nebo RTSP. Tento formát dat byl speciálně navržen pro přenos multimediálních dat po síti internet.

Speciálním formátem, který dnes přichází na scénu je formát RTMP, případně jeho ekvivalent RTMPT. Tento formát využívá předností protokolu TCP/IP a díky speciální serverové části lze dosáhnout velmi nízké latence, která může mít hodnotu zpoždění danou hardwarem spojových bodů. Tento protokol je v dnešní době často využíván spolu s komprimačním algoritmem H.264.

Pro případ, kdy není nutné vidět realtime náhled, ale stačí snímek jednou za pár minut, existuje u těchto kamer funkce „snap shot“, která po zavolání http requestem, vrátí jeden snímek. Tento snímek je ve většině případů ve formátu JPG.

### **Smart kamery (chytré kamery)**

Poslední typ kamer se příliš neliší až tak rozhraním, ale přístupem kamery k datům. Smart kamery jsou většinou konstruovány ke speciálním účelům. Mohou to být kamery určené pro snímání rychlých scén, např. balistická analýza, nebo naopak kamery s co nejvyšší kvalitou pořízeného snímku. Mohou být vybavené senzorem CMOS nebo CCD.

Při svém měření využijí především smart kameru, která byla vyvinuta na katedře měření ČVUT v Praze. Tato kamera disponuje rozhraním USB 2.0 a lze s ní vyčítat snímky v nekomprimovaném formátu a umožňuje nastavit vlastnosti senzoru díky speciálním registrům.

## **1.2.3 Kompresní algoritmy a prezentace dat obrázků**

Způsobů komprese obrazových dat existuje několik typů. Některé využívají jednoduché principy komprese, kde každý frame je kódován. Ty inteligentnější analyzují předchozí data a pomocí pravděpodobnosti nebo statistiky určují zda došlo ke změně v celém obrazu či jen v jeho části. Příkladem může být kodek H.264, který ve snímané scéně detekuje pouze změny, které nastaly a po několik následujících framů přenáší pouze tyto změny. Tato vlastnost výrazně snižuje datový tok. Nevýhodou je případná ztráta klíčového snímku, bez kterého nelze data obnovit.

JPEG (Point Photographic Expert Group) se řadí mezi ztrátové algoritmy. Pracuje na principu rozdělení obrázku na bloky dat po 8x8 prvcích. Na tyto bloky je aplikována diskrétní Kosinova transformace. Před samostatným spuštěním komprese máme možnost nastavit její úroveň. Algoritmus postupuje dle úrovně od nejhrubších detailů po ty nejméně viditelné. Data jsou reprezentována 8 bitově s použitím barevných profilů sRGP nebo Adobe RGB.

BMP (Windows Bitmap) nebo také DIB (device-independent bitmap) je formát obrazových dat, který umožňuje rastrovou reprezentaci obrazového souboru. Lze s ním

uložit digitální snímek o libovolné šířce, výšce, rozlišení, černobílé i barevné snímky v různých barevných hloubkách s volitelným kompresním poměrem dat, alfa kanálů a barevných profilů. Jeho vnitřní strukturu reprezentuje tabulka 1.2

Název stuktury	Velikost	Účel
Bitmap File Header	14B	Uložení obecných informací o souboru signatura, velikost souboru, offset
DIB Header	7 typů velikostí	Podrobné informace o obrázku Šířka, výška, počet bitů na pixel, komprese, velikost obrázku, color bitmask, alfa bitmask, gamma pro RGB barvu a další...
Color Table	Dle použitého formátu	Zde je možné nadefinovat vlastní barevnou paletu, která je použita na prezentaci obrázku
Image Data		Skutečné hodnoty pixelů v obrázku
ICC Color Profile		Zde může být nadefinovaný barevný profil pro správu barev. Lze dokonce i zadat cestu k externímu souboru.

Tab. 1.2 Struktura BMP souboru reprezentující obrazová data

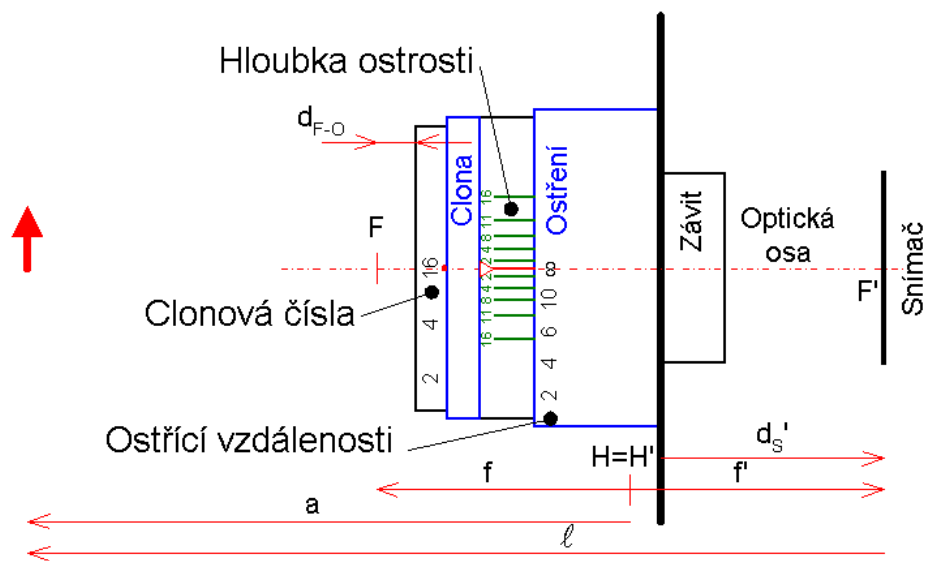
Díky jednoduchosti a mnohotvárnosti byl formát Bitmapy zvolen jako referenční pro ukládání náměrů v této práci.

### 1.3 Objektiv, struktura a jeho vliv na zobrazení obrazu

Objektiv sám o sobě je nedílnou součástí každé videokamery nebo fotoaparátu. Jeho účelem je soustředit shluk paprsků světla na plochu senzoru, zaostřit ji nebo přiblížit.

Objektivů máme několik typů. Základní rozdělení je na rybí oka (objektiv s úmyslnou deformací perspektivy), širokoúhlý (objektiv pro snímání a zobrazování krajiny), normální (objektiv s ohniskovou vzdáleností 50mm, což odpovídá zornému úhlu lidského oka), teleobjektivy a makroobjektivy (objektivy s měřítkem 1:1 a s minimální zaostřovací vzdáleností).

Typická konstrukce základního objektivu je zobrazena na obrázek 1.8. Skládá se z těla objektivu, závitu, ostření a nastavení clony.



Obr. 1.8 Náhled konstrukce objektivu.  $F$  a  $F'$  ohniska,  $f$  a  $f'$  ohniskové vzdálenosti, „ $a$ “ vzdálenost předmětu od hlavní roviny, „ $l$ “ zaostřovací vzdálenost (zdroj [8])

Parametry udávané výrobcem často bývají ohnisková vzdálenost (označuje se  $f'$  a vyjadřuje vzdálenost objektivu od snímače) a clonové číslo, jinými slovy světelnost (určuje kolik světla je objektiv schopen dopravit k senzoru). Definice clonového čísla vychází z poměru ohniskové vzdálenosti ku průměru maximálně otevřené clony objektivu

$$\frac{1}{k} = \frac{D_{VP}}{f'} \quad (1.1)$$

kde  $D_{VP}$  - průměr otvoru clony,  $f'$  - ohnisková vzdálenost,  $k$  - clona.

Clonová čísla jsou na objektivu volena dle geometrické řady tak, aby následující vyšší číslo znamenalo dopad polovičního optického výkonu na senzor.

Zvětšení určuje kolikrát se zvětší předmět před objektivem při promítnutí na plochu senzoru. Lze vyjádřit vztahem

$$\beta = \frac{f'}{z} \quad (1.2)$$

kde  $\beta$  - je zvětšení,  $f'$  - ohnisková vzdálenost,  $z$  - vzdálenost předmětu od předmětového ohniska.

Pro standardní objektiv musí platit Gaussova zobrazovací rovnice

$$\frac{1}{f'} = \frac{1}{a} + \frac{1}{a'} \quad (1.3)$$

Objektiv lze považovat za jednu z nejdůležitějších součástí celého měřicího systému. Díky vhodné volbě objektivu a jeho dobré kvalitě zpracování lze mnohonásobně zvýšit přesnost měření. Pro měření jsou zcela nevhodné objektivy s měnitelným rozsahem zvětšení, nebo objektivy s malou světelností.

Pro vyjádření specifických vlastností optických soustav a objektivů lze využít OTF (Optical Transfer Function) přenosovou funkci. OTF funkce vyjadřuje kontrast výsledného obrazce, který se objektivem snímá, ku předloze.

$$OTF(fs) = \frac{K_{0\_vyst}}{K_{0\_vstup}} \quad (1.4)$$

Předloha bývá ve většině případů složená z čárového obrazce. V případě nízkého kontrastu nelze čárové obrazce vůbec rozeznat a slijí se v šedou plochu. Tato optická vada je nejméně patrná ve středu objektivu a nejvíce se objevuje v krajních polohách snímku.

Vady objektivu způsobující chyby měření :

- Aberace - vada vzniká nestejným lomem světelných paprsků v optické soustavě. Světelné paprsky se lomí více na okraji čočky než poblíž optické osy. Z tohoto důvodu je lepší měřit objekt poblíž geometrické osy objektivu.
- Vinětace – pokles intenzity ozáření do krajů obrazového pole. Vinětace roste s klesajícím clonovým číslem.
- Difrakce – ohyb záření na kruhovém otvoru. Projevuje se při velkém zaclonění objektivu a to změnou intenzity záření u okrajů zobrazeného objektu.
- Tangenciální zkreslení optické soustavy – tato chyba vznikne pokud není absolutně přesně umístěný objektiv vůči obrazovému senzoru. Tuto chybu lze očekávat u spíše levných kamer a u kamer vlastní výroby, kdy nemusíme dosáhnout přesné pozice umístění.

### 1.3.1 Další zdroje chyb ovlivňující přesnost měření

Optické záření umělých zdrojů osvětlení (žárovka, zářivka)

Tyto zdroje záření generují pro lidské oko nepostřehnutelné blikání o frekvenci 100Hz (doba periody 10ms) a některé zářivky i menší frekvenci a to 50Hz. Senzory typu CMOS, kde se používá elektronická závěrka, způsobí tato frekvence odlišnou expozici jednotlivých řádků. Ta se projevuje zkreslením obrazu typu roleta, žaluzie a v obraze se vytvoří orientované světlé nebo tmavé pruhy. Pro odstranění tohoto efektu je nutné nastavovat dobu expozice v násobku periody zvlnění.

Gama korekce

Některé kamery mají speciální funkci pro automatický převod jasových hodnot pixelů podle tzv. gama křivky. Jedná se o úpravu obrazu pro zobrazení na klasických monitorech a televizích. Pro měření je dobré, tuto korekci vypnout.

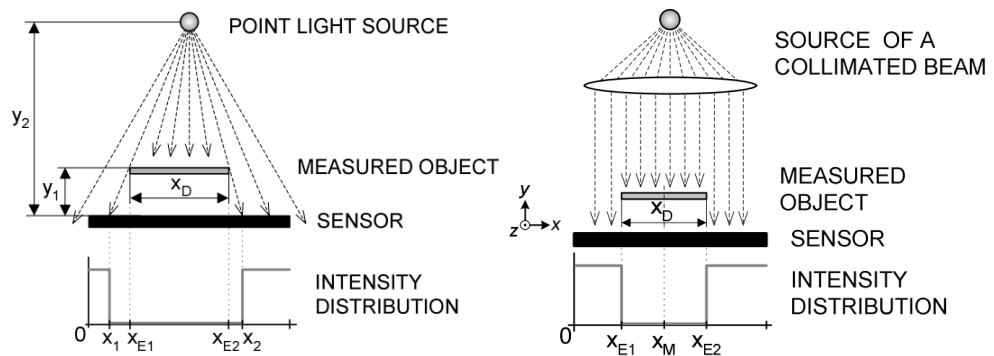
## 1.4 Metody optického měření rozměru objektu

Pro měření rozměrů objektu lze aplikovat dvě základní metody. Jedna ve své měřicí soustavě používá projekční objektiv a druhá využívá senzor přímo jako měřicí nástroj.

### 1.4.1 Měření bez projekčního objektivu

Bez projekčního objektivu lze systém přirovnat k analogii měření pomocí pravítka. Objekt přiložíme na pravítko a z určité vzdálenosti čteme hodnotu rozměru ze stupnice. V případě, že pravítko nahradíme senzorem (CCD nebo CMOS), místo pozorovatele dáme zdroj bodového záření a snímáný objekt umístíme v malé vzdálenosti od senzoru, dojde k promítnutí obrysu objektu na plochu senzoru.

V této metodě je ještě nutné zvážit fakt, zda-li se jedná o bodový zdroj záření, nebo je použit kolimátor. Tuto problematiku blokově zobrazuje obrázek 1.9.



Obr. 1.9 Blokové schéma principu měření s bodovým zdrojem záření nebo s kolimátorem (zdroj [9])

V případě použití kolimátoru je otisk měřeného objektu roven velikosti objektu. Pokud ovšem použijeme bodový zdroj záření, dochází (v závislosti na vzdálenosti zdroje a měřeného objektu od senzoru) ke zvětšení otisku. Skutečný rozměr objektu je roven  $x_{E1}, x_{E2}$ , změřený rozměr senzorem je roven  $x_1, x_2$ . Uvažujeme-li umístění bodového zdroje záření uprostřed senzoru, můžeme velikost těchto rozdílů vyjádřit rovnicemi

$$\begin{aligned} \frac{L - x_1}{y_2} &= \frac{x_{E1} - x_1}{y_1} \\ \frac{x_2 - L}{y_2} &= \frac{x_2 - x_{E2}}{y_1} \end{aligned} \quad (1.5)$$

kde  $2L$  je šířka senzoru,  $x_1$  naměřený rozměr senzorem,  $x_{E1}$  skutečná velikost objektu,  $y_1$  vzdálenost objektu od senzoru,  $y_2$  vzdálenost zdroje záření od senzoru,  $x_2, x_{E2}$  je analogicky totožný  $x_1, x_{E1}$ .



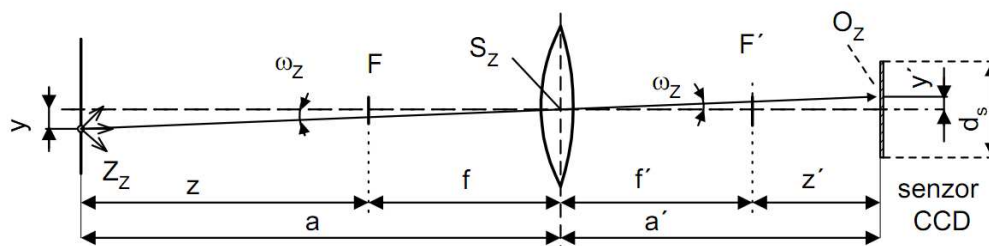
Výhodou měření bez projekční soustavy je odstranění veškerých chyb měření způsobených geometrickými vadami objektivu.

S projekčním objektivem je situace odlišná. Projekční objektiv způsobí lom paprsků vycházejících z bodového zdroje záření do bodu umístěného v ohniskové vzdálenosti. V ostatních polohách je zobrazen kroužek místo bodu. Pro měření s projekčním objektivem je nutné znát jeho ohniskovou vzdálenost. Z této hodnoty lze dopočítat zvětšení, pomocí kterého můžeme přepočítávat velikost pixelů z obrazu na skutečný rozměr objektu.

Měřicí soustavu s projekčním objektivem zobrazuje obrázek 1.10. Zde platí vztah pro výpočet zvětšení

$$-\beta = \frac{y'}{y} = \frac{a'}{a} \quad (1.6)$$

$\beta$  představuje zvětšení obrazu (záporné znaménko symbolizuje převrácení obrazu oproti předmětu),  $y'$  velikost předmětu na senzoru,  $y$  velikost skutečného objektu,  $a$  vzdálenost předmětu od osy čočky,  $a'$  vzdálenost senzoru od osy čočky.



Obr. 1.10 Blokové schéma měřicí soustavy s projekčním objektivem (zdroj [8])

### 1.4.2 Výpočet rozměrů objektu s projekční soustavou

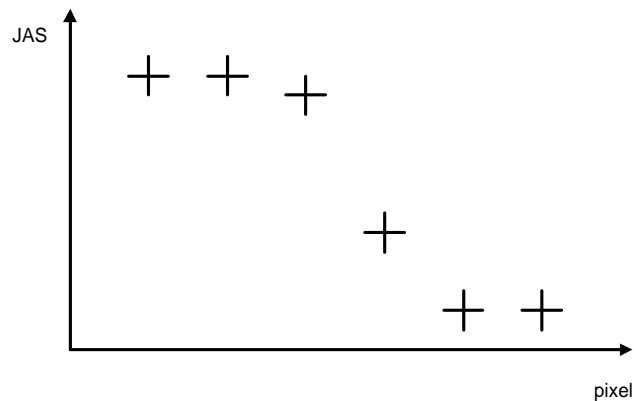
Objektiv ohnisková vzdálenost:	16mm
Vzdálenost předmětu od předmětové vzdálenosti objektivu:	53cm
Senzor má rozměr pixelu :	5,2um

$$\text{Zvětšení objektivu } \beta = \frac{f'}{z} = \frac{16}{530} = 0,030 .$$

Podle tohoto zvětšení můžeme určit velikost 1 pixelu na objektu. Tady v tomto případě je rovna  $\frac{1}{30}$  pixelu na senzoru. Můžeme tedy konstatovat, že posun objektu o 1mm vyvolá posun obrazu na senzoru o 30um.

## 1.5 Metody detekce polohy hrany objektu

Existuje několik metod určení přesné polohy hrany. Většina se ovšem zakládá na matematických předpokladech průběhu přechodu hrany. Při pořízení snímku hrany a vykreslení jasových hodnot jednoho řádku, obrázek 1.11, získáme obrazovou informaci o tom jaký pixel je ještě součástí hrany a který ne. Tímto jsme tedy schopni určit s přesností jednoho pixelu pozici hrany.



Obr. 1.11 Graf přechodu hrany jednoho řádku

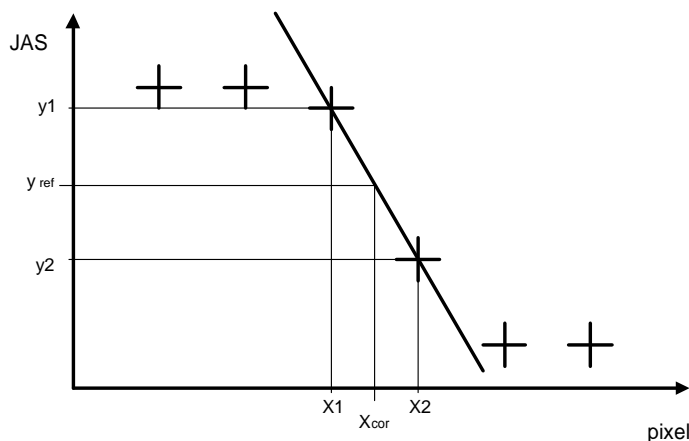
Zpřesnit naše měření můžeme proložením dvou nebo více bodů ležících na hraně měření. Otázkou však zůstává zdali námi zvolená metoda proložení dostatečně přesně charakterizuje skutečný průběh hrany.

Mě i vedoucímu práce nejsou doposud známi žádné dostupné náměry, které by dokazovaly jaký má hrana skutečný průběh a jak velkou chybu do ní vnáší ony metody prokladu.

Jedním z cílů moji práce je přesné změření přechodu hrany a tím i stanovení co nejvyšší hranice dosažitelné přesnosti měření.

### 1.5.1 Určování polohy hrany lineární interpolací

Jedna z jednoduchých metod pro určení pozice hrany ze snímku se nazývá lineární interpolace. Jedná se o nalezení dvou bodů, co nejbližší ke komparační úrovni, která se volí ve většině případu na hodnotu 50% rozsahu signálu. Správná hodnota by ovšem měla díky lomu světla na hranách objektu činit 25%. Lepší představu o této metodě ukáže následující obrázek 1.12.



Obr. 1.12 Lineární interpolace

Komparační úroveň v našem případě je dána vztahem

$$y_{ref} = \frac{1}{2} * (U_H - U_L) \quad (1.7)$$

kde  $U_H$  je horní mez rozsahu signálu,  $U_L$  dolní mez rozsahu signálu

Pro výpočet interpolované hodnoty použijeme vztah

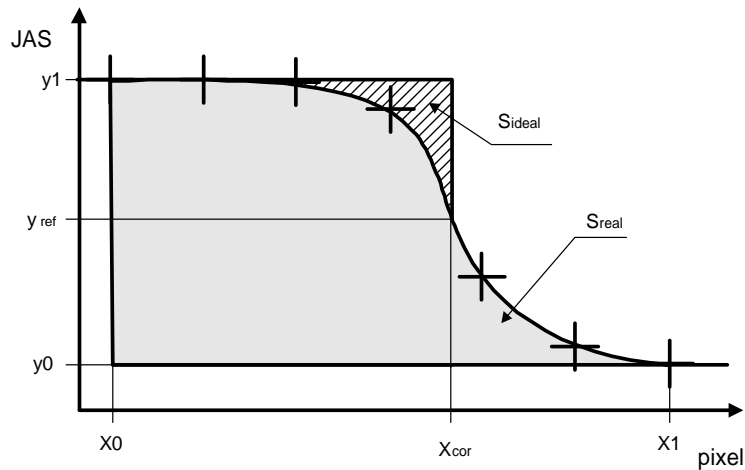
$$x_{cor} = x_1 + \frac{y_{ref} - y_1}{y_2 - y_1} * 1 \quad (1.8)$$

kde  $x_{cor}$  hodnota pozice interpolované hrany,  $x_1, y_1$  souřadnice bodu na komparační úrovni,  $x_2, y_2$  souřadnice bodu pod komparační úrovní.

V některých případech se může stát, že přechod jasového profilu hrany není z bílé do černé, jak je zobrazen na obrázku 1.11, ale může být úplně obrácený a to z černé do bílé. V tomto případě lze použít stejný vzorec jako je uvedený v rovnici 1.8, při zachování jako prvního v pořadí bodu  $x_1, y_1$ , a druhého bodu  $x_2, y_2$ .

## 1.5.2 Určování polohy hrany fotometrickou interpolací

Další metoda určování polohy hrany zakládá na rovnosti obsahu ideálního průběhu přechodu hrany a plochy reálného průběhu přechodu hrany. Princip je zobrazen na obrázku 1.13. Důležité u této metody je zvolení vhodných parametrů pro výpočet obsahu.



Obr. 1.13 Fotometrická interpolace

Výpočet plochy ideálního a reálného přechodu se musí rovnat. Tento vztah lze popsat rovnicí

$$(x_{cor} - x_0) * (y_1 - y_0) = \int_{x_0}^{x_1} (y(x) - y_0) dx \quad (1.10)$$

poté výpočet pozice hrany vyjádřená  $x_{cor}$  je

$$x_{cor} = x_0 + \Delta x * \frac{\sum_{n=x_0}^{x_1} (y[n] - y_0)}{y_1 - y_0} \quad (1.11)$$

kde  $\Delta x$  je v našem případě roven 1.

Tento vztah platí pro přechod hrany z bílé do černé. Pokud se však jedná o přechod hrany obrácený z černé do bílé – použijeme modifikovaný vztah

$$x_{cor} = x_1 - \Delta x * \frac{\sum_{n=x_0}^{x_1} (y[n] - y_0)}{y_1 - y_0} \quad (1.12)$$

## 1.6 Určování pozice objektů a jejich vzdálenosti s využitím polohy těžiště

Pro určení pozice nebo změny pozic objektů ve 2D prostoru se nejlépe osvědčila metoda nalezení těžiště. Obecný výpočet je dán vztahem

$$x_T = \frac{\sum mi * xi}{m}, y_T = \frac{\sum mi * yi}{m}, z_T = \frac{\sum mi * zi}{m} \quad (1.13)$$

kde  $x_T, y_T, z_T$  jsou souřadnice těžiště,  $m$  je hmotnost celého tělesa,  $mi$  je hmotnost  $i$ -té části tělesa a  $xi, yi, zi$  poloha těžiště v  $i$ -té části.

Rovnice můžeme upravit pro naše účely do 2D prostoru a to následujícím způsobem

$$x_T = \frac{\sum Ii * xi}{I}, y_T = \frac{\sum Ii * yi}{I} \quad (1.14)$$

kde  $x_T, y_T$  jsou souřadnice těžiště,  $I$  je součet všech jasových hodnot zahrnutých v měření,  $Ii$  je jasová hodnota  $i$ -té části objektu a  $xi, yi$  je poloha těžiště v  $i$ -té části.

Pro výpočet dosažitelné rozlišitelnosti měření, nebo zjištění vzdálenosti dvou bodů musíme ve výpočtech uvážit fakt, že nikdy nejsme schopni nastavit pojezd takovým způsobem, abychom objektem pohybovali vždy pouze v jednom směru osy. Z tohoto důvodu je nutné počítat s možností pohybu nejen ve sloupcích, ale i v řádcích senzoru.

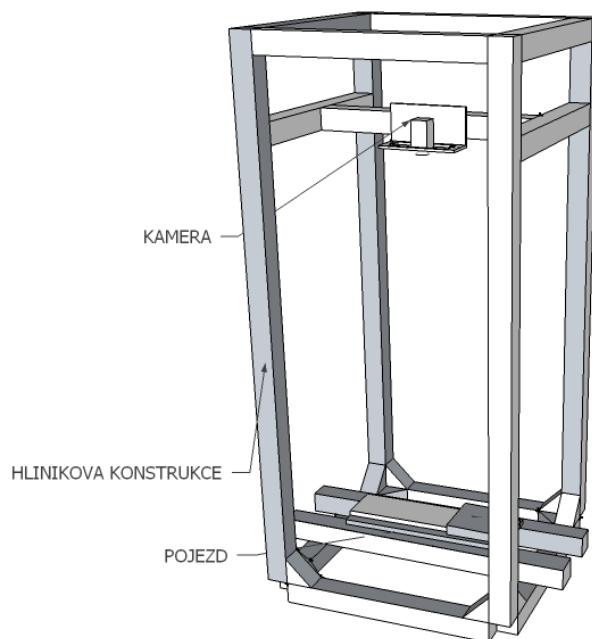
Pro výpočet vzdálenosti dvou těžišť použijeme upravený vztah pro výpočet vzdálenosti dvou bodů v rovině. Body jsou zadány souřadnicemi těžiště.

$$d(AB) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1.15)$$

kde  $d(AB)$  je vzdálenost dvou těžišť,  $x_1, y_1$  jsou souřadnice prvního těžiště a  $x_2, y_2$  souřadnice druhého těžiště.

## 2 NÁVRH A REALIZACE MĚŘÍCÍHO PRACOVISTĚ

Pro stabilní a opakovatelné pořízení náměrů bylo nutné mít kameru pevně spojenou s konstrukcí pojezdu. Tímto by měla být zachována dobrá konstrukční tuhost. Dále konstrukce vyžadovala upevnění kamery jak v horizontálním tak vertikálním směru, volný přístup k měřenému objektu a snadnou výměnu pojezdového zařízení. Návrh konstrukce ukazuje obrázek 2.1

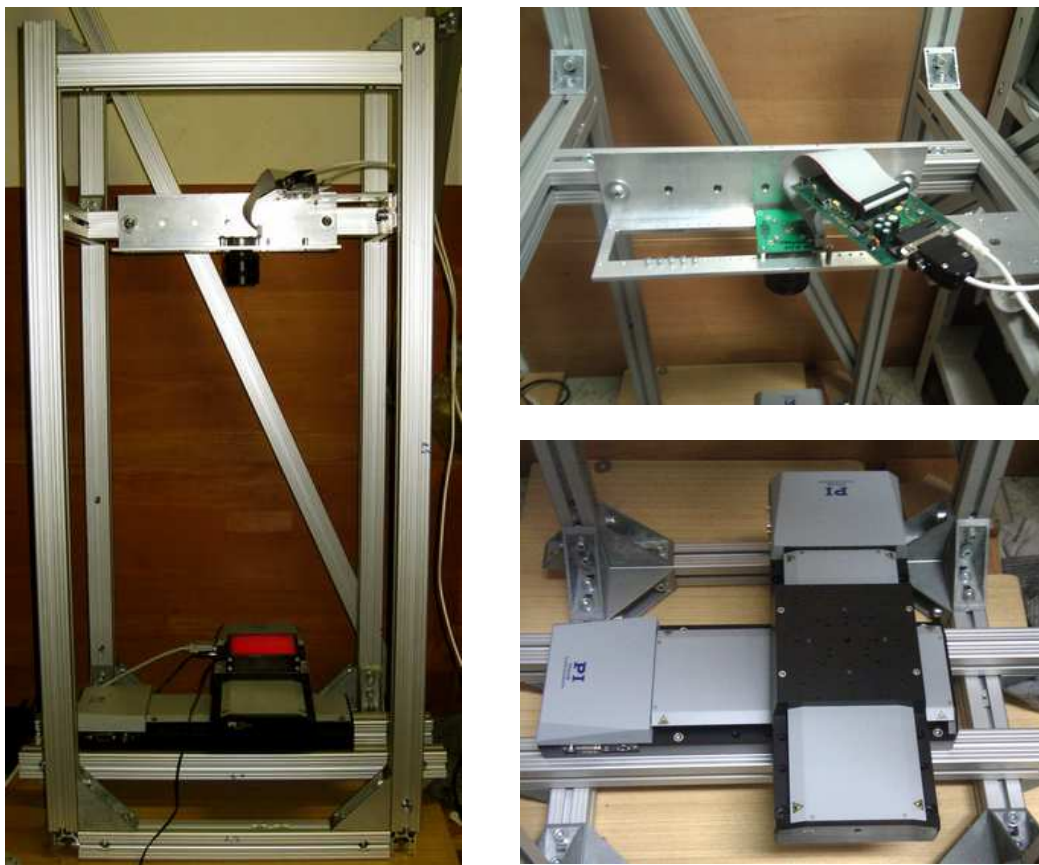


Obr. 2.1 Návrh konstrukce měřicí soustavy

### 2.1 Konstrukce měřicí soustavy

Jako konstrukční materiál byly zvoleny hliníkové profily od společnosti MayTec. Tyto profily se vyznačují přesným drážkovaním a standardizovanými velikostmi. Výrobce též dodává i přesný spojovací materiál jako jsou např. pravouhlé rohy, což výrazně urychlí stavbu.

Jako tvar konstrukce byl zvolen kvádr. Základna má rozměr 480x480mm a na výšku 1000mm. Pro lepší mechanickou tuhost byly přidány dvě vzpěry. Spodní část konstrukce umožňuje ukotvení např. do betonu nebo ke zdi. Hotovou konstrukci můžeme vidět na obrázku 2.2. Na obrázku vlevo je zobrazen celý monolit konstrukce s upevněnou kamerou připravenou k měření. Vpravo nahoře je zobrazen speciální držák pro kamery typu Visor, který umožňuje variabilní posun a otáčení kamery dle aktuální potřeby. Snímek vpravo dole ukazuje namontovaný osový kříž velmi přesných pojezdů, na kterých byla provedena finální měření.



Obr. 2.2 Fotodokumentace konstrukce měřící soustavy

Největší slabinou této konstrukce je typ materiálu, ze kterého je postavena. Hliník se vyznačuje velmi velkou délkovou roztažností a pokud chceme měřit v řádu tisícín milimetrů musíme tento fakt vzít na vědomí.

Z rovnice délkové roztažnosti kovu si určíme jak velká změna nastane při změně teploty o  $1^{\circ}\text{C}$ . Vztah pro délkovou roztažnost je

$$\Delta l = \alpha * l_0 * \Delta T \quad [m] \quad (2.1)$$

kde  $\Delta l$  je změna délky profilu,  $\alpha$  součinitel teplotní délkové roztažnosti,  $l_0$  délka materiálu,  $\Delta T$  změna teploty.

Délková roztažnost hliníku činí  $2,4 * 10^{-5} / \text{K}^{-1}$ . Po dosazení

$$\Delta l = 2,4 * 10^{-5} * 1 * 1 = 24 * 10^{-6} m$$

nám vyjde, že konstrukce změní svoji délku o  $24 \mu\text{m}$ .

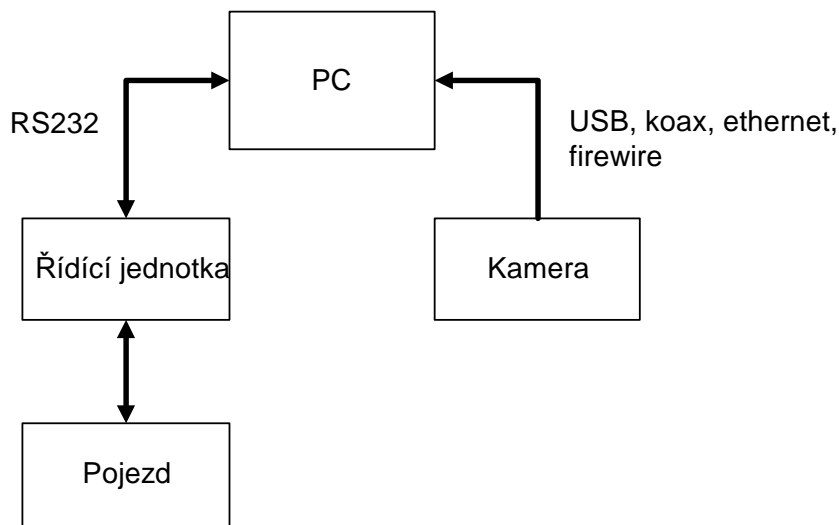
Pro změnu rozměru do šířky

$$\Delta l = 2,4 * 10^{-5} * 0,4 * 1 = 9,6 * 10^{-6} m$$

získáme výsledek změny o  $9,6\mu m$ .

Předběžně odhadnuté výsledky měření hovoří o přesnosti měření na  $\frac{1}{10}$  až  $\frac{1}{100}$  pixelu. Při takové přesnosti už nelze tuto věc brát na lehkou váhu a je třeba konstrukci mít umístěnou v místnosti s minimálním teplotním výkyvem se mít zajištěný konstantními vztažnými podmínkami.

## 2.2 Blokové schéma zapojení měřící soustavy



Obr. 2.3 Blokové schéma měřící soustavy

Na obrázku 2.3 je blokově znázorněno zapojení měřící soustavy. Skládá se z PC, kamery, řídicí jednotky pojezdu a pojezdu. Programové vybavení běžící na PC sbírá data z kamery a pomocí řídicí jednotky reguluje velikost posuvu pojezdu. Komunikace mezi kamerou a PC je u analogových kamer jednosměrná. Pokud máme připojenou kameru Visor lze nastavovat její parametry pomocí speciálních příkazů.

Řídící jednotka pojezdu je připojena pomocí sběrnice RS232 nebo převodníku USB/RS232.

## 2.3 Softwarové vybavení pracoviště

Před popisem funkčnosti je nutné uvést terminologii, se kterou se setkáme v dalším textu.

**Snímek** – jedná se o jeden obrázek, který lze zobrazit v jakémkoliv grafickém programu.

**Náměr** – jedná se o jeden cyklus měření od načtení pozice až po její inkrementaci. Může obsahovat jeden nebo více snímků, které se pak mohou průměrovat.



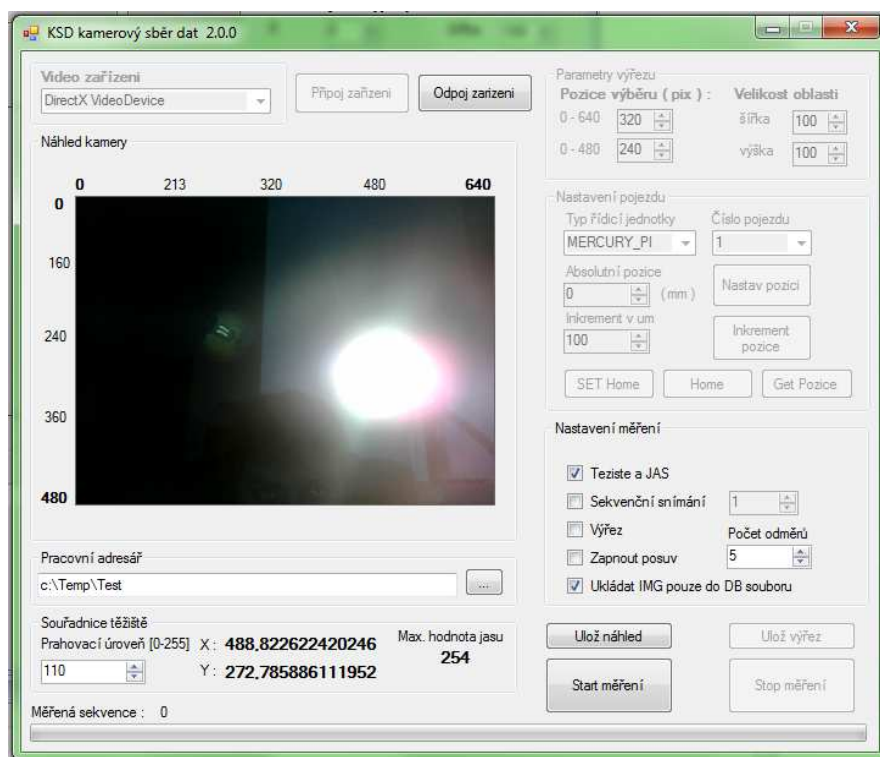
Softwarové vybavení pracoviště lze rozdělit na tři samostatné programy. Software pro sběr dat z kamer, server obsluhující řízení pojezdu a software pro vyhodnocení měření – případně algoritmy použitelné v prostředí Matlab.

Poslední zmiňovaný program se nachází stále ve stavu pracovní verze, kdy pro určité specifikace vyhodnocení bylo ve většině případů nutné zasahovat do zdrojových kódů a tedy neexistuje univerzální řešení zadaného problému.

### 2.3.1 KSD - Software pro sběr dat z kamery

Aplikace slouží k připojení datového rozhraní videokamer do měřicí soustavy. Umožňuje připojit standardní videozařízení systému Windows, jako jsou webkamery, videopřevodníky nebo speciální grabovací karty, dále rozhraní smart kamery Visor, která byla vyvinuta na katedře měření ČVUT [11 a 12] a v neposlední řadě IP kameru firmy Bosh zapůjčenou v rámci jiné diplomové práce.

Na obrázku 2.4 je zobrazeno základní rozložení ovládacích prvků aplikace. Skládá se z osmi částí. Video zařízení, Parametry výřezu, Nastavení pojezdu, Nastavení měření, Pracovní adresář, Souřadnice těžiště, Náhled kamery, sekce pro spuštění a vypnutí měřicího cyklu.

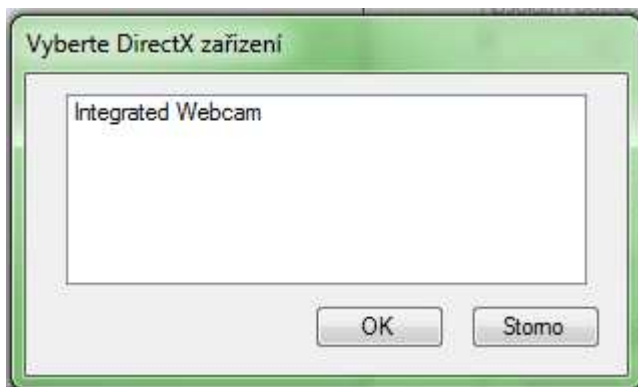


Obr. 2.4 Screen shot základního zobrazení aplikace KSD pro náměr obrazových dat

#### Video zařízení

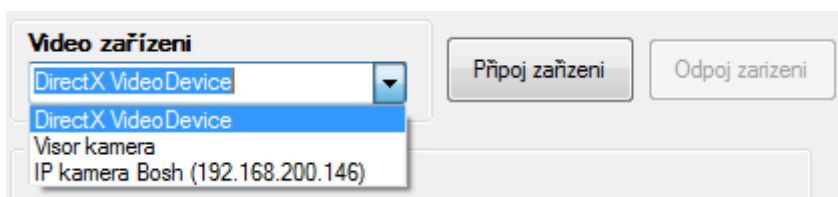
Tato sekce obsahuje seznam předem před-instalovaných modulů jednotlivých typů

videozařízení, s kterými umí tato aplikace pracovat. Tlačítkem „Připoj zařízení“ dojde k detekci připojených zařízení (obrázek 2.5) a po výběru začne generovat obrazový náhled přijatých dat.



Obr. 2.5 Screen shot – výběr video zařízení – seznam připojených video zařízení k PC

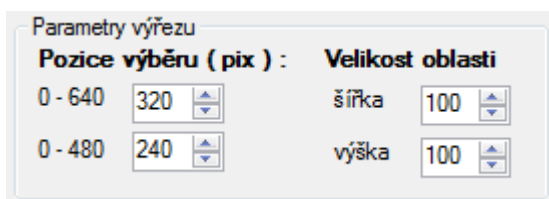
Data jsou vždy převedena do monochromatického formátu a to jak v náhledu tak i při ukládání. Tlačítko „Odpoj zařízení“ ukončí veškeré procesy a vlákna přivlastněné aplikaci a vnitřně uvede aplikaci do defaultního stavu. Možnosti volby zařízení ukazuje obrázek 2.6



Obr. 2.6 Screen shot – sekce Video zařízení – výběr vstupního zařízení

### Parametry výřezu

Aplikace umožňuje ukládat a zpracovávat oblasti zobrazovaného snímku. Výhodou je malé množství ukládaných dat. Sekce se zpřístupní po zaškrtnutí volby „Výřez“ v sekci „Nastavení měření“. Poté se v náhledu vygeneruje žlutý obdélník, který označuje oblast výřezu, se kterou aplikace pracuje. Obrázek 2.7 ukazuje možnosti nastavení. „Pozice výběru“ určuje pozici levého horního rohu obdélníku. „Velikost oblasti“ určuje plochu výřezu. Rozsah hodnot pro pozici výběru je generován automaticky dle nastavení příslušené kamery nebo přijatých dat.



Obr. 2.7 Screen shot – sekce Parametry výřezu – nastavení pozice a velikosti výřezové oblasti

## Nastavení pojezdu

Program umožňuje vzdálené ovládání serverového modulu pro řízení pojezdu. V této verzi programu KSD 2.0.0 je nastavena adresa serveru na localhost (127.0.0.1) port 6001. Jeho možnosti nastavení zobrazuje obrázek 2.8

Implementovány jsou dva typy řídicích jednotek a to pro jednotku MERCURY PI a MARSII. Ovládání těchto jednotek a bližší informace jsou uvedeny v kapitole 3.3.1 a 3.3.2. „Typ řídicí jednotky“ přiřazuje konkrétní typ pojezdu. „Číslo pojezdu“ určuje adresu řídicí jednotky. V případě, že je vybrána jednotka MARSII čísla reprezentují znakové označení adres a to 1 = A, 2 = B, 3 = C.

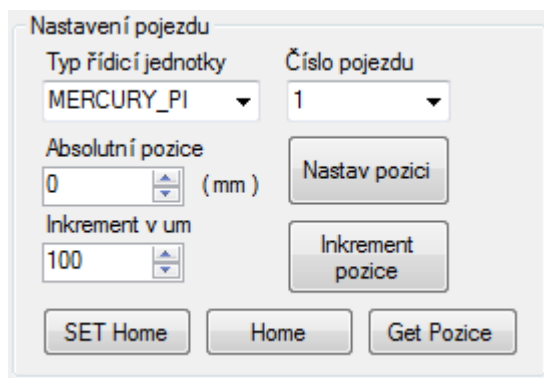
Tlačítko „Nastav pozici“ umožňuje posunout pojezd na hodnotu zadanou v kolonce „Absolutní pozice“ (lze používat i záporné hodnoty). Hodnoty jsou brány v milimetrových jednotkách.

Tlačítko „Inkrement pozice“ způsobí posunutí pojezdu o relativní hodnotu zadanou v kolonce „Inkrement v um“. Hodnoty jsou zadávány v mikrometrech a lze užít i záporné hodnoty. Tato **hodnota je používána i pro cyklický odměr dat**.

Tlačítko „SET Home“ vynuluje aktuální počítadlo polohy v řídicí jednotce posuvu a danou polohu označí jako výchozí (HOME). Doporučuji používat pro jednotné náměry a jejich jednodušší zpracování.

Tlačítko „Home“ – vrací pojezd do výchozí pozice

Tlačítko „Get Pozice“ – vyčte z pojezdu jeho aktuální pozici



Obr. 2.8 Screen shot – sekce Nastavení pojezdu – možnosti ovládání

## Nastavení měření

Zde se nachází hlavní část pro nastavení celého procesu měření. Umožňuje nastavit různé varianty pořizování náměrů. Obrázek 2.9

Těžiště a JAS – Zprovozní se detekce polohy těžiště a nejvyšší jasové hodnoty ve vyobrazeném snímku. Podle této hodnoty si můžeme zkorigovat velikost clony na objektivu, abychom nepořizovali snímky v saturaci.

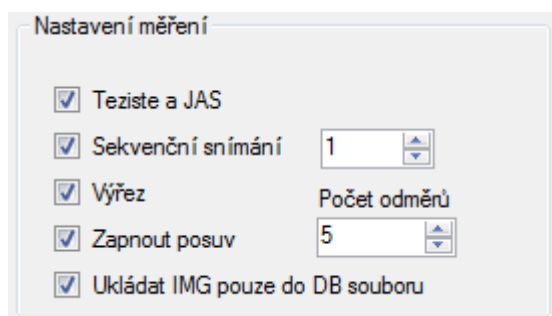
Sekvenční snímání – Zaškrtnutím této volby můžeme pořizovat více snímků na jeden krok pojezdu. Počet pořizovaných snímků určuje hodnota uvedená v sousední buňce. Této vlastnosti užíváme při nutnosti průměrování náměrů pro odstranění šumu.

Výřez – Touto volbou aktivujeme možnost uložení pouze určité části, která se vyčlení v náhledu pomocí žlutého obdélníkem.

Zapnout posuv – Aktivace je nutná pro zpřístupnění pojezdu. Není podmínkou pro spuštění procesu měření a ukládání snímků.

Ukládat IMG pouze do DB souboru – Zde si můžeme navolit možnost ukládat pořízené snímky i do jednotlivých souborů, nebo pouze do DB souboru. V případě nezaškrtnutí této volby se snímky ukládají jak do Bitmapových tak i do databázového souboru.

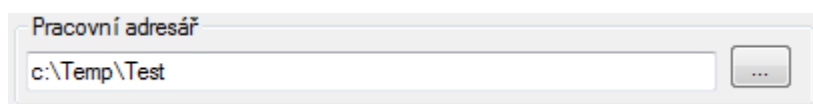
Počet odměrů – zde nastavujeme počet odměrů.



Obr. 2.9 Screen shot – sekce Nastavení měření – možnosti nastavení měřicího cyklu

### Pracovní adresář

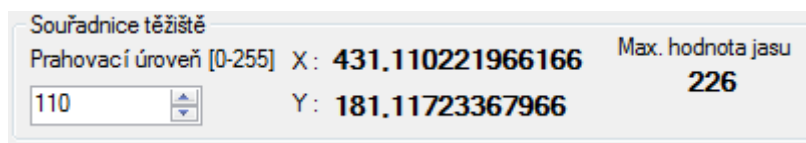
Pro ukládání náměrů je možnost zvolit jakékoliv umístění. Při každém spuštění cyklu měření dochází ke kontrole, zda-li zadaný adresář neexistuje. V případě, že ano, vytvoří se adresář s přívlastkem. Tato funkce byla aplikována pro ochranu přepsání již naměřených dat. Nastavení ukazuje obrázek 2.10.



Obr. 2.10 Screen shot – sekce Pracovní adresář – volba umístění naměřených dat

### Souřadnice těžiště

Na obrázku 2.11 je znázorněn vizualizační nástroj pro hrubé zjištění umístění těžiště s nastavením prahovací úrovně (threshold). Dále je možné dle maximální hodnoty jasu nastavit kameru tak, aby obrazové hodnoty nebyly v saturaci. Proces zobrazení a výpočtu se aktivuje v sekci „Nastavení měření“.



Obr. 2.11 Screen shot – sekce Souřadnice těžiště – vizualizace polohy těžiště a maximální jasové hodnoty

## Náhled kamery

Zobrazuje náhled kamery. Data jsou zobrazena již po konverzi do škály šedi, případně po výpočtu těžiště. V mnoha případech nelze dosáhnout rychlosti 25fps díky náročnosti těchto operací. Hodnoty rozlišení zobrazené vedle náhledu se dynamicky mění dle aktuálního nastavení kamery a mají pouze informativní charakter pro lepší orientaci v obraze.

### Sekce pro spuštění a vypnutí měřicího cyklu.

Tlačítko „START“ – spouští měřicí cyklus – načtení polohy, pořízení snímku, inkrement polohy.

Tlačítko „STOP“ – přerušuje proces měřicího cyklu. Aktuální odměr je dokončen.

Tlačítko „Ulož náhled“ – umožňuje uložit aktuální náhled do bitmapového souboru

Tlačítko „Ulož výřez“ – umožňuje uložit výřez do bitmapového souboru

## 2.3.2 Software pro řízení pojezdu

Aplikace běží v konsolovém režimu, aby zbytečně nespotřebovávala výkon potřebný pro zpracování dat. Jedná se o modifikaci webového serveru, který pracuje na principu dotaz odpověď.

Při spuštění se načítají konfigurační data ze souboru „config.ini“. V něm jsou uloženy informace potřebné k nastavení portu, na kterém aplikace poslouchá a zařízení, které má obsluhovat. Soubor má formát textového dokumentu.

### Př. konfiguračního souboru :

IP\_PORT:6002

MERCURY\_PI:COM6|1,2,3

MARSII:COM3|A,B,C

„MERCURY\_PI:COM6|1,2,3“ znamená, že server se pokusí připojit řídicí jednotku typu MERCURY na sériovém portu COM6 a bude schopen adresovat zařízení s adresou 1, 2 a 3.

MARSII:COM3|A,B,C znamená, že server se pokusí připojit řídicí jednotku typu MARSII na sériovém portu COM3 a bude schopen adresovat zařízení s adresou A, B a C.

### Příklady možné modifikace souboru config.ini :

IP\_PORT:6002

MERCURY\_PI:NOCONNECT

MARSII:COM3|A,B,C

IP\_PORT:8080  
MERCURY\_PI:COM6|1,2,3  
MARSII:NOCONN

IP\_PORT:65002  
MERCURY\_PI:COM6|1  
MARSII:COM3|A,B  
a další....

### **Http requesty :**

Základními parametry http requestu jsou : typ, address, command, value.

**typ** - nabývá hodnoty MARSII nebo MERCURY\_PI a rozlišuje komu data budou poslána

**address** - v případě MARSII hodnota A, B, nebo C

- v případě MERCURY\_PI hodnot 1, 2, 3 – až 16, dle aktuálního nastavení řídicí jednotky

**command** - inkrement, setRelPozition, setAbsPozition, getPozition

**value** - v případě příkazu inkrement se hodnota udává v mikrometrech

- v ostatních případech v milimetrech a to ve tvaru : 0\_342 ( 0,342mm )

Http requesty mají standardní tvar a to ve formátu

*http:// ip\_adresa : port /? parametr1 = hodn & parametr2 = hodn ....*

### **Příklady použití**

Inkrement pozice :

*http://127.0.0.1:6002/?typ=MARSII&address=A&command=inkrement&value=100*

Nastavení relativní pozice

*http://127.0.0.1:6002/?typ=MARSII&address=B&command=setRelPozition&value=0\_002*

Nastavení absolutní pozice

*http://127.0.0.1:6002/?typ=MARSII&address=C&command=setAbsPozition&value=2\_000*

Vyčtení dat o pozici

*http://127.0.0.1:6002/?typ=MARSII&address=1&command=getPozition*

Každý request, až na getPozition, který je platný a je předán k vykonání, je potvrzen textovou zprávou „OK“. V případě nezdaru nebo chybného parametru zprávou „FAIL“.

V případě getPozition je vrácena hodnota a to ve tvaru celého čísla neboli tzv. unit (jednotek). Přepočítání jednotek na milimetry a mikrometry je následující :

#### MERCURY

1mm = 10000unit

1um = 10unit

#### MARSII

1mm = 1000unit

1um = 1unit

### 2.3.3 Software pro zpracování dat

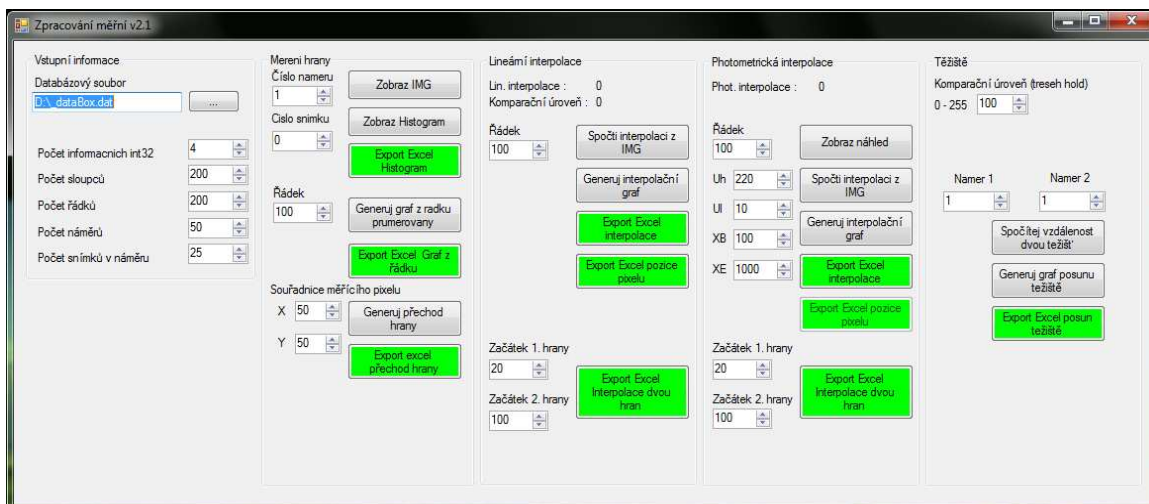
Z hlediska rozmanitosti a modifikace jednotlivých úloh vyhodnocení výsledků, nebylo možné všechnu funkcionalitu naprogramovat do jediné aplikace. Vzniklo tedy hned několik verzí programu a k tomu některé skripty pro zpracování v Matlabu.

Jako první vznikla aplikace pro vyhodnocení naměřených dat se souborovým zpracováním. Jedna z prvotních verzí aplikace KSD neuměla ukládat data do databázového formátu. S každým náměrem se vygeneroval samostatný soubor v jehož názvu bylo zakódováno číslo náměru, pozice pojezdu.

Dalším požadavkem byla nutnost zaimplementovat průměrování a náběr sekvencí snímků. Tímto přibyl další parametr v názvu a značení souboru do tvaru číslo náměru, pozice pojezdu a číslo snímku (sekvence). Zde, už díky velkému množství generovaných souborů, začala být situace neúnosná. V aplikaci často docházelo k pádu i při sebemenší chybě nastavení. Bylo tedy nutné vymyslet jiný formát dat pro uchování pořízených náměrů.

Nejlepší variantou se jeví přímé ukládání obrazových a pomocných dat do jednoho velkého kontejneru. V této práci ho zmiňuji jako databázový formát souboru. Veškeré zmíněné formáty dat jsou podrobně rozepsány v kapitole 2.4.

V této kapitole popíšu pouze poslední verzi aplikace (obrázek 2.12) a možnosti nastavení ovládání. Skripty a algoritmy zpracování budou uvedeny v příslušné sekci kapitoly 3.



Obr. 2.12 Screen shot – aplikace pro Zpracování měření v2.1

## Vstupní informace

### *Databázový soubor*

– cesta k souboru s naměřenými daty, přípona \*.dat

### *Počet informačních int32*

– udává počet informačních bloků před snímkem bližší informace viz. kapitola 2.4 .

### *Počet sloupců*

– počet pixelů ve sloupci – šířka načítaného snímku

### *Počet řádků*

– počet pixelů v řádku – výška načítaného snímku

### *Počet náměrů*

– počet cyklů měření, které obsahuje soubor

### *Počet snímků v náměru*

– počet snímků v jednom náměru – tyto snímky se při vyhodnocení průměrují

## Měření hrany

### *Zobraz IMG*

– zobrazí reálný náhled snímku z náměru. Lze použít pro kontrolu obrazových dat, zda-li vyhodnocujeme správné náměry

### *Zobraz Histogram*

– vykreslí histogram zadaného snímku z náměru

### *Export Excel Histogram*

– export histogramu do formátu \*.csv – lze sním pracovat v programu Excel



*Generuj graf z řádku průměrovaný*

– vykreslí graf jasových hodnot zadaného řádku a snímku v náměru

*Export Excel Graf z řádku*

– export grafu jasových hodnot řádku

*Generuj přechod hrany*

– vygeneruje graf přechodu hrany ze všech náměrů pro zadanou pozici pixelu

*Export Excel přechod hrany*

– export grafu přechodu hrany do Excelu

## **Lineární interpolace**

*Spočti interpolaci z IMG*

– vypočítá a zobrazí hodnotu lineární interpolace ze zadaného řádku a snímku nastaveného v sekci Měření hrany.

*Generuj interpolační graf*

– vygeneruje a zobrazí průběh interpolace ze všech snímku v databázovém souboru. Řádek nastaven příslušným vstupem.

*Export Excel interpolace*

– export interpolačního grafu do formátu Excel

*Export Excel pozice pixelu*

– vygeneruje a exportuje do excelu hodnoty pozic pixelů

*Export Excel Interpolace dvou hran*

– generuje a exportuje interpolaci dvou rovnoběžných hran v měření. Je nutné nastavit kde začíná první hrana a kde druhá

## **Photometrická interpolace**

- zde je ovládání obdobné jako u lineární interpolace až na podrobnější zadání parametrů pro výpočet

Uh – horní hranice přechodu hrany (hodnota jasu)

Ul – dolní hranice přechodu hrany (hodnota jasu)

XB – začátek přechodu hrany (pixel)

XE – konec přechodu hrany (pixel)

## **Těžiště**

*Komparační úroveň*

– nastavení komparační úrovně pro vyhodnocení těžiště – velmi citlivé na

výslednou hodnotu

*Velikost pixelu*

– velikost pixelu senzoru v mikrometrech, použita při přepočtu hodnot

*Spočítej vzdálenost dvou těžišť*

– spočítá vzdálenost dvou těžišť zadaných parametry Náměr1 a Náměr2 – hodnoty jednotlivých snímků jsou průměrovány

*Generuj graf posunu těžiště*

– vygeneruje a zobrazí graf posunu těžiště v závislosti na skutečném posunu pojezdu

*Export Excel posun těžiště*

– exportuje graf posunu těžiště do formátu Excel

## 2.4 Formáty ukládaných dat

První verze náměrů byly ukládány do barevné 24b bitmapy. Název souboru se skládal z čísla náměru, polohy pojezdu a čísla snímku.

Příklad uložení dat v souborovém formátu :

1-APB=0\_100\_0.bmp

1-APB=0\_100\_1.bmp

1-APB=0\_100\_2.bmp

2-APB=0\_200\_0.bmp

2-APB=0\_200\_1.bmp

2-APB=0\_200\_2.bmp

Jedná se o 2 náměry po 3 snímcích s posuvem 100 mikrometrů.

Druhá verze ukládání náměrů do souborového formátu ukládá již v osmibitovém indexovaném formátu bitmapy (gray scale). Název je složen z čísla náměru, čísla snímku a polohy pojezdu v „unit“. Tento formát nebyl mnohokrát použit, jelikož bylo směřováno k užívání databázového formátu ukládání dat.

Příklad uložení dat v souborovém formátu 8b indexed formát :

1\_0-0\_0.bmp

1\_1-0\_0.bmp

2\_0-1000\_0.bmp

2\_1-1000\_0.bmp

Názvy těchto souborů nám sdělují, že se jedná o dva náměry po dvou snímcích s posunem o 1000 jednotek na pojezdu.

Třetím typ ukládaného formátu dat nazývám tzv. databázovým formátem dat. Jedná se o jeden velký soubor (kontejner), ve kterém jsou uloženy veškeré datové informace. Jeho struktura je velmi jednoduchá. Před každým snímkem jsou vloženy 4 integery libovolných informací o velikosti 32 bitů. Poté následují jednotlivé hodnoty pixelů v 16 bitovém formátu. 16b formát byl implementován z důvodu budoucího měření se senzory větší bitové hloubky. Smluvní dohoda o konvenci a dodržování standardu byla následující.

První Int32 – číslo náměru ( frame )

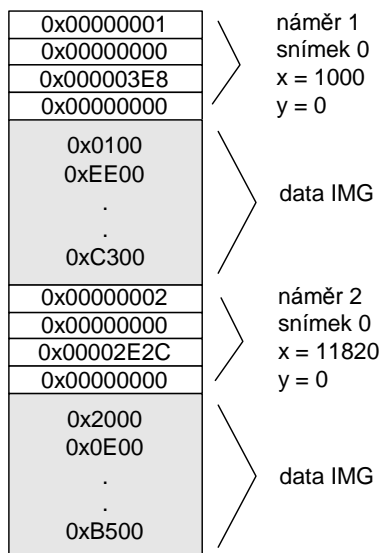
Druhý Int32 – číslo snímku ( subframe )

Třetí Int32 – pozice pojezdu v ose X

Čtvrtý Int32 – pozice pojezdu v ose Y (zůstává prázdný v případě jednoho pojezdu)

Následuje blok 16b. dat o velikosti = počet řádků \* počet sloupců (první bajt je roven dolní polovině „slova“ a druhý bajt je roven horní polovině „slova“).

Strukturu tohoto formátu pro názorné pochopení zobrazuje obrázek 2.13.



Obr. 2.13 Struktura databázového formátu ukládání dat

### 8b indexovaný formát

Tento typ formátu se získá převedením obrázku do formátu škály šedi (gray scale) spolu s vytvořením tzv. Color palety, která v našem případě měla hodnotu jasu 0 – 255. Pro výpočet a převod hodnoty jednotlivých pixelů u barevného obrázku bylo použito procentuálního zastoupení jednotlivých složek barev RGB.

$$grayScale = 0.30 * R + 0.59 * G + 0.11 * B \quad (2.2)$$

## 2.5 Zásady pro pořízení a vyhodnocení náměrů

Pro pořízení náměrů a jejich správné vyhodnocení je nutné dodržet určité zásady sepsané níže, které vyplynuly ze zkušenosti s používáním.

1, Než je spuštěn proces měření – zkontrolujte zda máte zapnut pojezd nebo zda jste vhodně nastavili výřez obrazu

2, Zkontrolujte zda v adresáři pro uložení je dostatek místa. V případě ukládání celých snímků může velikost narůst až do řádu GB dat.

3, Při nastavování clony objektivu si aktivujte měření těžiště – zde se vám bude zobrazovat max. hodnota jasu v obraze. Vhodným nastavením nedojde k přesaturování obrazu.

4, Před spuštěním procesu měření vynulujte měřidlo pojezdu.

5, Inkrement pojezdu používejte v kladných hodnotách. Důvodem je kompatibilita s programem pro vyhodnocení měření. V případě, že budete vyhodnocovat snímky vlastním softwarem je volba na vás.

**POZN.** – Software pro zpracování naměřených dat je omezen na vyhodnocení přechodu hrany a to z tmavé do světlé (z černé do bílé). V případě, že by přechod hrany byl obrácený je nutná úprava zdrojových kódů.

### 3 ALGORITMY MĚŘENÍ A VYHODNOCENÍ DAT

Před popisem jednotlivých algoritmů nebo zdrojových kódů je nutné uvést vhodnou terminologii, kterou používám v textu.

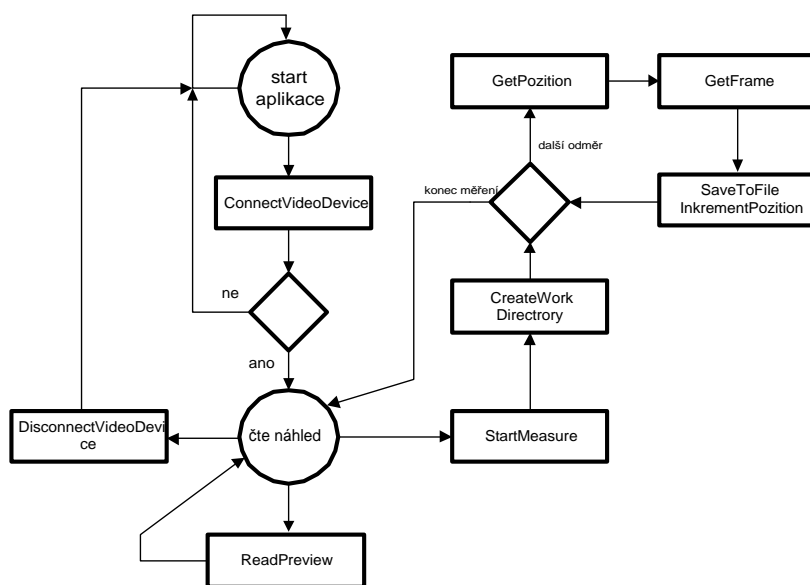
Snímek – jedná se o jeden obrázek, který lze zobrazit v jakémkoliv grafickém programu

Náměr – jedná se o jeden cyklus měření od načtení pozice až po její inkrementaci. Může obsahovat jeden nebo více snímků, které se pak mohou průměrovat.

#### 3.1 Software pro náměr dat

Základním sekvence běhu aplikace pro náměr dat ukazuje diagram na obrázek 3.1. Po spuštění aplikace je nutné vybrat vstupní zařízení, ze kterého se mají načítat data. Při správném připojení se spustí vlákno, které každých 100ms načítá obrazová data z tohoto zařízení a poté tyto vykreslí v náhledu zobrazení.

V případě, že máme připravené a nastavené všechny parametry pro měření, stisknutím tlačítka start dojde k deaktivaci vlákna a spustí se samotný proces měření. Ten se skládá ze tří fází: načti pozici, načti obrazová data, ulož obrazová data a inkrement pozice.



Obr. 3.1 Sekvenční diagram – aplikace pro náměr dat

Celý proces od načtení pozice až po její uložení a inkrementaci trvá přibližně 2-2,5s. Takto dlouhá doba je způsobena vyčítáním hodnoty z pojezdu. Pojezd sám o sobě neumí nahlásit, že doreguloval na správnou polohu. Je proto nutné cyklicky vyčítat data o pozici. Když se hodnota 20x nezmění, můžeme prohlásit, že pojezd je na správné

pozici a pokračovat v měřícím cyklu. Tuto funkci za nás obstarává software pro řízení pojezdu.

Pořízení některých náměrů je velmi časově náročné. Chceme-li pořídit 1000 náměrů, kde každý bude mít 50 snímků, lze si lehce dopočítat dobu měření.

(50snímků načteme cca za 30s)

$$1000 * 2,5 + 1000 * 30 \cong 9hodin$$

V rámci optimalizace a hlavně zkrácení doby měření, byla použita pointerová aritmetika v C#. Níže uvádím algoritmus převodu 24b Bitmapy do 8bitového indexovaného formátu.

```
private byte[] GetByteArrayFromBmp_speed(Bitmap original)
{
    byte[] buffer = new byte[original.Width * original.Height];
    BitmapData originalData = original.LockBits(new Rectangle(0, 0,
        original.Width, original.Height),

        ImageLockMode.ReadOnly, PixelFormat.Format24bppRgb);
    unsafe
    {
        uint n = 0;
        //set the number of bytes per pixel
        int pixelSize = 3;
        for (int y = 0; y < original.Height; y++)
        {
            //get the data from the original image
            byte* oRow =
                (byte*)originalData.Scan0 +
                (y*originalData.Stride);
            for (int x = 0; x < original.Width; x++)
            {
                //create the grayscale version
                byte grayScale =
                    (byte)((oRow[x * pixelSize] * .11) + //B
                    (oRow[x * pixelSize + 1] * .59) + //G
                    (oRow[x * pixelSize + 2] * .3)); //R

                buffer[n++] = grayScale;
            }
        }
        original.UnlockBits(originalData);
    }
    return buffer;
}
```

Bez použití pointerů trval převod Bitmapy o rozměru 720x576 cca 900ms. Při použití pointerů se doba zkrátila na 100ms.

## 3.2 Software pro zpracování dat

Software pro zpracování dat je nutný k získání informací o snímcích uložených v databázovém souboru. Má v sobě naimplementované funkce, které umí vrátit jeden konkrétní snímek, nebo zpracovat celou sadu náměrů dle konkrétního požadavku. Většinu výsledků umí exportovat do formátu \*.csv.

### 3.2.1 Formát CSV

Jedná se o jednoduchý souborový formát určený pro výměnu tabulkových dat. Tento formát není standardizován žádnou normou, ale umí s ním pracovat tabulkové programy typu Excel, Matlab a OpenOffice.

Formát zápisu se skládá z řádků, kde jednotlivé hodnoty jsou odděleny středníkem.

Pro ukázkou zápisu:

0;70,82

10;70,826

20;70,839

30;70,85

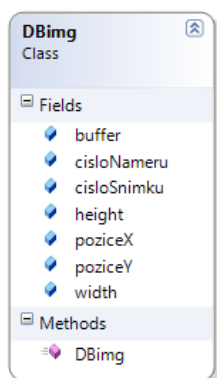
Tato data se v Matlabu nebo Excelu automaticky rozparsují viz. obrázek 3.2. a pak můžeme z hodnotami libovolně pracovat jako v klasickém tabulkovém programu.

	A	B	C
1	0	70,82	
2	10	70,826	
3	20	70,839	
4	30	70,85	
5			

Obr. 3.2 Screen Excel – rozparsování dat CSV formátu

### 3.2.2 Třída DBimg

Třída *DBimg* byla vymyšlena z důvodů zjednodušení operací nad zdrojovými daty. Po instanci reprezentuje objekt jednoho snímku, se kterým lze provádět různé operace popsané níže. Struktura této třídy je zobrazena na obrázku 3.3.



Obr. 3.3 Třída DBimg

Reprezentuje v sobě buffer s vlastními daty obrázku (16bitová data), číslo náměru, ke kterému patří, číslo snímku (říká kolikátý je to snímek pořízený v jednom náměru), šířku, výšku a pozici pojezdu X a Y. Pokud máme pojezdy zapojené do kříže, X je

pohyb ve směru řádků kamery, Y pohyb ve směru sloupců kamery.

Funkce, která vrátí objekt *DBimg* se jmenuje „*ReadImgFromDB(...)*“

```
private DBimg ReadImgFromDB(String urlFile, int cisloNameru,
                             int cisloSnimku, int pocetSnimkuVnameru,
                             int pocetInfoInt32, int sirka, int vyska)
{
    int bufferSize = sirka * vyska * 2;
    int dataInfoSize = pocetInfoInt32 * 4;
    int dbImgSize = bufferSize + dataInfoSize;

    byte[] bufferDBimg = new byte[dbImgSize];
    byte[] bufferInfoInt32 = new byte[dataInfoSize];
    byte[] buffer = new byte[bufferSize];

    long offset = (long)cisloNameru * (long)pocetSnimkuVnameru *
                  (long)dbImgSize + (long)cisloSnimku *
                  (long)dbImgSize;
    DBimg imgDB = null;
    using (FileStream fsSource = new FileStream(urlFile,
                                                FileMode.Open,
                                                FileAccess.Read))
    {
        // read DATA
        fsSource.Seek(offset, SeekOrigin.Begin);
        fsSource.Read(bufferDBimg, 0, bufferDBimg.Length);
        fsSource.Close();
    }

    //zkopiruju data info
    for (int i = 0; i < dataInfoSize; i++)
    {
        bufferInfoInt32[i] = bufferDBimg[i];
    }
    for (int i = 0; i < bufferSize; i++)
    {
        buffer[i] = bufferDBimg[i + dataInfoSize];
    }

    imgDB = new DBimg(BitConverter.ToInt32(bufferInfoInt32, 0),
                     BitConverter.ToInt32(bufferInfoInt32, 4),
                     BitConverter.ToInt32(bufferInfoInt32, 8),
                     BitConverter.ToInt32(bufferInfoInt32, 12),
                     sirka, vyska, buffer);

    return imgDB;
}
```

Tato funkce si nejprve spočítá přesné umístění snímku ve fyzickém souboru. Poté otevře soubor a přenesení ukazatel na požadované místo čtení ze souboru. Načte zadaný počet informačních Int32 a poté zbylá obrazová data do bufferu. Po dokončení cyklu čtení vytvoří objekt *DBimg* reprezentující načtený snímek.



Obdobnou funkci pro vyčítání dat z databázového souboru (v Matlabu), uvádím níže.

```
pocetNameru = 1;
pocetSnimku = 50;
nameFile = '_dataBox.dat';
sloupce = 1280;
radky = 1024;

%obrazek
IMG =uint8(zeros(radky,sloupce));
prumIMG = zeros(radky,sloupce);

fid=fopen(nameFile,'rb');

for i = 1:pocetNameru
    for j = 1:pocetSnimku
        %nacte info data
        count = 4;
        INFO = fread(fid, count, 'int32');

        %nacte radek IMG
        count = sloupce;
        for k = 1:radky
            radek =uint8((fread(fid, count, 'uint16'))');
            IMG(k,:) = radek;
        end

        if pocetSnimku == 1
            prumIMG = double(IMG);
        else
            prumIMG = prumIMG + double(IMG);
        end
    end
    if not (pocetSnimku == 1)
        prumIMG = prumIMG / pocetSnimku;
    end

    %ZDE FUNKCE nebo OPERACE nad průměrovanými snímky
end
fclose(fid);
```

Jednou z důležitých funkcí pracujících se strukturou DBimg se jmenuje „GetBrightFromDBimg(...)“. Tato funkce vrátí jasovou hodnotu námi určeného pixelu. Její zdrojový kód je následující :

```
private double GetBrightFromDBimg(DBimg dImg, int row, int column)
{
    double jas = 0;
    byte[] buffer = dImg.buffer;
    int lbuffer = 0;

    //radky
    for (int i = 0; i < dImg.height; i++)
    {
        //sloupce
        for (int j = 0; j < dImg.width; j++)
        {
            if (i == row && j == column)
            {
                jas = (double)buffer[lbuffer];
                return jas;
            }
            lbuffer++;
            lbuffer++;
        }
    }
    return jas;
}
```

Tato funkce prohledává buffer celého snímku po sloupcích a řádcích. V případě, že

narazí na pixel, který hledáme, převede a vrátí nám jeho jasovou hodnotu.

### 3.2.3 Algoritmus vykreslení přechodu hrany

Funkce pro vykreslení přechodu hrany se nazývá „*Thread\_GenerCrossEdge()*“. Její zkrácený výpis kódu je následující

```
double pozition = 0;
double bright = 0;
DBimg dImg = null;
PointPairList tabDataIMG = new PointPairList();

//projde vsechny namery
for (int i = 0; i < pocetNameru; i++)
{
    //projde vsechny snimky
    for (int j = 0; j < pocetSnimku; j++)
    {
        dImg = ReadImgFromDB(...);
        bright += GetBrightFromDBimg(...);
        pozition += dImg.poziceX;
    }
    bright = bright / pocetSnimku;
    pozition = pozition / pocetSnimku;

    tabDataIMG.Add(pozition, bright);

    bright = 0;
    pozition = 0;
}
```

Funkce prochází všechny jednotlivé snímky a náměry, které jsou obsažené v databázovém souboru.

Po načtení snímku funkcí *ReadImgFromDB(...)* se získá hodnota jasu zvoleného pixelu funkcí *GetBrightFromDBimg(...)*. Hodnota jasu se akumuluje a poté průměruje ku počtu všech snímků v daném náměru. Takto vypočtený průměr je uložen spolu s pozicí do tabulky pro pozdější vykreslení grafu.

### 3.2.4 Algoritmus výpočtu lineární interpolace hrany

Funkce pro vykreslení přechodu hrany se nazývá „*Thread\_GenerLinInterpolaceGraf()*“. Její zkrácený výpis kódu je následující

```
double pozition = 0;
double interpolace = 0;
DBimg dImg = null;
PointPairList tabDataIMG = new PointPairList();
PointPairList pomocTab = new PointPairList();

//projde vsechny namery
for (int i = 0; i < pocetNameru; i++)
{
    //projde vsechny snimky
    for (int j = 0; j < pocetSnimku; j++)
    {
        pomocTab.Clear();
        dImg = ReadImgFromDB(...);
        //nacte cely radek snimku
        pomocTab = GetListPointCrossFromIMG(...);
        interpolace += SpoctiLinInterpolaci(pomocTab);
        pozition += dImg.poziceX;
    }
    interpolace = interpolace / pocetSnimku;
    pozition = pozition / pocetSnimku;
}
```

```

tabDataIMG.Add(pozition, interpolace);
interpolace = 0;
pozition = 0;
}

```

Po načtení snímku ze souboru *ReadImgFromDB(...)* se vyparsuje z této struktury funkcí *GetListPointCrossFromIMG(...)* tabulka hodnot přechodu hrany jednoho řádku. Z tohoto řádku se spočítá lineární interpolace funkcí *SpoctiLinInterpolaci(...)*. Tyto hodnoty interpolace se akumulují a průměrují ze všech snímků v náměru.

Funkce *SpoctiLinInterpolaci(...)* počítá interpolaci dle vztahu (1.9). Zde je přepis do programovacího jazyka C#.

```
X_hrany = XnYn.X + ((Ukompr - XnYn.Y) / (Xn1Yn1.Y - XnYn.Y));
```

Pro výpočet komparační úrovně je použito vztahu (1.7), kde hodnota  $U_h$  je maximální hodnota jasu v daném řádku a  $U_l$  má nejmenší jasovou hodnotu v daném řádku.

$x_n y_n$  – hodnota prvního nalezeného pixelu pod komparační úrovní

$x_{n1} y_{n1}$  – hodnota prvního nalezeného pixelu nad komparační úrovní

Hodnoty všech parametrů se generují pro každý výpočet znovu.

### 3.2.5 Algoritmus výpočtu fotometrické interpolace hrany

Funkce pro vykreslení přechodu hrany se nazývá *Thread\_GenerPhotInterpolaceGraf()*. Její kód je naprosto shodný s kódem v 3.2.4 až na záměnu funkce „*SpoctiLinInterpolaci()*“ za „*SpoctiPhotInterpolaci()*“.

Funkce *SpoctiPhotInterpolaci(...)* počítá interpolaci dle vztahu (1.12), kde přepis do programovacího jazyka C#

```

for (int i = XB; i < XE+1; i++)
{
    dX = 1;
    yN = data[i].Y;
    sumYdX += dX * (yN - U1);
}
X_hrany = XE - (sumYdX / (Uh - U1));

```

Všechny vstupní podmínky jako  $U_h$  (v (1.12) rovno  $y_1$ ),  $U_l$  (v (1.12) rovno  $y_0$ ),  $X_B$  (v (1.12) rovno  $x_0$ ) a  $X_E$  (v (1.12) rovno  $x_1$ ) jsou zadávány před spuštěním výpočtu obsluhou.

### 3.2.6 Algoritmus výpočtu souřadnic těžiště

Funkce pro nalezení těžiště se nazývá „*GetTeziste(...)*“. Výpis kódu je následující.

```
private Hashtable GetTeziste(byte[] buffer, int pRadku,
                             int pSloupcu, int komparUrov)
{
    Hashtable souradnice = new Hashtable();
    double m = 0;
    double mixi = 0;
    double miyi = 0;
    int poc = 0;

    for (int radek = 0; radek < pRadku; radek++)
    {
        for (int sloupec = 0; sloupec < pSloupcu; sloupec++)
        {
            if (komparUrov < buffer[poc])
            {
                mixi += buffer[poc] * sloupec;
                miyi += buffer[poc] * radek;
                m += buffer[poc];
            }
            poc++;
        }
    }

    souradnice.Add('X', (double)mixi / m);
    souradnice.Add('Y', (double)miyi / m);

    return souradnice;
}
```

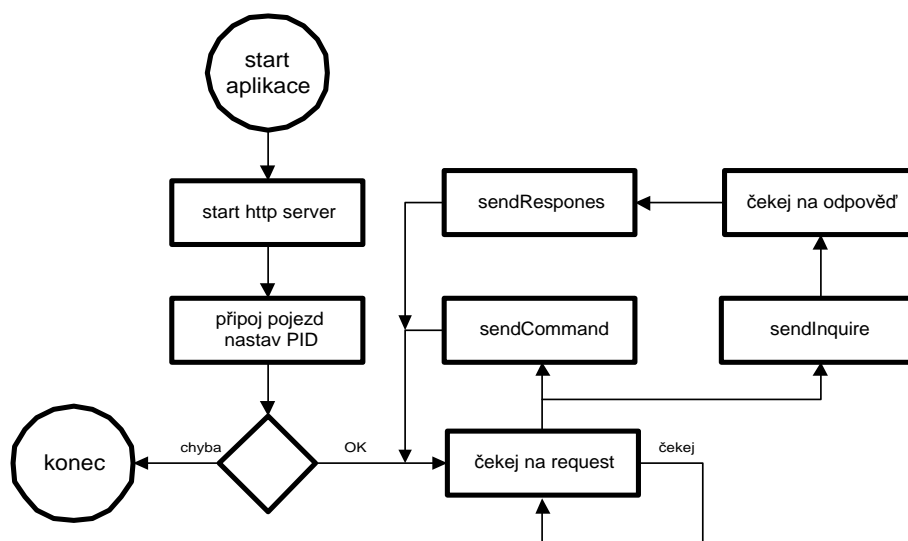
Tato funkce počítá těžiště dle vztahu (1.14) a to pouze pro světlé objekty nacházející se nad komparační úrovní, kterou nastavujeme před spuštěním skriptu.

Funkce, která generuje data pro vykreslení grafu závislosti změřené vzdálenosti na posunu pojezdu se nazývá „*Thread\_GenerGrafPosunuTeziste()*“.

Tato funkce nám vrací vzájemný rozdíl posunu dvou těžišť. První těžiště je staticky počítáno z náměru číslo 0, a druhé se postupně počítá dle náměru dalšího v pořadí. Touto funkcí dostaneme závislost posunu těžiště na skutečném posunu pojezdu. Těžiště je průměrováno z počtu snímků v náměru.

## 3.3 Software pro řízení pojezdu

Aplikace pro řízení pojezdu je modifikovanou komponentou web-serveru, která zajišťuje prostředníka a překladač příkazů mezi aplikací pro náměr dat a řídicí jednotkou pojezdu. Příkazy, bližší popis uveden v kapitole 2.3.2, můžeme rozdělit do dvou skupin a to příkazy, které nevyžadují zpětnou odezvu řídicí jednotky (command) a příkazy, které se dotazují na nějaký stav (inquire). Sekvenční diagram této konsolové aplikace je na obrázku 3.4.



Obr. 3.4 Sekvenční diagram – aplikace pro řízení pojezdu

Po startu aplikace je automaticky nastartován web server a provedeny dotazy směrem k pojezdům, které jsou nakonfigurovány v souboru config.ini. V případě, že ani jeden z nich neodpoví, aplikace se uvede do módu, kdy přijme request, ale vrací žadateli automatickou odpověď „FAIL“. V případě, že inicializace proběhla úspěšně, řídicím jednotkám(ce) je poslán příkaz pro výchozí nastavení PID regulátoru a motorů. Aplikace nyní čeká na příchozí žádost.

### 3.3.1 Řídicí jednotka MARSII

Připojení této řídicí jednotky a jejího ovládání nebyl žádný problém díky dobré dokumentaci, kde autor jednoznačně popisuje jak se jednotka chová a jakým způsobem komunikuje.

Tabulka 3.1 uvádí nastavení portu pro komunikaci. Jako jednu z nevýhod této jednotky bych chtěl vytknout posílání hodnot pozice pojezdu v milimetrech nikoliv v jednotkách rozlišitelnosti.

Tab. 3.1 Nastavení parametrů sériového portu

Rozhraní	Sériový port
Rychlost	9600 kBaud/s
Parita	Žádná
Počet bitů	8
Počet stop bitů	dva
RTS signály	Zapnuty

Formát příkazů :

Příkazy jsou posílány v textové formě.

Tabulka 3.2 vypisuje nejčastěji používané příkazy. Symbol „x“ značí jaký motor je adresován (A,B,C).

Dosažitelná přesnost a přepočet jednotek na milimetry je následující

$$1 \text{ unit} = 1\mu\text{m}, 10 \text{ unit} = 10\mu\text{m}, 1000 \text{ unit} = 1\text{mm}$$

Tab. 3.2 Tabulka použitých příkazů

Příkaz	Popis
REGPx: <i>hodn</i>	Proporcionální složka reg.
REGIx: <i>hodn</i>	Integrační složka reg.
REGDx: <i>hodn</i>	Derivační složka reg.
HHx:	Hledání ref. poloh
GRx: <i>hodn</i>	Relativní posun
Gx: <i>hodn</i>	Absolutní posun
APx?	Vrátí pozici
RELEASEx:	Odpojení regul. mot. x
CLEARx:	Vypnutí řízení a nulování mot. x
STOPx:	Zastavení pohybu

Příklady použití příkazů :

```
//funkce pro poslani dat na seriový port
void SendCommand(String data)
{
    ...
    ...
}

//nastavení regulatoru P motoru A na hodnotu 100
SendCommand("REGPA:100");
//posun o relativní hodnotu motoru B o hodnotu 100,123mm
SendCommand("GRB:100.123");
//nactení hodnoty pozice motoru C
SendCommand("AGC?");
```

### 3.3.2 Řídicí jednotka MERCURY PI

S touto řídicí jednotkou i pojezdem samotným bylo nemálo problémů. Manuál neodpovídal skutečnosti, nehledě na funkčnost některých příkazů. Nejprve bylo nutné identifikovat jednotlivé příkazy a to způsobem odposlechu na portu pomocí speciálního softwaru [13]. Některé další parametry nastavení portu musely být zjištěny experimentálně.

K dispozici byly dva typy řídicích jednotek. Jednu je možné ovládat pouze pomocí sériového portu a druhou pomocí USB i sériového portu ovšem ne obojím zároveň. Tato blokáce byla výrobcem vymyšlena přímo HW uspořádáním konektorů.

Jednotka má možnost volby vlastní adresy díky přepínačům umístěným vně své krabičky a též i komunikační rychlosti. Starší model ovšem disponuje zavádějícím popisem na krabičce jednotky. Adresovat lze pouze prvními čtyřmi přepínači, nikoliv všemi jak naznačuje popis. Další dva slouží pro konfiguraci rychlosti a zbývající jsou

pro volné užití.

Tab. 3.3 Nastavení parametrů sériového portu

Rozhraní	Sériový port
Rychlost	9600 kBaud/s
Parita	Žádná
Počet bitů	8
Počet stop bitů	Jeden
RTS signály	Vypnuty

Formát příkazů :

Každý příkaz je nutné ukončit bytem 0x0D (hex). Bez tohoto ukončení jednotka zahlásí error. Jednou z výhod této řídicí jednotky je možnost přijímat více příkazů zároveň. Příkazy se musí ovšem oddělit znakem čárky (, , “).

Hodnoty nastavení pozice nebo při vyčtení se uvádějí v tzv. krokových jednotkách. Pro přepočítání na milimetry platí vztah

$$1 \text{ unit} = 100\text{nm}, 10 \text{ unit} = 1\mu\text{m}, 10000 \text{ unit} = 1\text{mm}$$

V tabulce 3.4 je výčet všech identifikovaných příkazů, které jsem byl schopen detekovat.

Tab. 3.4 Tabulka identifikovaných příkazů

Příkaz	Popis
BF	Brzda off
BN	Brzda on
DH	Nulování měřidla polohy
DD	Derivační složka
DI	Integrační složka
DP	Proporcionální složka
MF	Motor off
MN	Motor on
MR	Relativní posuv
MA	Absolutní posuv
SA	Set akcelerace
SV	Set rychlost
TP	Vrátí pozici
TB	Vrátí adresu
TT	Vrátí cíl
WS	Čekej a pak stop
FE	Hledání ref polohy

Inicializace pojezdu :

Před posláním jakéhokoli příkazu je nutné zaslat adresu jednotky, pro kterou jsou příkazy určeny. Určení adresy je následující, platí analogie až do zařízení s adresou 16.

zařízení 1 adr. 0 ( 0x01 0x30 )

zařízení 2 adr. 1 ( 0x01 0x31 )

....

Odlišnosti obou verzí řídicí jednotky (novější s rozhraním USB, starší pouze s rozhraním seriového portu):

Při zapnutí jednotky a pojezdu je pojezd automaticky zabrzděn. U starší jednotky je třeba tento pojezd odbrzdit a to příkazem „BF“. Novější jednotka toto dělá automaticky.

Příkaz MF a MN fungují pouze u starší jednotky, novější při zavolání tohoto příkazu zahlásí error.

Příklady užití příkazů :

```
byte endByte = 0x0D; //ukoncovaci znak prikazu
void SendCommand(String data); //posle textová data na seriový port
void SendCommand(Byte data); //posle byte dat na seriový port
void SendCommand(Byte[] data); //posle pole byte data na seriový port

//nastaveni adresy 0 zarizeni
Byte[] pole = new Byte() { 0x01, 0x30 }
SendCommand(pole);

// nastaveni parametru regulatoru
SendCommand("DP120,DI15,DD300");
SendCommand(endByte);

// nastaveni rychlosti a akcelerace pojezdu
SendCommand("SV56000,SA450000");
SendCommand(endByte)
// odbrzdeni, nalezeni referencni polohy (1s regulace), nulovani pozice
SendCommand("BF,FE1,WS1000",DH);
SendCommand(endByte);

// relativni posun motoru o 1 mm (nezapomenout odbrzdit)
SendCommand("MR10000");
SendCommand(endByte);

// absolutni posun motoru o 1 mm (nezapomenout odbrzdit)
SendCommand("MA10000");
SendCommand(endByte);
```



## 4 ZPRACOVÁNÍ NÁMĚRŮ A VÝSLEDKY MĚŘENÍ

Jak jsem již zmínil v předchozí kapitole, jedním z cílů této diplomové práce je pořízení reálných náměrů průběhů hrany a vyhodnocení dosažené přesnosti pomocí lineární a fotometrické interpolace, zjištění dosažitelné přesnosti při sledování pohybu světelného objektu a analýza zpětnovazebního řízení při vyhodnocování snímaného obrazu.

Také jsem se zmínil o neúspěchu při hledání obdobného měření v rozsáhlých odborných člancích a síti internet. Tato skutečnost by mohla být zapříčiněna tím, že většina odborných pracovišť nedisponuje potřebným vybavením pro realizaci těchto náměrů.

### 4.1 Určení hranice počtu vzorků pro průměrování

Při obecném měření platí, že čím více odměrů pořídíme, tím více jsme schopni potlačit vliv šumu na výslednou hodnotu měření. Matematicky vyjádřeno, hodnota šumu poklesne o druhou odmocninu z počtu pořízených snímků.

Ovšem průměrování má svá omezení. Především se jedná o situace, kdy je třeba rychle měřit. Tam jsme omezeni kapacitou pamětí RAM nebo není možné více náměrů pořídít. Toto měření nám určilo hrubý odhad, kolik odměrů je v určité míře přesnosti dostatečná.

#### Vstupní informace :

Typ senzoru (kamery) : IP kamera Bosch NWC-0455-10P

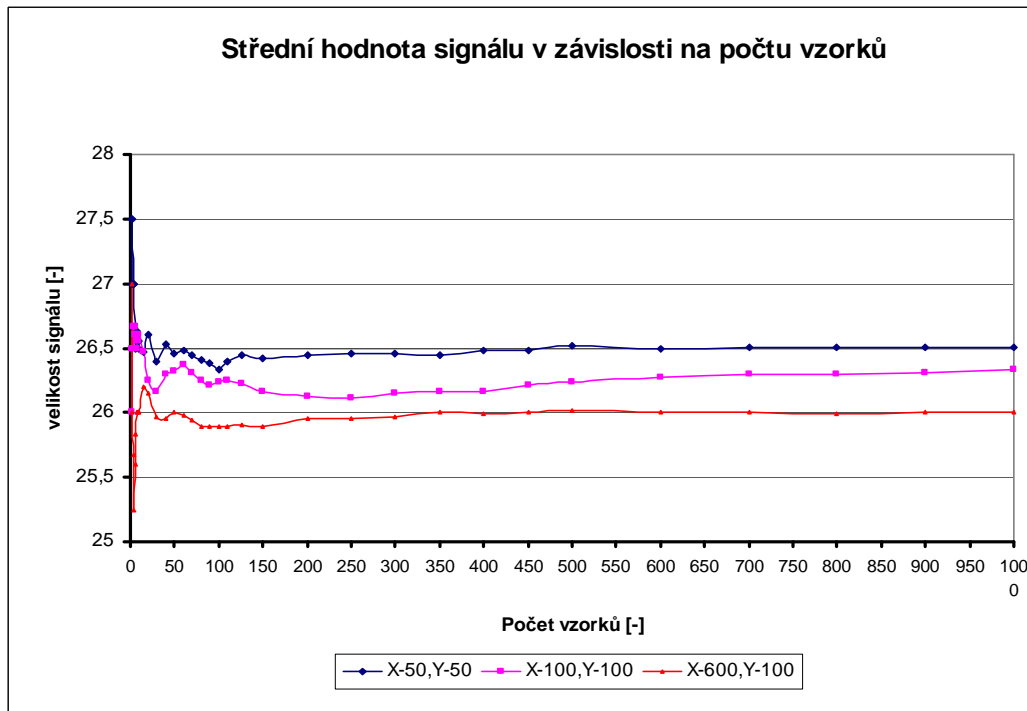
Počet snímků : 1000

Rozlišení kamery : 704x576

Vyhodnocení bylo provedeno v třech různých místech senzoru. Tabulka 4.1 a vypisuje několik měření a jejich střední hodnotu signálu.

	Souřadnice měřícího bodu [pixel]		
Souřadnice měřícího bodu [pixel]	X-50,Y-50	X-100,Y-100	X-600,Y-100
počet snímků v průměrování	střední hodnota signálu	střední hodnota signálu	střední hodnota signálu
1	27	26	27
5	26,60	26,6	25,6
10	26,5	26,6	26
50	26,46	26,32	26
100	26,34	26,24	25,89
300	26,46	26,157	25,973
500	26,514	26,24	26,022
1000	26,507	26,331	26,009

Tab. 4.1 Střední hodnoty signálu pixelu v závislosti na počtu snímků v průměrování



Obr. 4.1 Graf střední hodnoty signálu v závislosti na počtu vzorků

Každá hodnota v grafu je počítána dle vztahu

$$velikosti\_signálu = \frac{\sum_{n=0}^{n=pocetVzorku} I [n]}{pocetVzorku}$$

kde  $I$  je hodnota signálu pro daný vzorek  $n$ .

Pohledem na graf, obrázek 4.1, můžeme říci, že optimální počet snímků pořízených senzorem se pohybuje kolem hranice 300 a více snímků. Bohužel tato hodnota je příliš vysoká a časově by trvalo velmi dlouho, než by se náměry odměřily a vyhodnotily. Proto v rámci našeho měření lze požadovat za přiměřené průměrovat 25-50 snímků, kde hodnota šumu poklesne cca 5 – 7 krát.

## 4.2 Měření šumu za tmy a jeho vliv na měření

Šum za tmy sám o sobě nelze odstranit. Úkolem následujícího měření bylo určit do jaké míry ovlivňuje měření. Tento šum je způsobený resetovacím proudem jednotlivých pixelů senzoru a v mnohých případech se mu též říká KTC šum. Tento šum do měření vnáší určitou aditivní chybu a pokud máme jeho vzorek, lze ho z měřeného signálu úplně odstranit. Pokud nemáme jeho vzorek, můžeme jej potlačit průměrováním vstupního signálu.

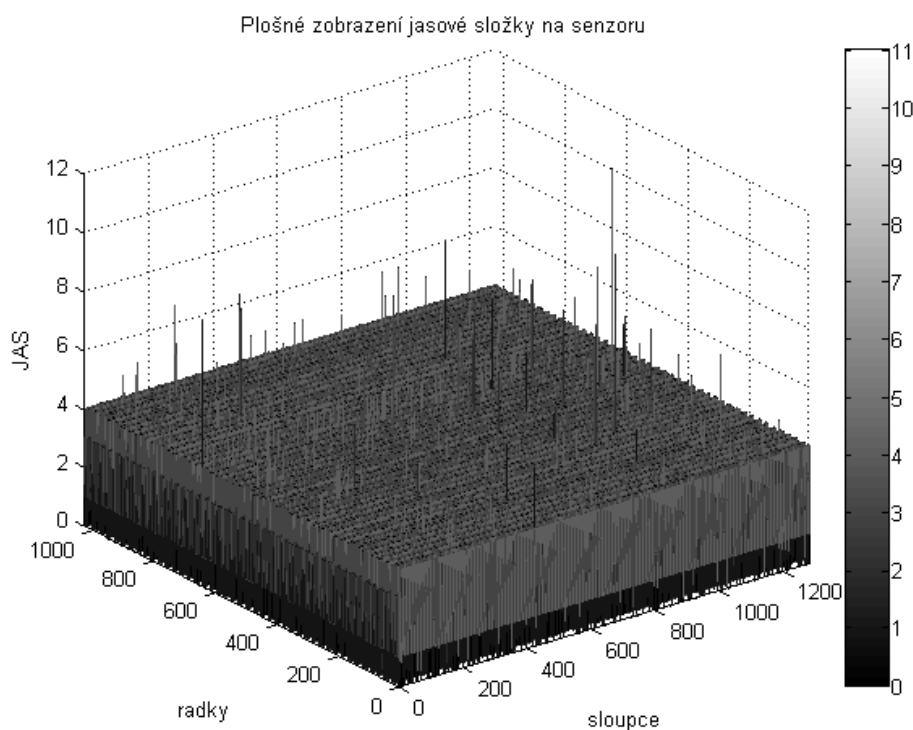
### Vstupní informace :

Typ senzoru (kamery) : Visor / CMOS MT9M001

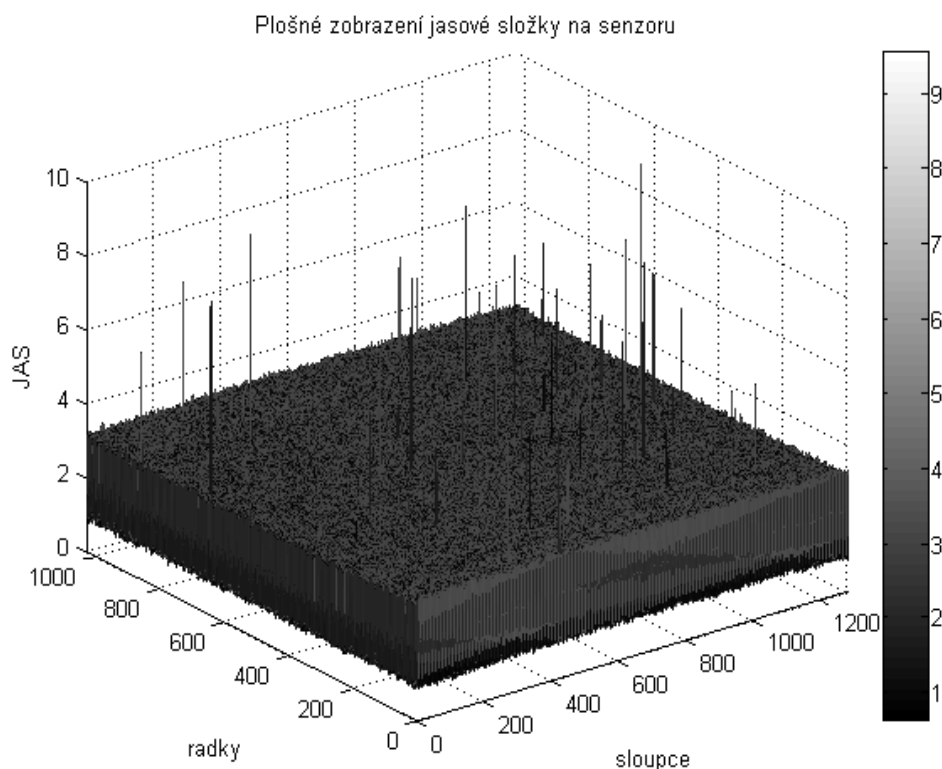
Počet snímků : 200

Rozlišení senzoru : 1280x1024

Na obrázku 4.2 je zobrazen prostorový graf jasového rozložení šumu senzoru při úplné tmě bez průměrování. Hodnota velikosti dosahuje hodnot až 4 jasových jednotek. Obrázek 4.3 ukazuje prostorový graf šumu senzoru s průměrováním 50 snímků. Hodnota šumu průměrováním poklesla zhruba na velikosti jednoho kvanta rozsahu. (1 jasová jednotka )

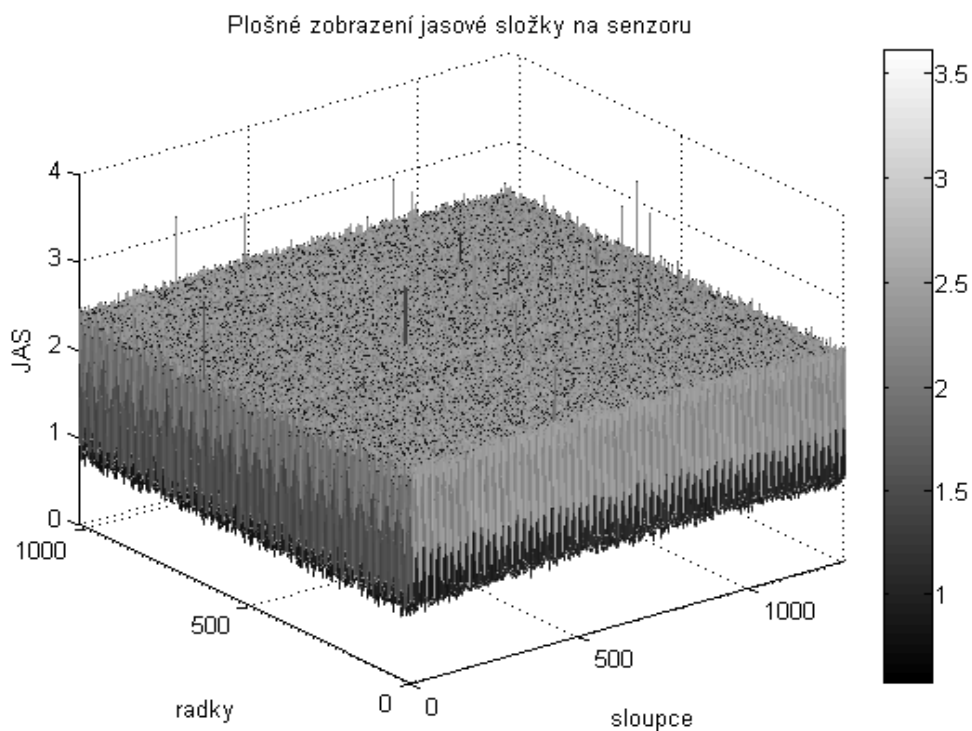


Obr. 4.2 Graf plošného jasového rozložení signálu na senzoru CMOS 1,2MPix – bez průměrování. Závěrka 0x419hex (defaultní hodnota)

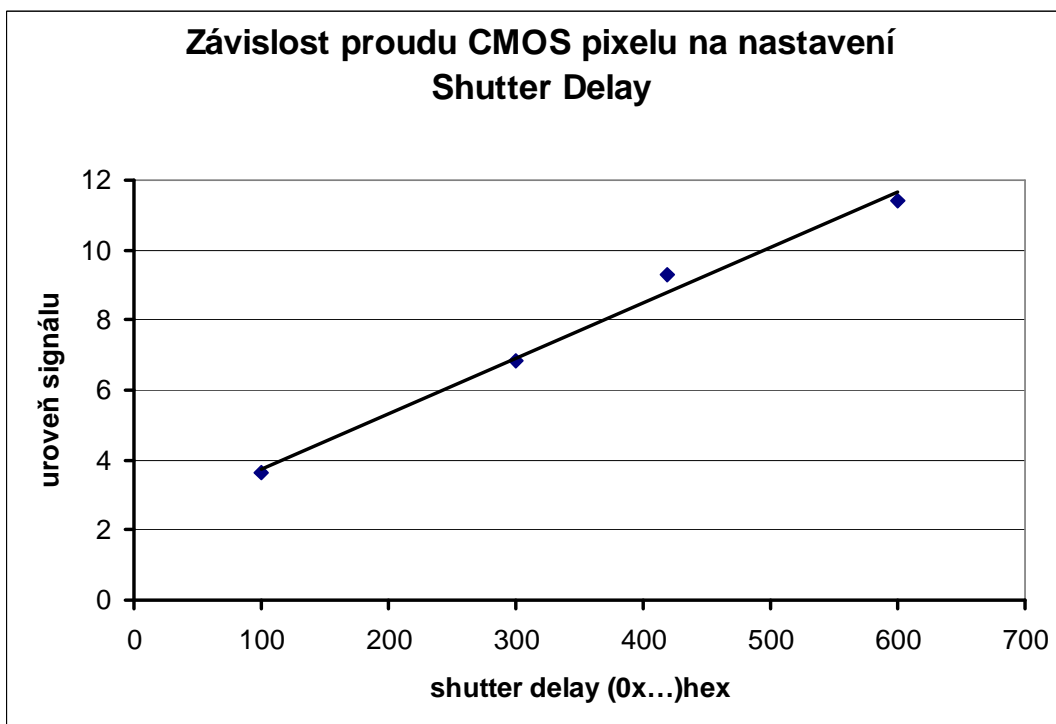


Obr. 4.3 Graf plošného jasové rozložení signálu na senzoru CMOS 1,2MPix – s průměrováním 50 snímků. Závěrka 0x419hex (defaultní hodnota)

Za povšimnutí určitě stojí vystřelující peaky některých pixelů. Ty jsou způsobeny pixely, které byly při výrobě špatně vyrobeny nebo jinak poškozeny. Abychom mohli s jistotou tvrdit, že se opravdu jedná o poškozené pixely, uskutečnili jsme pokus se změnou času elektronické závěrky na čipu senzoru. V případě že se jedná o špatné pixely bude se s prodlužováním času měnit délka peaku. Obrázek 4.4 ukazuje graf rozložení šumu při nastavení závěrky (shutter delay) na hodnotu 0x100hex. Jednotlivé peaky se zmenšily, čímž lze říci že se skutečně jedná o vadu výroby a pixelem senzoru protéká větší proud než u normálních. Závislost velikosti tohoto proudu na nastavení závěrky zobrazuje obrázek grafu 4.5 .



Obr. 4.4 Graf plošného jasové rozložení signálu na senzoru CMOS 1,2MPix – s průměrováním 50 snímků. Závěrka 0x100hex



Obr. 4.5 Graf závislosti proudu poškozeného CMOS pixelu na nastavení elektronické závěrky (shutter delay)

### 4.3 Měření přechodu ostré hrany

Při měření jsem z počátku používal průmyslovou kameru s CCD snímačem a převodníkem kompozitního signálu na digitální. Bohužel jsem si díky přepočtu řádků nemohl být jistý správností naměřených dat a mým zájmem bylo odměřit tyto náměry co nejpřesněji. Volba tedy dospěla k použití smart kamery Visor. U ní jsme byl schopen vyčíst data přímo ze senzoru bez jakýchkoliv úprav.

Pro posun hrany byl nainstalován pojezd značky ESSA s řídicí jednotkou MARSII, který měl maximální dosažitelnou rozlišitelnost 1 $\mu$ m. Tato rozlišitelnost ovšem byla velmi relativní. Při takto jemném posuvu často docházelo k podregulování hodnoty nebo nebylo možné ustálit regulátor. Bylo tedy nutné volit větší krok posunu a to 10 $\mu$ m. Tento krok ovšem nebyl jak se později ukázalo dostačující, pro stanovení chyby měření.

Vykreslení hrany ukazuje obrázek 4.6 a na obrázek 4.7 je graf chyby linearit v lineární části přechodu.

#### Vstupní informace :

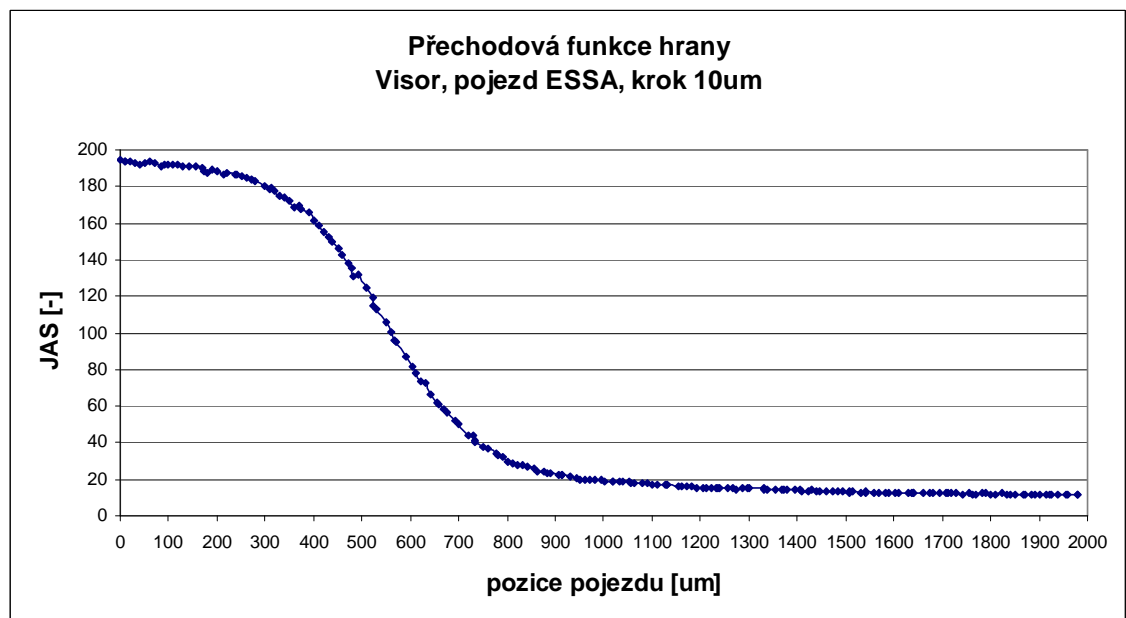
Senzor : Visor 1,2Mpx / CMOS MT9M001

Krok pojezdu: 10 $\mu$ m

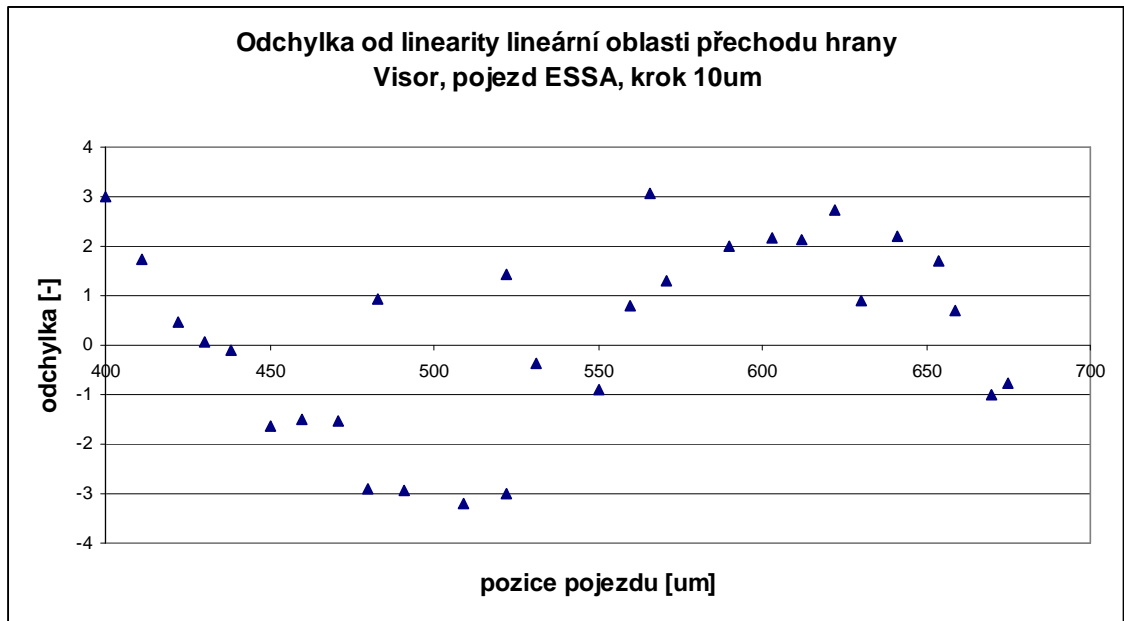
Zvětšení objektivu : 0,0291

Velikost pixelu: 5,2 $\mu$ m

Backlight : trubicový, v místnosti rozsvíceno zářivky



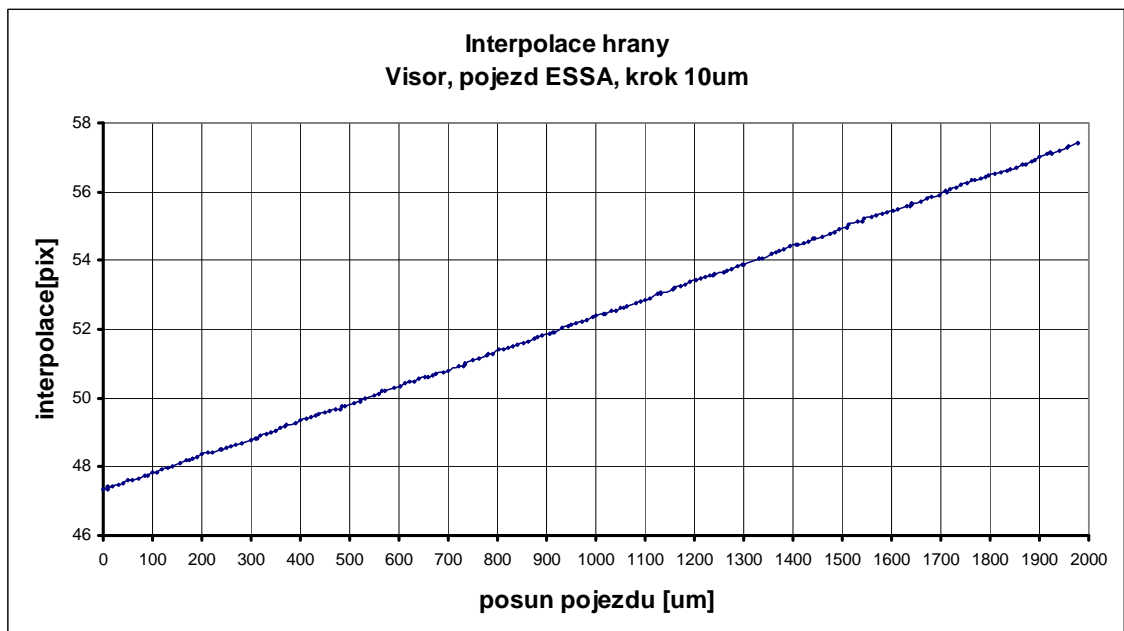
Obr. 4.6 Graf přechodu hrany – kamera Visor, pojezd ESSA, krok 10 $\mu$ m, průměrováno z 9 snímků



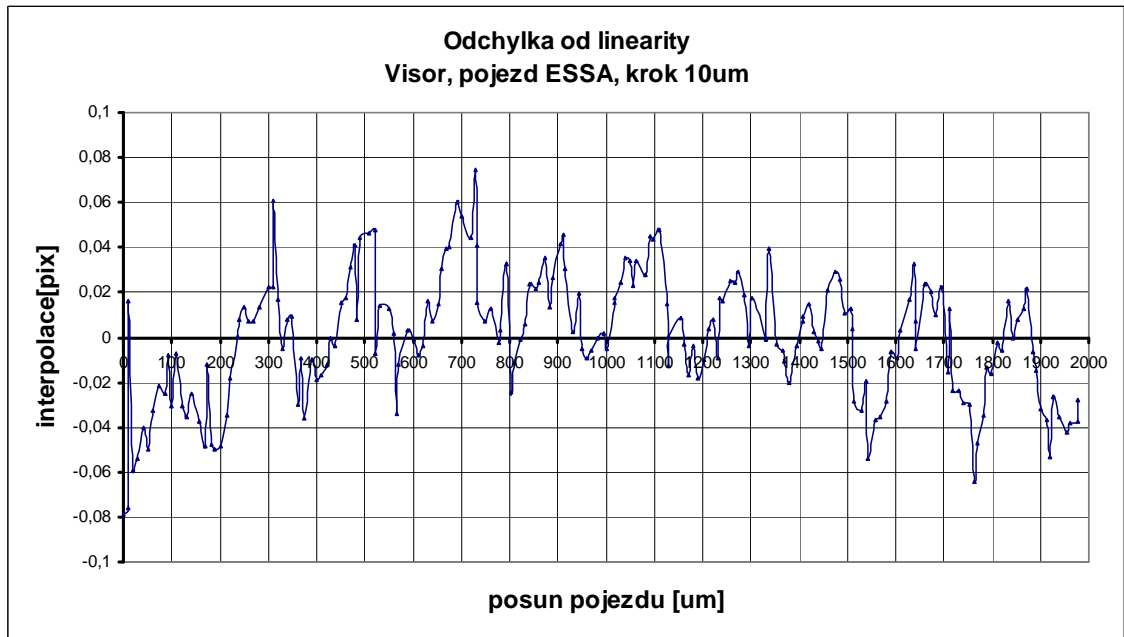
Obr. 4.7 Graf odchyly od linearity lineární oblasti přechodu z grafu 4.6

Dle grafu na obrázku 4.7 byla absolutní hodnota maximální odchyly od linearity 3,17. Takto vysoká hodnota je pravděpodobně způsobená fotonovým šumem na senzoru, kde hodnoty byly průměrovány z 9 snímků.

Výsledky aplikace algoritmu lineární interpolace můžete vidět na obrázku 4.8.



Obr. 4.8 Graf lineární interpolace hrany, kamera Visor, pojezd ESSA, krok 10um



Obr. 4.9 Graf odchyly od linearity interpolačního grafu na obrázku 4.8

Na obrázku 4.9 je zobrazen graf odchyly od linearity interpolace. Očekávaný průběh ale nebyl dost přesný a nijak nevytvídal o tvaru pixelu, který jsem očekával. V grafu se sice objevily jisté periodicity, bohužel nebylo možné určit, kde začíná a končí hranice pixelu.

Pro ještě jemnější regulaci posunu byl tedy nainstalován pojezd od firmy Physikinstrumente s řídicí jednotkou Merkurs PI. Tento pojezd se vyznačuje maximální rozlišovací schopností pojezdu 100nm. Pro další náměry jsem zvolil krok 1um, což se mi zdálo při úvaze zvětšení objektivu plně dostačující a provedli jsem další měření přechodu hrany.

**Vstupní informace :**

Senzor : Visor 1,2Mpx / CMOS MT9M001

Krok pojezdu: 1um

Zvětšení objektivu : 0,0291

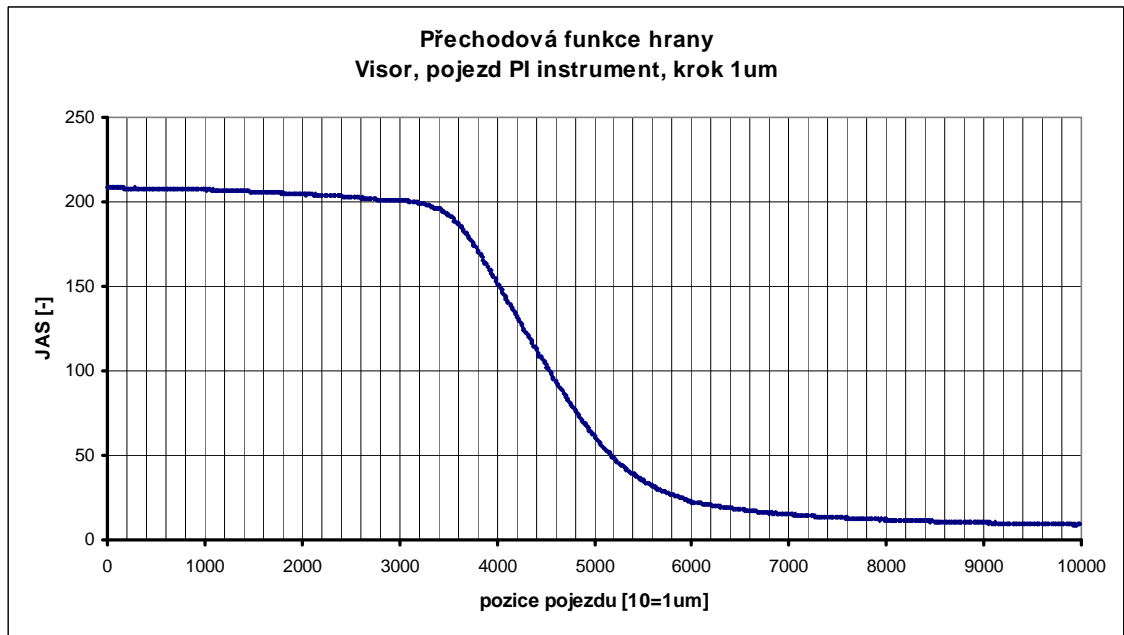
Velikost pixelu: 5,2um

Backlight : LED, v místnosti zhasnuto

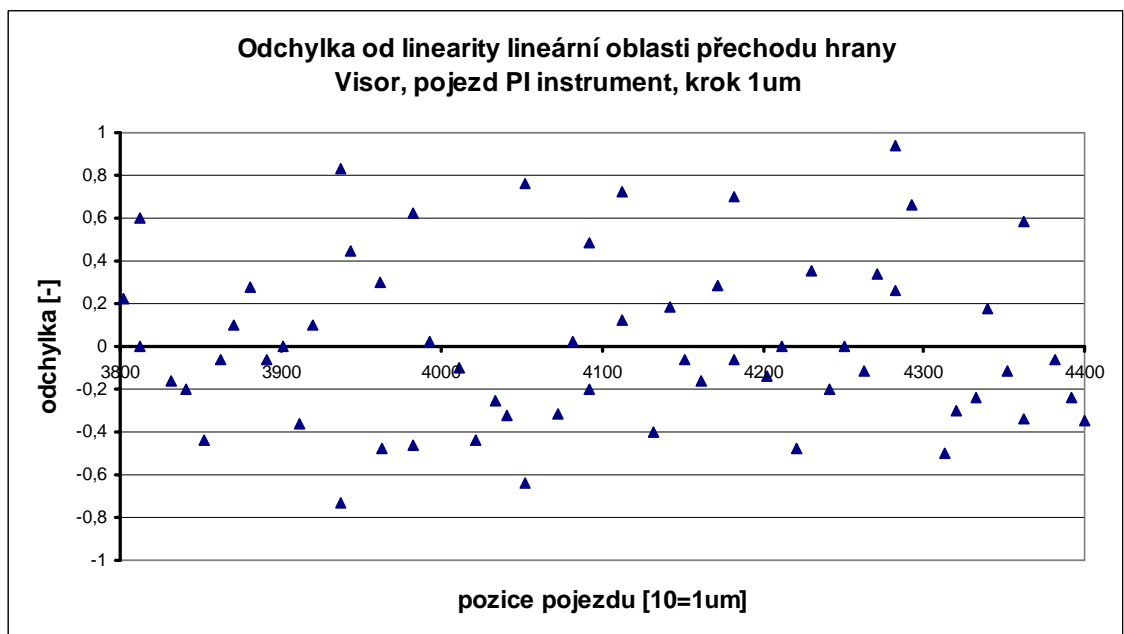
Výřez obrazu : 200x200, počet snímků pro průměrování 25, počet náměrů 1000, vyhodnocen řádek 100

Na obrázku 4.10 je vykreslen naměřený přechod hrany pomocí velmi přesného pojezdu physic instrument . Na obrázku 4.11 je vykreslena odchyly od linearity úseku přechodu hrany.





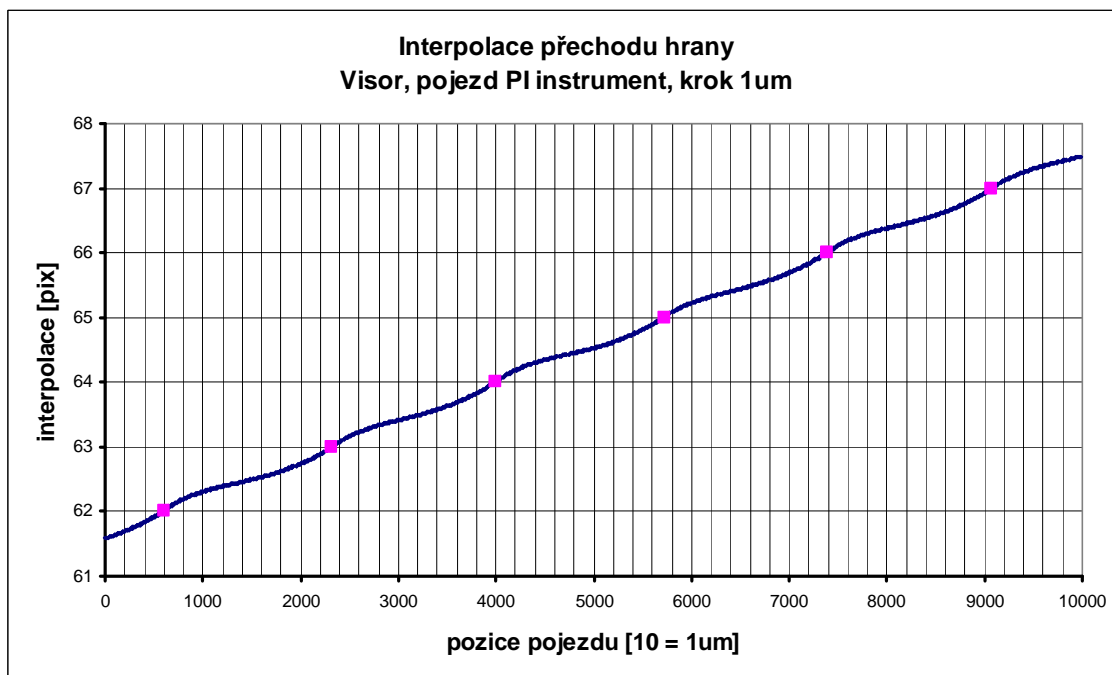
Obr. 4.10 Graf přechodu hrany – kamera Visor, pojezd PI instruments, krok 10um, průměrováno z 25 snímků



Obr. 4.11 Graf odchyłky od linearity lineární oblasti přechodu z grafu 4.10

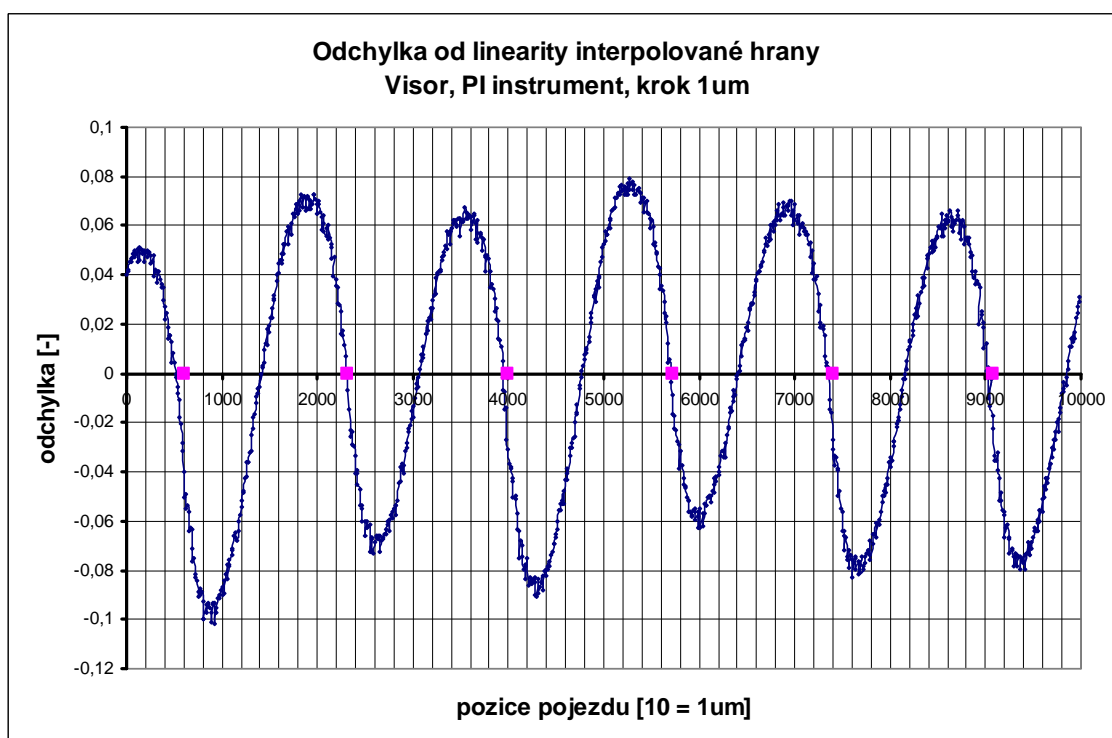
Dle grafu na obrázku 4.7 byla absolutní hodnota maximální odchyłky od linearity 0,94. Při pohledu na vykreslené hodnoty lze konstatovat, že se jedná o zcela nahodilé úrovně hodnot a tímto tyto odchyłky můžeme prohlásit za šum, který je generován senzorem. Nesmíme ovšem pominout fakt, že hodnoty byly průměrovány z 25 snímků.

Výsledky aplikace algoritmu lineární interpolace můžeme vidět na obrázku 4.12.



Obr. 4.12 Graf lineární interpolace hrany, kamera Visor, pojezd PI instrument, krok 1um

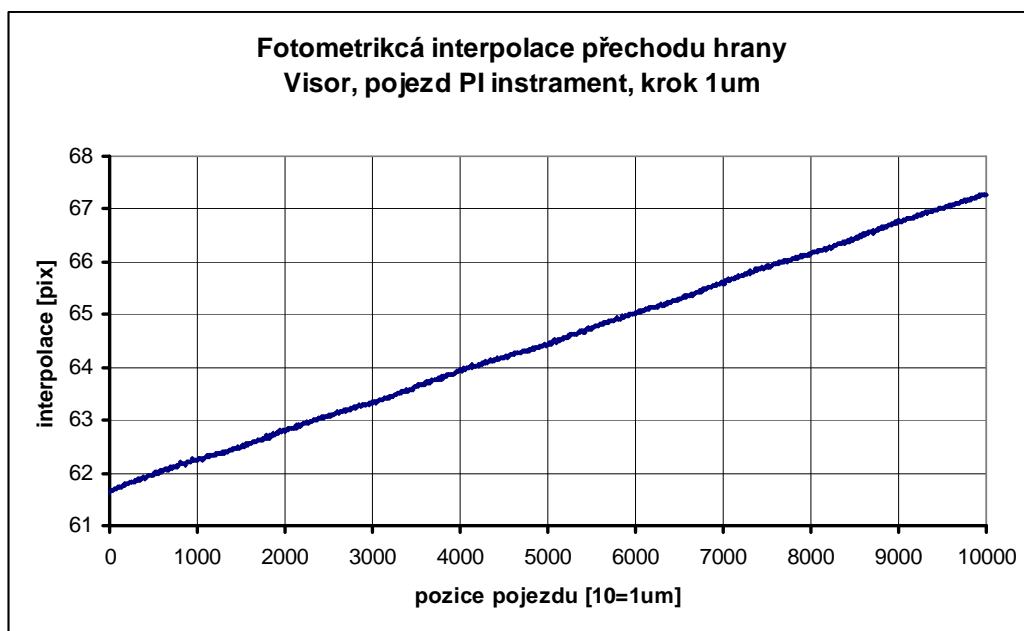
Již z grafu jsou patrné opakované periodicity, které vyznačují citlivost plochy pixelu na intenzitu ozáření. Fialové body označují jednotlivé úseky pixelů. Na obrázku 4.13 je vykreslen graf odchyly od linearity grafu 4.12 spolu s pozicemi pixelů.



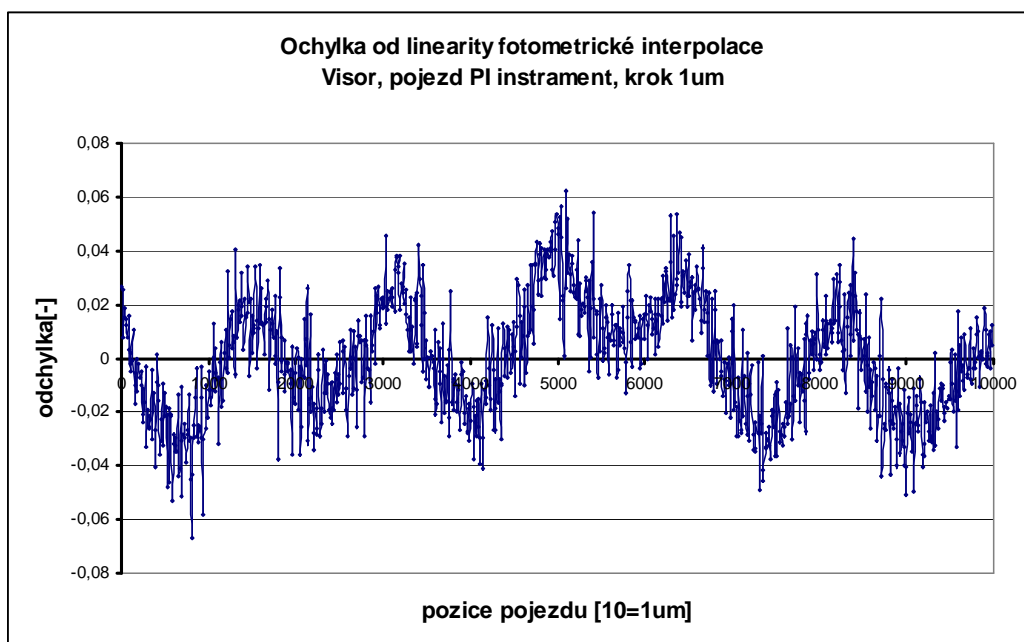
Obr. 4.13 Graf odchyly od linearity interpolačního grafu na obrázku 4.12

V grafu na obrázku 4.13 je maximální odchylka od linearity 0,1. Lze tedy říci, že přesnost měření pozice hrany pomocí lineární interpolace je v tomto případě s rozlišením na  $\frac{1}{10}$  pixelu.

Pro porovnání dosažených výsledků na obrázku 4.14 vynesena graf fotometrické interpolace a na obrázku 4.15 odchylka od linearity této fotometrické interpolace. Parametry pro vyhodnocení byly následující  $U_h = 220$ ,  $U_l = 5$ ,  $X_B = 40$ ,  $X_E = 80$ .



Obr. 4.14 Graf fotometrické interpolace hrany, kamera Visor, pojezd PI instrument, krok 1um



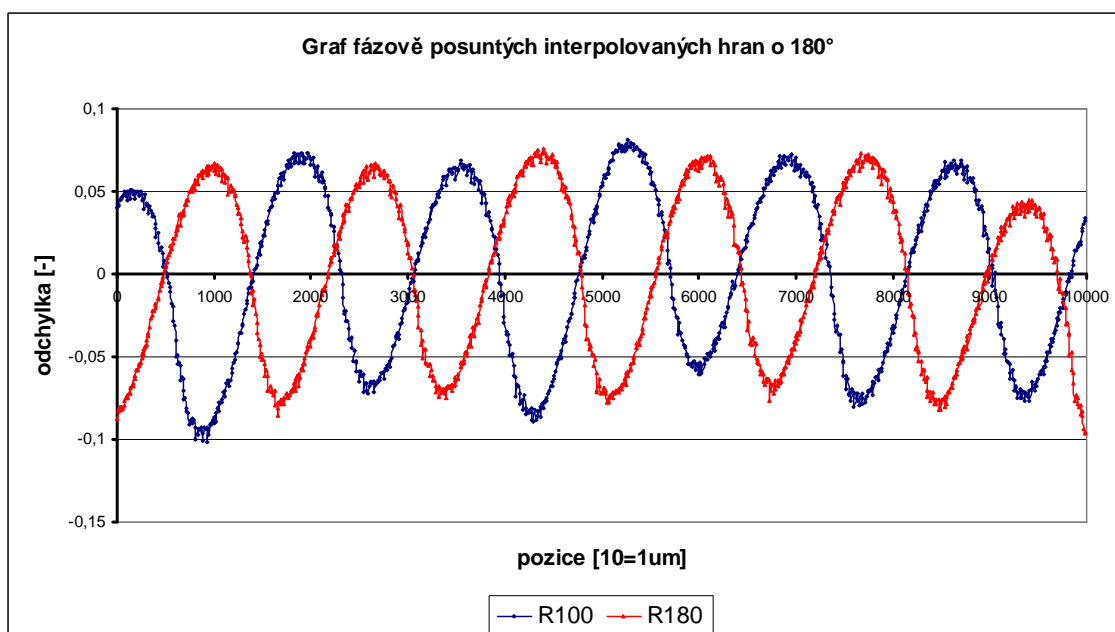
Obr. 4.15 Graf odchylky od linearity grafu na obrázku 4.14

V grafu na obrázku 4.15 je maximální odchylka od linearity 0,066. Lze tedy říci, že přesnost určení pozice hrany pomocí fotometrické interpolace je v tomto případě s rozlišením na  $\frac{1}{15}$  pixelu.

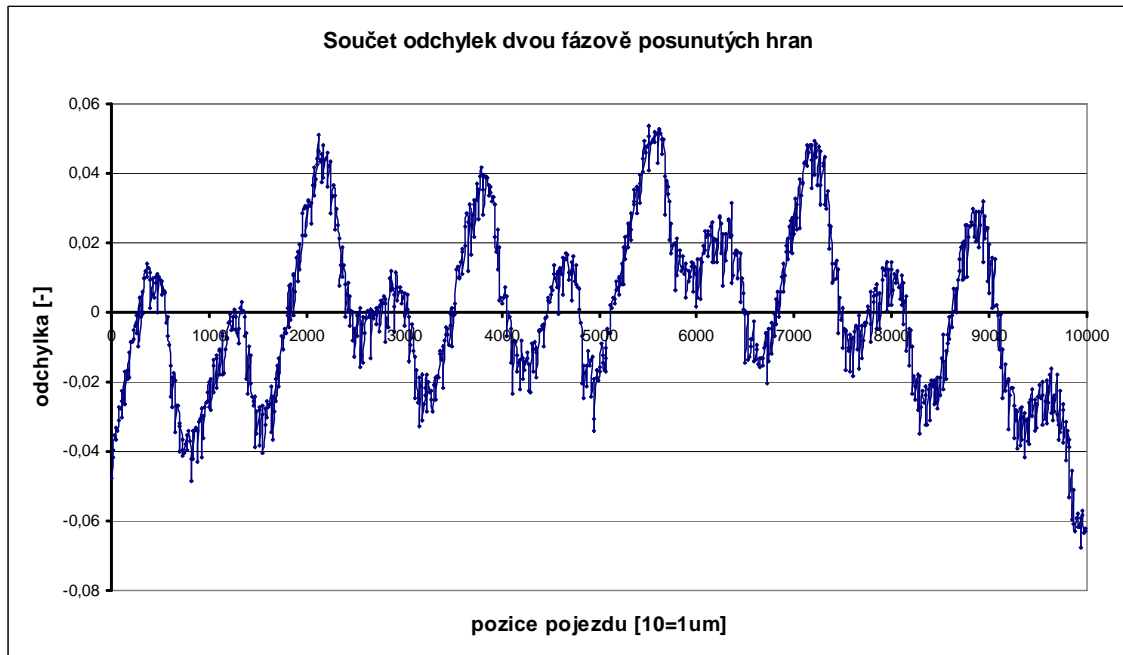
### 4.3.1 Zpřesnění interpolační metody, fázový posun hrany

Myšlenka této metody spočívá v nalezení takového řádku ve snímku, kde bude odchylka od linearity interpolace fázově posunutá oproti řádku výchozímu. Jinými slovy hledáme fázově posunutý pixel na hraně objektu o 180 stupňů. Tento efekt lze dosáhnout např. tím, že měřenou hranu dáme trochu šikmo.

Tato metoda byla ověřena na naměřených datech z předešlé kapitoly. Její výsledky ukazují obrázek 4.16 a obrázek 4.17. Výchozí řádek měl číslo 100 a řádek v protifázi hodnotu 180.



Obr. 4.16 Graf fázově posunutých interpolovaných hran.

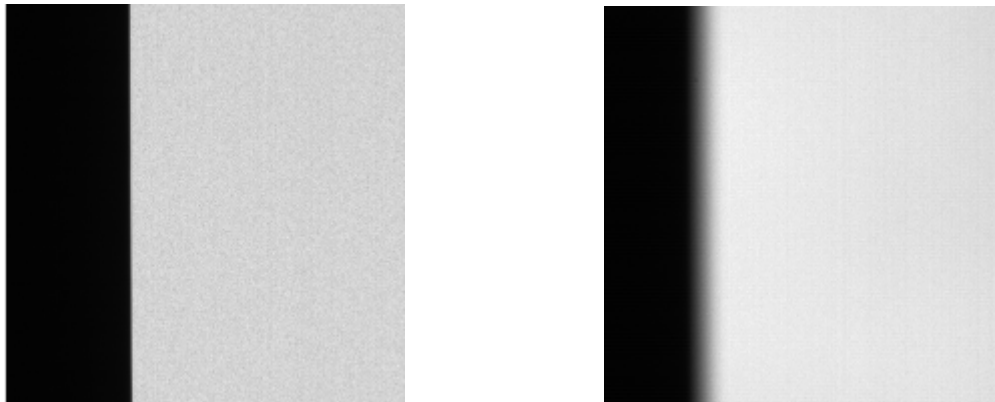


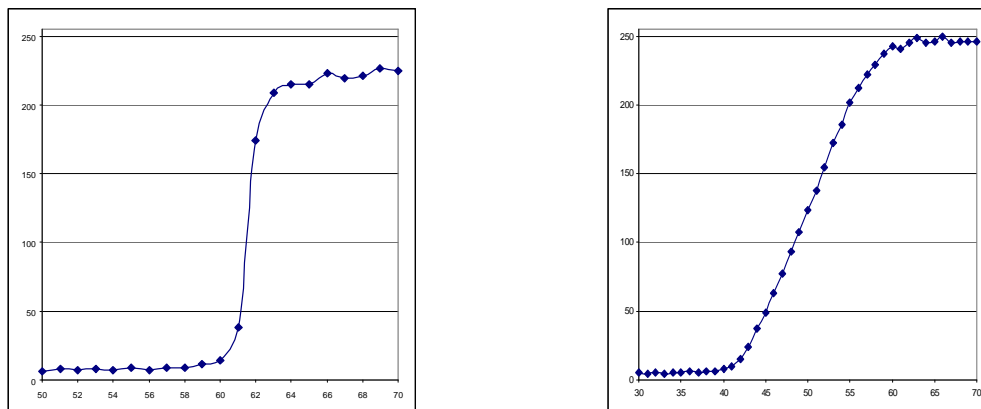
Obr. 4.17 Graf součtu dvou fázově posunutých hran

Z obrázku 4.17 je patrné, že skutečně došlo k částečnému zmenšení odchylky od linearity. Zmenšení chyby není, ale tak markantně vysoké, jak by se dalo očekávat. Určitě je důležité podotknout, že pro každou interpolaci jak řádku 100 tak řádku 180 byla spočítána vlastní regresivní přímka. Tím se ovšem vyskytuje otázka zdali je možné takto spojit dvě různá vyhodnocení dvou různých přímek. Došel jsem k názoru, že možné to je. Obě vygenerované přímky byly téměř rovnoběžně posunuté o konstantní vzdálenost. Jejich parametr sklonu se lišil v řádu tisícín což, v našem případě lze akceptovat. Takto velké rozlišitelnosti nejsme schopni prozatím v měření dosáhnout.

#### 4.4 Měření přechodu rozostřené hrany

Rozdíl mezi zaostřenou a rozostřenou hranou je vidět z následujících obrázku 4.18 a jejich strmosti přechodu.



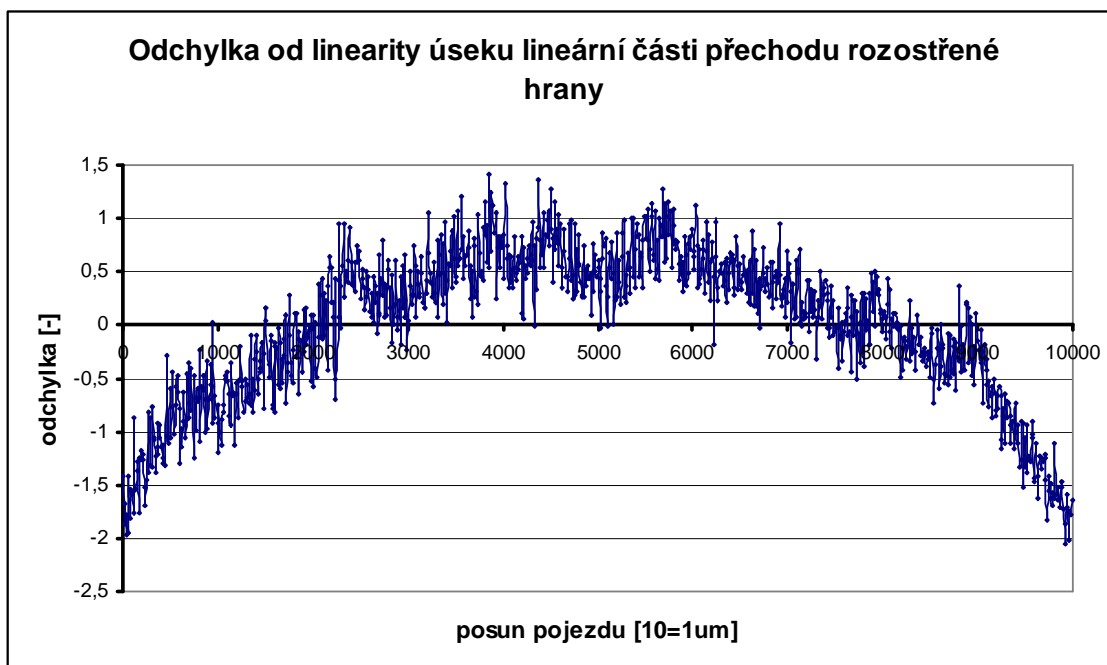


Obr. 4.18 obrázky vlevo zaostřená hrana, vpravo rozostřená hrana

Při zaostřeném stavu nám do přechodu hrany zasahují pouze 2 pixely. V případě rozostřené hrany je to pixelů 14. Tento fakt lze využít při zpřesnění polohy středu hrany. Mým úkolem bylo otestovat jak se bude takto rozostřená hrana chovat v případě aplikace lineární a fotometrické interpolace.

Bohužel při nastavování měření jsem hranu rozostřil příliš a posun pojezdu s krokem jednoho mikrometru a 1000 odměry nepřesáhl šířku celé hrany, která dle počtu cca 21 pixelů, je široká přes 3752 mikrometrů. Hrubým odhadem by toto měření trvalo 33 hodin a již nebylo reálné hranu z novu naměřit.

V datech, které jsem naměřil mohu použít vykreslení alespoň částečného úseku hrany. Nastavil jsem pro vyhodnocení vykreslení lineární oblasti přechodu a zobrazil odchylku od linearity, kterou můžeme vidět na obrázek 4.19



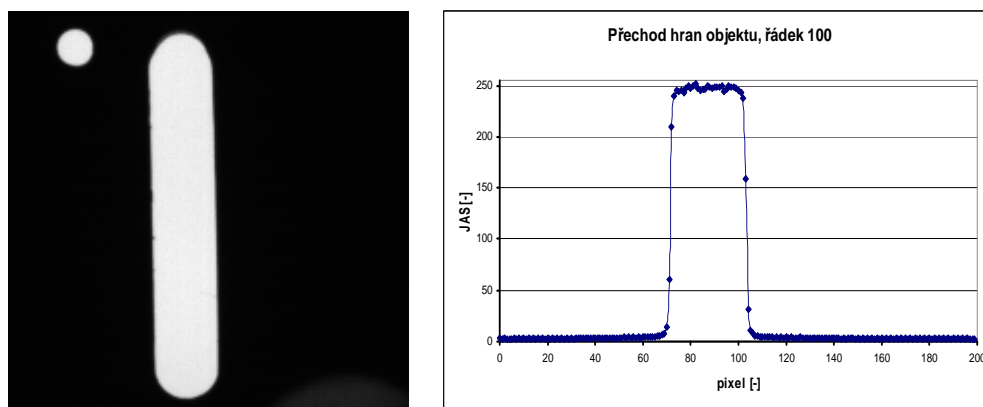
Obr. 4.19 Graf odchylky od linearity přechodu rozostřené hrany

Z grafu lze vidět, že lineární část hrany je spíše prohnutá než-li lineární. Tento fakt

může být způsoben množstvím pixelů, skrze které hrana přechází, nehledě na fakt, že nám výrazně zvyšuje chybu měření.

## 4.5 Hledání pozice středu objektu ohraničeném dvěma hranami

Tato metoda se snaží s maximální přesností nalézt pozici středu objektu, který má svoji velikost ohraničenou dvěma hranami. Objekt je zobrazen na obrázku 4.20 a vedle je vykreslen přechod hran jednoho snímku z řádku 100. Jedná se o podlouhý obdélník zakončený půlkružnicemi.



Obr. 4.20 Objekt pro měření pozice ohraničené dvěma hranami.

Tímto objektem jsem si posouval jemným krokem jako v předešlých případech a zároveň jsem aplikoval na obě hrany lineární nebo fotometrickou interpolaci. V každém kroku posunu získáme jednu hodnotu interpolace přechodu z černé do bílé a druhou hodnotu interpolace z bílé do černé. Z těchto dvou hodnot spočítáme střední hodnotu a měli bychom dle hodnoty interpolace určit pozici hrany.

### Vstupní informace :

Senzor : Visor 1,2Mpx / CMOS MT9M001

Krok pojezdu: 1um

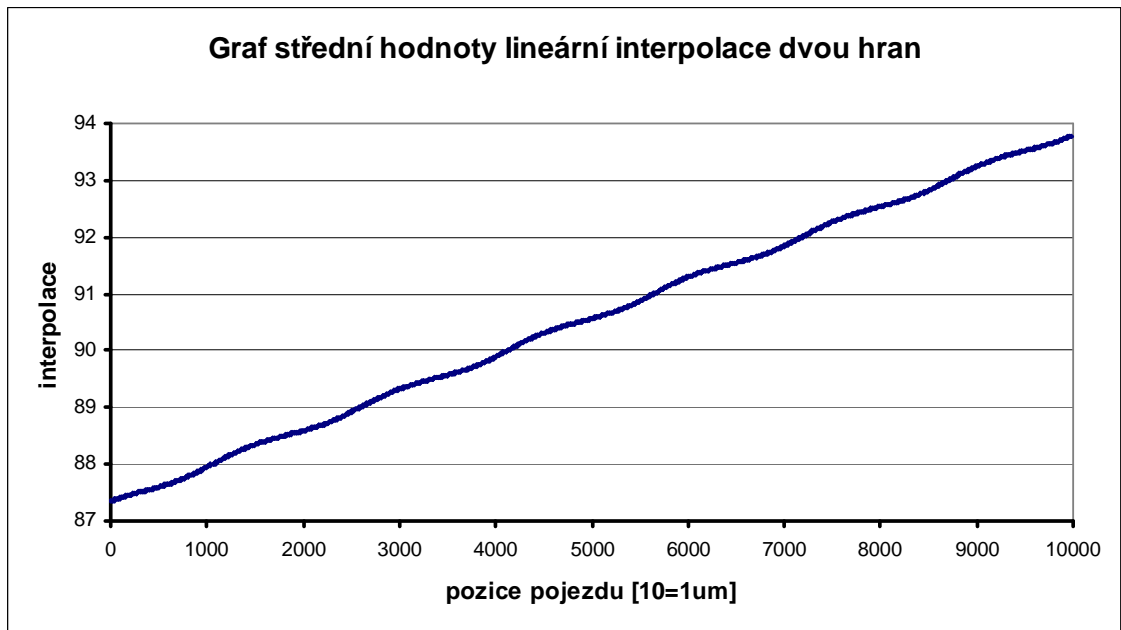
Zvětšení objektivu : 0,0347

Velikost pixelu: 5,2um

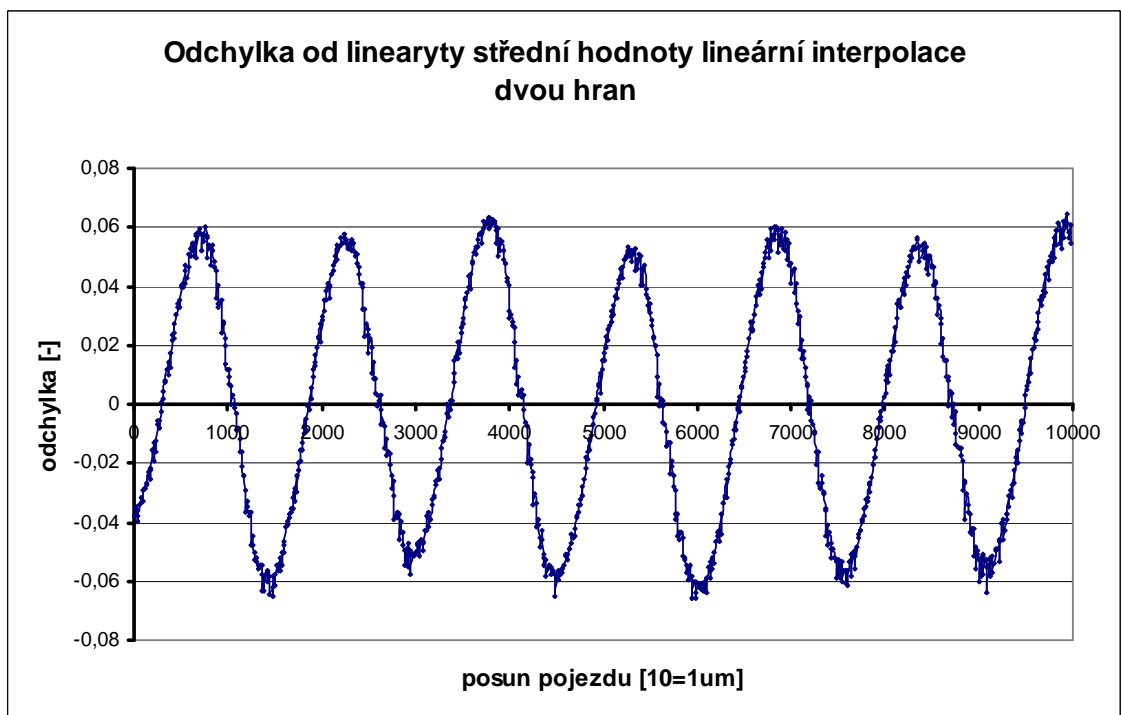
Backlight : LED, v místnosti zhasnuto

Výřez obrazu : 200x200, počet snímků pro průměrování 25, počet náměrů 1000, vyhodnocen řádek 100

Následující grafy na obrázku 4.21 a na obrázku 4.22 zobrazují vyhodnocení aplikace algoritmu lineární interpolace



Obr. 4.21 Graf střední hodnoty lineární interpolace dvou hran



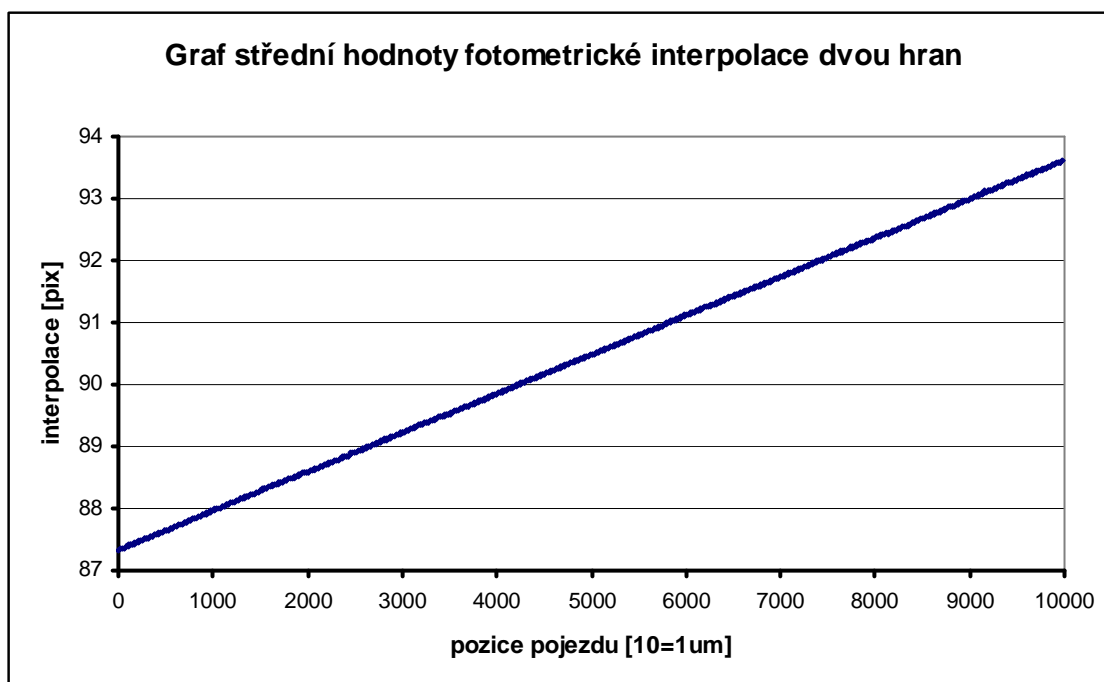
Obr. 4.22 Graf odchylky od linearity z grafu na obrázku 4.21

V grafu na obrázku 4.22 je maximální odchylka od linearity 0,0654. Lze tedy říci, že přesnost měření pozice středu objektů pomocí lineární interpolace je v tomto případě s rozlišením na cca  $\frac{7}{100}$  pixelu.

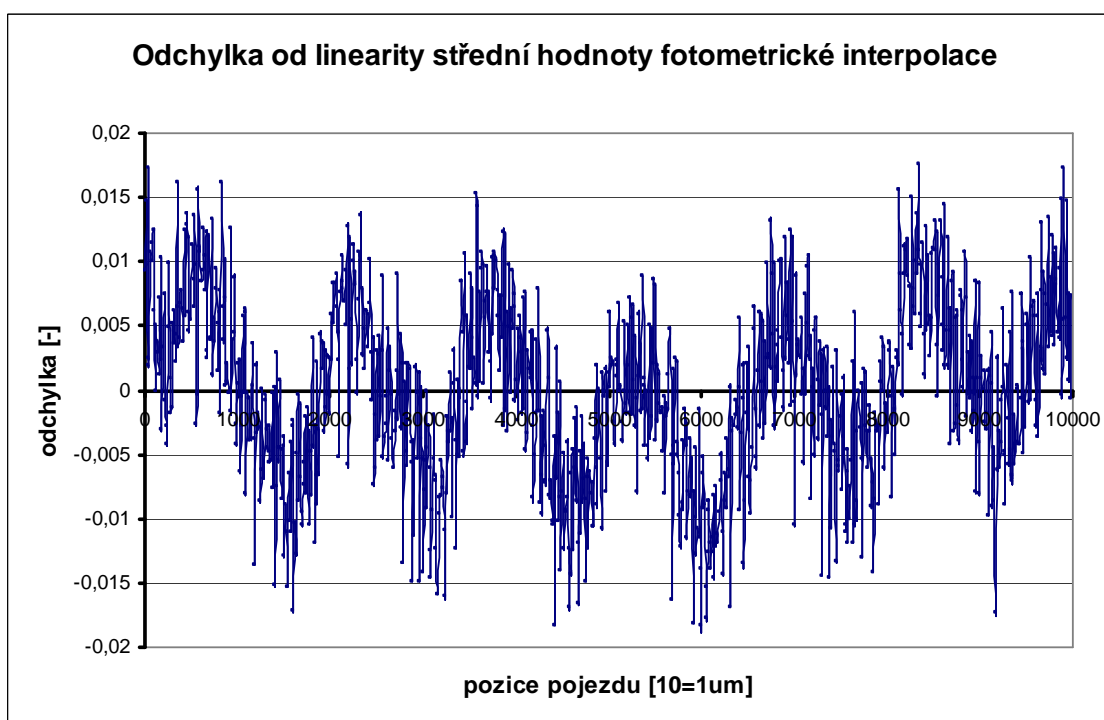
Obrázek grafu na obrázku 4.23 a na obrázku 4.24 zobrazují vyhodnocení aplikace



algoritmu fotometrické interpolace.



Obr. 4.23 Graf střední hodnoty fotometrické interpolace dvou hran



Obr. 4.24 Graf odchylky od linearity z grafu na obrázku 4.23

V grafu na obrázku 4.24 je maximální odchylka od linearity 0,018. Lze tedy říci, že přesnost měření pozice středu objektů pomocí fotometrické interpolace je v tomto případě s rozlišením na cca  $\frac{1}{50}$  pixelu.

## 4.6 Měření vzdálenosti posunu světelného bodu

Jako objekt pro měření byla zvolena difuzní LED dioda o průměru 5mm. Byla připojena k regulovatelnému zdroji napětí, abych mohl nastavit hodnotu vyzařovaného výkonu. Na obrázku 4.25 je ukázán způsob jakým byla připevněna a náhled snímku kamerou.



Obr. 4.25 Vlevo snímek z kamery světelného bodu, vpravo snímek upevnění diody na pojezd

Při měření bylo důležité, aby světelný bod nebyl v saturaci z důvodu dynamického rozložení světelného záření. Před zahájením měření, za okolního světla, bylo provedeno ostření a až poté zhasnuto.

### **Vstupní informace :**

Senzor : Visor 1,2Mpx / CMOS MT9M001

Krok pojezdu: 1um

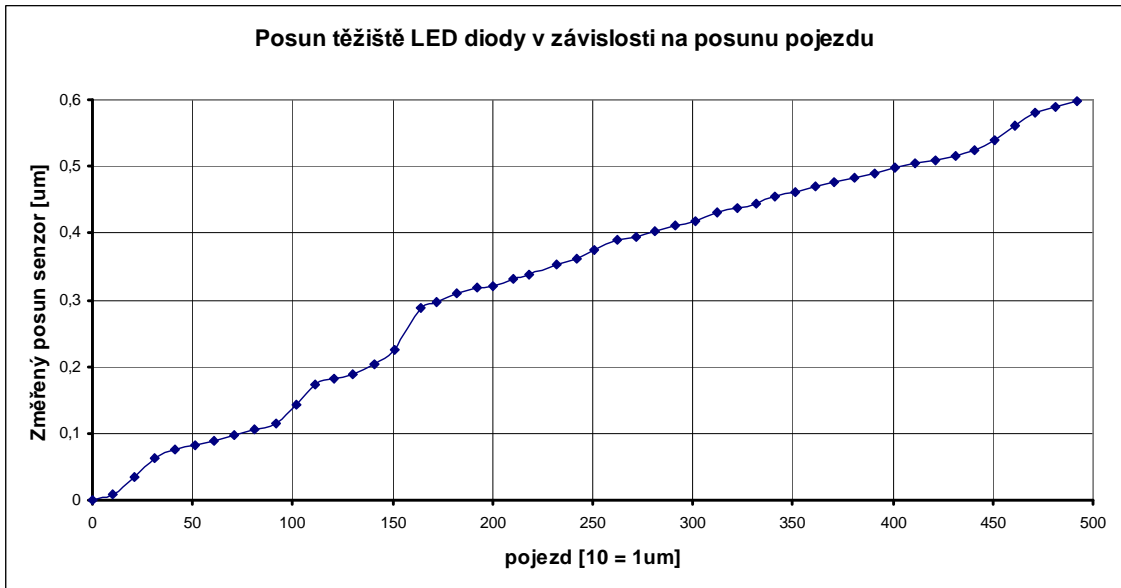
Zvětšení objektivu : 0,0305

Velikost pixelu: 5,2um

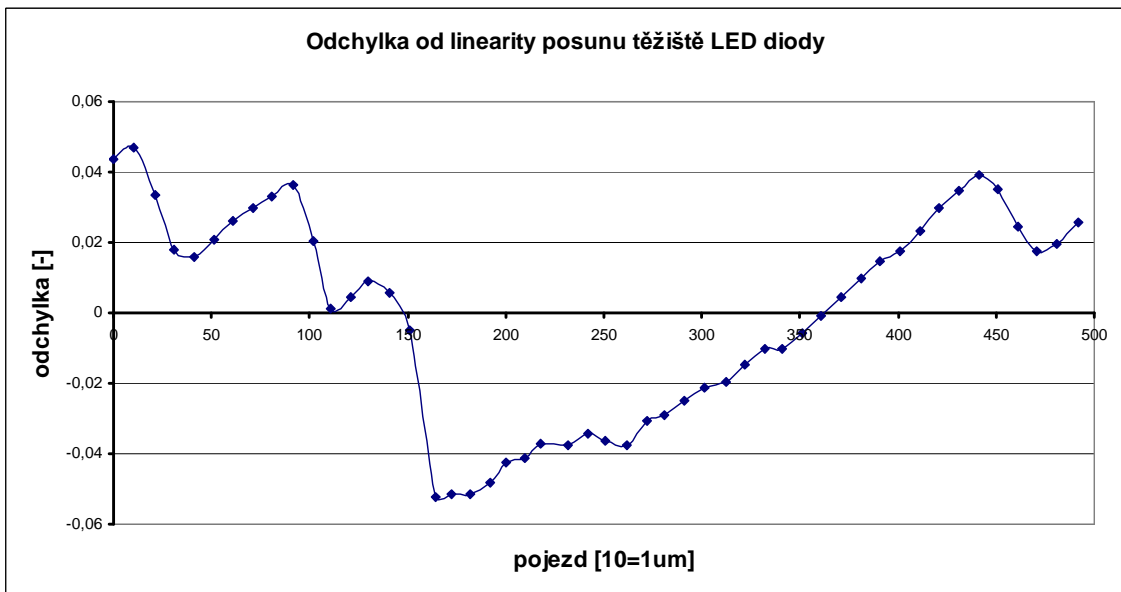
Backlight : off, v místnosti zhasnuto

Výřez obrazu : 200x200, počet snímků pro průměrování 25, počet náměrů 50, vyhodnocen řádek 100

Na obrázku 4.26 je graf závislosti posunu těžiště na reálném posunu pojezdu. Při bližším prozkoumání lze zjistit, že hodnota posunu těžiště na senzoru dosahuje pouze poloviny hodnoty skutečného pojezdu. Tuto skutečnost příkládám na vinu nedokonalému posunu v řádcích senzoru, ale i částečnému posunu ve sloupcích. Pro tento případ je aplikován na výpočet vztah pro vzdálenost dvou bodů v rovině.



Obr. 4.26 Graf posunu těžiště LED diody v závislosti na posunu pojezdu



Obr. 4.27 Graf odchylka od linearity posunu těžiště LED diody

V grafu na obrázek 4.27 je maximální odchylka od linearity 0,052. Pokud hodnotu přepočítáme na pixel můžeme říci, že přesnost měření pozice těžiště je pro tento případ s rozlišením na  $\frac{1}{100}$  pixelu.

V tomto případě bychom měli ještě zvážit jeden fakt o dosažené rozlišitelnosti a to je celkový posun bodu pouze o 50um. Ve skutečnosti tento posun nepřesáhne ani třetinu pixelu.

## 5 ZÁVĚR

Cílem této práce bylo vyhodnotit možnosti měření pomocí CCD a CMOS kamer se zaměřením na jednotlivé části měřícího ústrojí, ovlivňující výslednou kvalitu pořízeného snímku a poukázat na problematiku reálného nasazení v provozu. Dále diagnostikovat reálné schopnosti měření pomocí velmi rozšířeného senzoru typu CMOS a CCD a vytvořit pracoviště, kde bude možné ověřit metody pro detekci pozice hran a posunu světelných objektů.

Výsledky analýzy možností přesného měření rozměrů pomocí kamer jsou, z pohledu typu senzoru, srovnatelné. Největší podíl na kvalitě zpracování a též i přesnosti měření má samotná elektronika kamery a objektiv. Většina komerčně dostupných kamer bohužel nedisponuje možnostmi vyčíst data přímo ze senzoru, bez jakýchkoliv úprav, a to je pro tento účel znehodnocuje. Řešením zůstává použití kamer vyrobených na zakázku, podobně jako např. v mé práci používaná kamera Visor.

Dle zadání bylo vybudováno speciální pracoviště sloužící k měření a testování kamer. Konstrukce měřícího monolitu byla umístěna dvě podlaží pod budovou školy, vedle bývalého protiatomového krytu. Prostor místnosti byl velmi dobře izolován proti průniku vibrací a okolnímu hluku. Teplota prostředí si držela konstantní hodnoty jak v letním tak zimním období. Měření bylo možné provádět i za úplné tmy.

Programové vybavení bylo navrženo speciálně k měření posunu objektů pomocí pojezdových jednotek zapojených do osového kříže. Největší problémy vznikly při zprovoznění velmi přesného pojezdu od firmy Physicinstrument, kde nastavení podle manuálu neodpovídala skutečnosti. Při testování a pokusných měřeních, jedna jednotka přestala fungovat a nebylo možné ji nadále použít pro měření. Závada se pravděpodobně vyskytla na inkrementálním snímači pozice, který vysílal špatné informace řídicí jednotce.

Při jednom z mnoha odměrů byla zjištěna nedostatečná přesnost používané pojezdové jednotky ESSA a musel jsem instalovat již výše zmíněný pojezd od firmy Physicinstrument. Bylo též nutné přeprogramovat celkovou strukturu měřícího softwaru, aby bylo možné i v budoucnu jednoduchým způsobem přidat další typy kamer bez nutnosti rozsáhlých úprav kódu.

Software pro zpracování naměřených dat také prošel, v rámci této diplomové práce, velkými úpravami. Nejmarkantnější byl přechod z formátu ukládání po jednotlivých snímcích, ke zpracování dat z jednoho velkého souboru. Tato forma ukládání dat se velmi osvědčila a doporučuji ji používat i nadále.

Výsledky měření byly po celou dobu velmi podrobně diskutovány s mým vedoucím práce a byl kladen velký důraz na vztažné podmínky, které při měření nastaly.

Výsledky prokázaly, že pro velmi přesná měření, je nutné průměrovat až 300 hodnot, abychom mohli konstatovat, že výsledná hodnota signálu je téměř neměnná.

Šum za tmy, lze považovat za aditivní chybu, kterou můžeme díky průměrování

jednotlivých snímků výrazně snížit nebo při znalosti jeho profilu zcela odstranit.

S tímto souvisí i problematika kvality výroby senzorů typu CMOS. Měření ukázalo, že některé pixely mohou mít velmi odlišný proud za tmy oproti ostatním. S tímto faktem je nutné počítat při určení velikosti měřeného bodu např. při měření těžiště objektu. Světelný objekt musí mít větší plochu než je velikost 2 pixelů.

Při použití různých algoritmů pro vyhodnocení přechodu hrany vycházela jako nejlepší metoda fotometrická interpolace. U ní jsem byl schopen měřit polohu hrany s přesností na 1/50 pixelu. Tato metoda je ovšem velmi citlivá na nastavení vyhodnocovacích podmínek. Pro lineární interpolaci jsem dosáhl nejvyšší přesnosti 7/100 pixelu.

Při měření pohybu světelného objektu difúzní LED diody jsem dosáhl přesnosti 1/100 pixelu. To je, dle mého úsudku, již velmi slušný výsledek. Při přepočtu, na vzdálenost 50cm jsem byl schopen rozlišit změnu pozice o 2 mikrometry.

Z důvodu časové náročnosti měření se nepodařilo vyhotovit náměry bez projekční soustavy. Tento krok by vyžadoval náročný zásah do konstrukce celého měřicího systému a tím by znemožnil zpětné ověření již naměřených výsledků.

## REFERENCE

- [1] Molecular expressions [online] - dostupné dne 15.11.2010 na  
<http://micro.magnet.fsu.edu/primer/digitalimaging/concepts/fullframe.html>
- [2] CCD – server Wikipedie [online] - dostupné dne 10.11.2010 na  
<http://cs.wikipedia.org/wiki/CCD>
- [3] Sony ICX204AK – datasheet výrobce – viz. příložené CD
- [4] CMOS – Bildsensor [online] – dostupné dne 10.11.2010 na  
<http://www.ride-downhill.de/blog/?p=709>
- [5] Micron MT9M001 – datasheet výrobce – viz. příložené CD
- [6] CLIFF W., Measuring composite-video signal performance requires understanding differential gain and phase [online] - dostupné dne 15.11.2010  
<http://www.eetimes.com/design/industrial-control/4009973/Measuring-composite-video-signal-performance-requires-understanding-differential-gain-and-phase-Part-1-of-2>
- [7] Ověření výpočtů geometrické optiky [online] – dostupné dne 15.11.2010  
[http://measure.feld.cvut.cz/groups/edu/osv/Uloha\\_overeni.html](http://measure.feld.cvut.cz/groups/edu/osv/Uloha_overeni.html)
- [8] Optická zobrazovací soustava – výukové materiály – viz. příložené CD
- [9] RADIL Tomáš, Analysis and Optimisation of The Projection Methods of Dimension and Positron Measurement – DOCTORAL THESIS, 2005
- [10] ŠEDIVÝ J, Precision Single-line Dimension Measurement Usány CMOS Image Sensor and Photometric Interpolation, 2005 – článek CVUT FEL
- [11] PRIBULA O., Vývojový modul pre spracovanie obrazu so signálovým procesorom ADSP-Blackfin, 2007 – diplomová práce CVUT-FEL
- [12] PRIBULA O., Videometrické metody presného merania polohy a rozmerov objektov, 2009 odborná štúdia CVUT FEL
- [13] Free serial port monitor – software  
<http://www.serial-port-monitor.com/>
- [14] PRIBULA O., Precise Position Measurement with Low-Cost CMOS, dne 1.1.2010 nepublikovaný článek
- [15] Fischer, J : Optoelektronické senzory a videometrie. Skripta ČVUT, Praha 2002.
- [16] Heiden F. : Image Based Measurement Systéme. Hobojem, NJ, Wiley, 1994.
- [17] Kučera, J. : Systém pro automatické ověřování videometrických metod měření rozměru a polohy. – diplomová práce CVUT-FEL

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CMOS	Complementary Metal–Oxide–Semiconductor
CCD	Charge Coupled Devices
USB	Universal Seriól Bus
ADC	Analog Digital Converter
A/D	Analog/Digital převodník
ISO	International Organization for Standardization
IP	Internet Protocol
GSM	Global Systém for Mobile
3G	3rd Generation
UDP	User Datagram Protokol
http	Hypertext Transfer Protocol
RTP	Real Time Protocol
RTMP	Real Time Messaging Protocol
RTMPT	Real Time Messaging Protocol Tuneling
OTF	Optical Transfer Function

# SEZNAM PŘÍLOH

A Obash DVD

70



## **A OBASH DVD**

Zdrojové kódy

- obsahuje zdrojové kódy aplikací

Software

- vyexportované programy a knihovny

Export – námery

- vyexportované náměry do excelu

Screen – aplikace

- screenshoty aplikací

Materiály

- datasheety a jiné materiály použité při zpracování práce