

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra experimentální fyziky



Diplomová práce

Název práce:

**Vývoj algoritmů pro automatizaci měřícího procesu  
optické phase-shifting profilometrie**

Autor:

Studijní program:

Studijní obor:

Forma studia:

Vedoucí práce:

Datum zadání práce:

Datum odevzdání práce:

Radek Baránek

Fyzika

Aplikovaná fyzika

Prezenční

Mgr. Michal Pochmon

prosinec 2009

duben 2011

## **Prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval sám pod vedením Mgr. Michala Pochmona za použití materiálů uvedených na konci práce.

Podpis autora:

## **Poděkování**

Na tomto místě bych chtěl poděkovat mému vedoucímu Mgr. Michalu Pochmonovi za trpělivou pomoc a odborné rady při vytváření této diplomové práce.

## Bibliografická identifikace

Autor:	Radek Baránek
Název práce:	Vývoj algoritmů pro automatizaci měřicího procesu optické phase-shifting profilometrie
Typ práce:	Diplomová práce
Pracoviště:	Katedra experimentální fyziky
Vedoucí práce:	Mgr. Michal Pochmon
Rok obhajoby práce:	2011
Počet stran:	53
Počet příloh:	0
Jazyk:	Čeština

### Abstrakt

Diplomová práce se zabývá vývojem algoritmů pro automatizaci měřicího procesu optické phase-shifting profilometrie (PSP). Nejprve je obecně popsán princip triangulace, který tvoří základ PSP. Dále je navázáno popisem PSP, jakožto neinterferometrické optické měřicí metody pro získávání kvantitativních informací o profilu povrchu předmětu. Hlavní část práce tvoří popis jednotlivých úkonů spojených s nastavením, měřením a kalibrací experimentální sestavy pro PSP a následně také jejich automatizace pomocí matematického prostředí MATLAB<sup>®</sup>. Poslední část práce se zabývá vytvořením uživatelského grafického prostředí, které implementuje algoritmy popsané v předešlých částech práce a vytváří tak rozhraní pro jednoduchou obsluhu celé experimentální sestavy PSP.

**Klíčová slova:** phase-shifting profilometrie (PSP), 3D metrologie, digitální zpracování obrazu, MATLAB<sup>®</sup>.

## **Bibliographical identification:**

Author:	Radek Baránek
Title:	Development of algorithms for automation of measuring process of phase-shifting profilometry
Type of thesis:	Diploma
Department:	Department of experimental physics
Supervisor:	Mgr. Michal Pochmon
The year of presentation:	2011
Number of pages:	53
Number of appendices:	0
Language:	Czech

### **Abstract**

This diploma thesis deals with development of algorithms for automation of measuring process of phase-shifting profilometry (PSP). At first the shape profile measuring principle, triangulation is described. Then the phase-shifting profilometry, measuring method for quantitative measuring of object shape, is mentioned. The main part of thesis consists of description of particular operations connected with adjusting, measuring and calibration of experimental setup for PSP. Subsequently algorithms for their automation are discussed. For developing, MATLAB<sup>®</sup> programming environment is used. The last part is about implementation of developed algorithms into the Graphic User Interface.

**Keywords:** phase-shifting profilometry (PSP), 3D metrology, digital image processing, MATLAB<sup>®</sup>.

# Obsah

<b>1</b>	<b>Úvod</b> .....	<b>7</b>
1.1	Cíle diplomové práce .....	7
<b>2.</b>	<b>3D profilometrie</b> .....	<b>8</b>
2.1	Princip triangulace .....	9
2.2	Phase-shifting profilometrie .....	10
<b>3.</b>	<b>Programové prostředí MATLAB®</b> .....	<b>15</b>
3.1	Experimentální sestava pro PSP a její ovládání .....	15
3.1.1	Ovládání dataprojektoru z prostředí MATLAB .....	16
3.1.2	Ovládání kamery z prostředí MATLAB.....	18
<b>4.</b>	<b>Vývoj algoritmů pro automatizaci měření</b> .....	<b>21</b>
4.1	Korekce přenosové funkce dataprojektoru.....	21
4.1.1	Algoritmus korekce přenosové funkce dataprojektoru.....	23
4.2	Kalibrace příčných os (osy x a y).....	28
4.2.1	Kalibrace pomocí sinusové mřížky .....	29
4.2.2	Algoritmus kalibrace pomocí sinusové mřížky .....	34
4.2.3	Kalibrace vyšetřováním obsahu obrazce .....	36
4.2.4	Srovnání obou metod kalibrace příčných os.....	39
4.3	Proces měření metodou PSP .....	40
4.3.1	Algoritmus měřicího procesu .....	41
4.4	Kalibrace podélné osy (osy z).....	43
4.4.1	Algoritmus kalibrace osy z.....	45
<b>5.</b>	<b>Implementace algoritmů do grafického uživatelského rozhraní</b> .....	<b>47</b>
5.1	Popis grafického uživatelského rozhraní.....	48
<b>6.</b>	<b>Závěr</b> .....	<b>52</b>
	<b>Seznam použité literatury:</b> .....	<b>53</b>

# 1 Úvod

Pro pochopení celé řady měřících metod je třeba znát často poměrně složitou teorii. Možností, jak zpřístupnit takové metody personálu, který nemá požadovanou kvalifikaci, je automatizace procesů spojených s takovou měřící metodou. Automatizace tedy sehrává důležitou roli v rozšiřování pokročilých a přesných měřících metod do nejrůznějších oblastí, kde by tyto metody byly z důvodu omezené kvalifikace personálu nepoužitelné. Celkově tedy přispívá k rozšiřování oblastí, kde lze danou měřící metodu použít.

Měřící metoda, Phase-shifting profilometrie (PSP), které se věnuje tato práce, je nekoherentní, nekontaktní, optická metoda pro měření profilu předmětu. Je tedy zřejmé, že pro automatizaci bude použit počítač. Automatizace bude spočívat ve vývoji algoritmů, jež budou automaticky provádět jednotlivé kroky spojené jak s měřením, tak s nastavením a kalibrací. Snahou bude minimalizovat nároky na obsluhu, ale zároveň zachovat určitý stupeň variability.

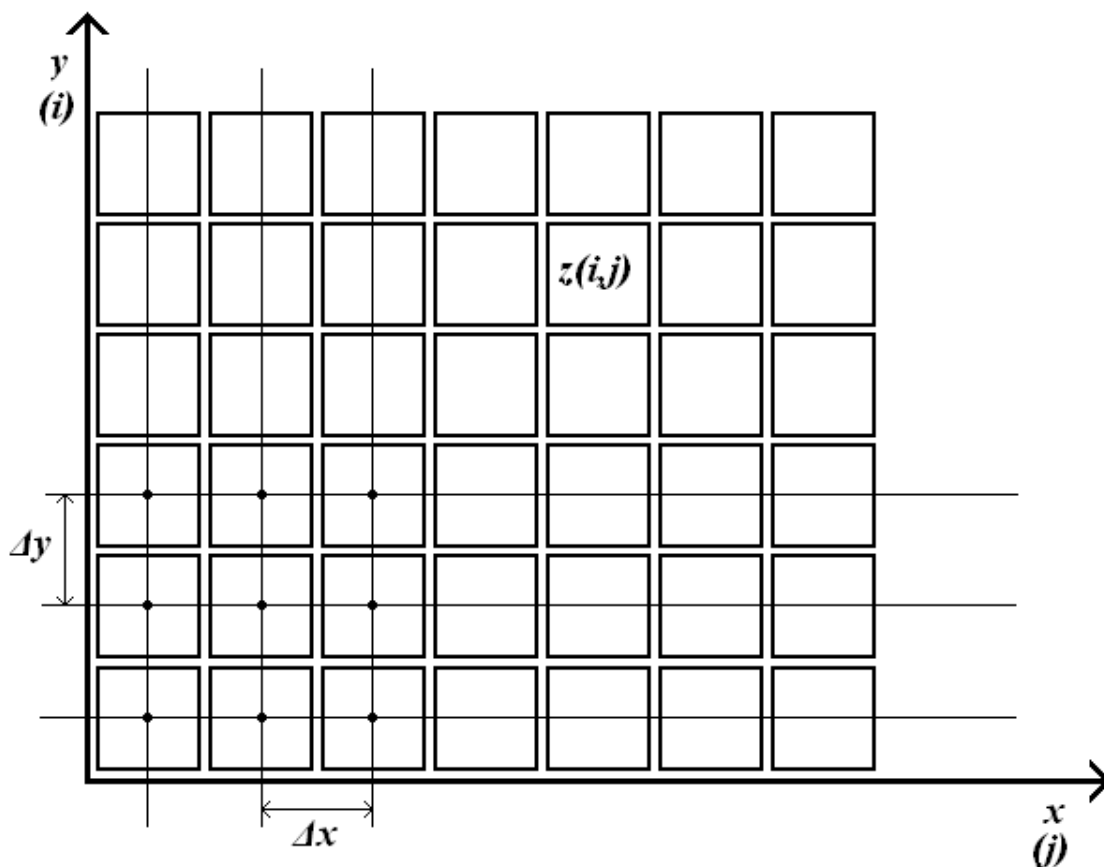
## 1.1 Cíle diplomové práce

Cílem diplomové práce bude popsat vývoj ovládacího programu pro obsluhu měřící sestavy pro Phase-shifting profilometrii. Začátkem budou popsány základní principy PSP a realizace experimentální sestavy. Popsány budou také jednotlivé úkony spojené s nastavením a kalibrací sestavy. Následně budou tyto úkony rozebrány s ohledem na jejich automatizaci. U každého úkonu bude popsán způsob a důvody automatizace. Algoritmy sloužící pro automatizaci budou vyvíjeny v matematickém prostředí MATLAB<sup>®</sup>. Závěrem bude okomentováno vytváření grafického uživatelského prostředí, které bude implementovat předchozí algoritmy a finálně tak vytvoří rozhraní mezi uživatelem a měřící sestavou.

## 2. 3D profilometrie

Tato práce se zabývá vývojem algoritmů pro automatizaci měření profilu metodou PSP. Je proto nezbytné uvést alespoň základy použité metody, aby bylo možné pochopit jednotlivé úkony při měření, nastavení a kalibraci a také jejich automatizaci. Pod pojmem měření profilu rozumíme kvantitativní určení plochy definující obálku měřeného předmětu. Z matematického hlediska se tato plocha skládá z nekonečně mnoha trojdimenzionálních bodů. Přesné určení plochy by tedy vyžadovalo znalost souřadnic všech těchto bodů.

V praxi však nejsme schopni zpracovat nekonečně mnoho bodů. Plochu tedy popisujeme množinou s konečným počtem bodů. S ohledem na úsporu paměti a jednoduchost reprezentace je výhodné nezaznamenávat všechny souřadnice pro jednotlivé body, nýbrž reprezentovat množinu bodů maticí. Vhodnou volbou soustavy souřadnic je možné zakódovat dvě souřadnice bodu do polohy prvku v matici. Např. sloupec, ve kterém je daný prvek matice, určuje souřadnici  $x$  a řádek určuje souřadnici  $y$ . Hodnota prvku pak reprezentuje souřadnici  $z$ . Reprezentace množiny bodů pomocí matice je patrná z obr. 2.1.



Obr. 2.1 Reprezentace množiny 3D bodů pomocí matice



Poloha prvku ve dvoj-dimenzionální matici je obvykle určena dvěma celočíselnými indexy  $i$  a  $j$ . Pro přepočítání z indexů na  $x$ -ové a  $y$ -ové souřadnice je třeba znát hustotu rastrování měřené plochy. Za předpokladu, že je rastrování rovnoměrné a stejné jak pro osu  $x$  tak pro osu  $y$ , stačí znát jen jeden parametr. Ten udává vzdálenost mezi dvěma body, které odpovídají dvěma sousedícím prvkům matice ( $\Delta x$  a  $\Delta y$  viz obr. 2.1). S určováním tohoto parametru bude souviset algoritmizace kalibrace příčných os, která bude popsána dále v této práci.

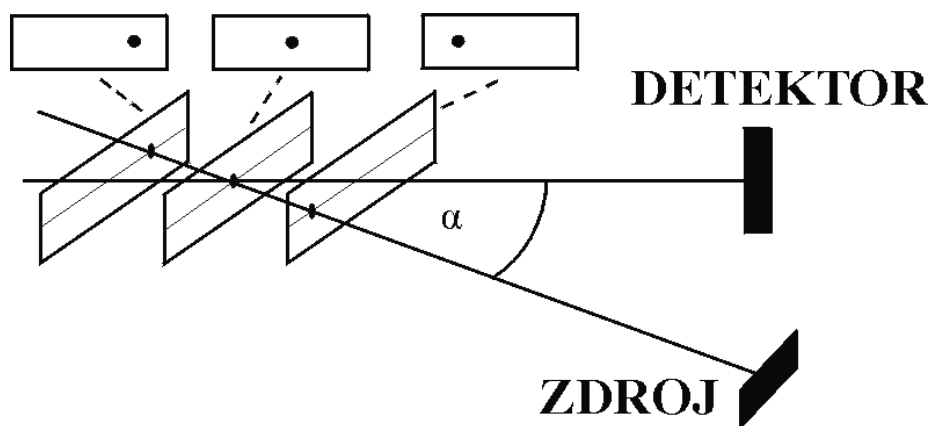
Pro měření profilu existuje mnoho měřících metod. Optické profilometrické měřící metody jsou s oblibou využívány především pro bezkontaktnost (při měření nedochází k mechanickému kontaktu s měřeným objektem). Vyhodnocení naměřených dat u optických metod bývá často složité, ovšem při výpočetní síle dnešních počítačů je tato nevýhoda zanedbatelná. Jeden ze způsobů jak opticky měřit profil je promítání speciálních optických struktur na měřený předmět. Nerovnost povrchu měřeného předmětu způsobí deformace promítnuté optické struktury. Vyhodnocením těchto deformací lze obdržet informace o profilu povrchu předmětu. Mezi metody využívající výše zmíněný způsob patří např. moiré topografie, fourierovská profilometrie (FTP) nebo phase-shifting profilometrie. Všechny tyto uvedené metody jsou založeny na principu triangulace.

## 2.1 Princip triangulace

Princip triangulace tvoří základ pro většinu nekoherentních optických profilometrických metod. Umožňuje získat informaci o vzdálenosti předmětu ve směru pozorování. Tvoří-li jednotlivé prvky sestavy (předmět, snímač, zdroj) trojúhelník, pak vzdálenost předmětu od snímače je možné určit z polohy stopy, kterou promítá zdroj na předmět. Tato situace je patrná z obr. 2.2, kde je ilustrována změna polohy stopy v závislosti na změně vzdálenosti v ose pozorování. Podmínkou je, aby úhel  $\alpha$  nabýval hodnot z intervalu  $(0, \pi/2)$ . Vzdálenost předmětu od detektoru je tedy možné získat nepřímo z polohy stopy na předmětu. Pokud uvažujeme detektor i zdroj v nekonečnu, potom mezi změnou polohy stopy a změnou vzdálenosti v ose pozorování platí jednoduchý lineární vztah:

$$\Delta z = k \cdot \Delta x, \quad (2.1)$$

kde  $\Delta z$  je změna vzdálenosti v ose pozorování,  $\Delta x$  je změna polohy stopy a  $k$  je konstanta závislá na úhlu  $\alpha$ .



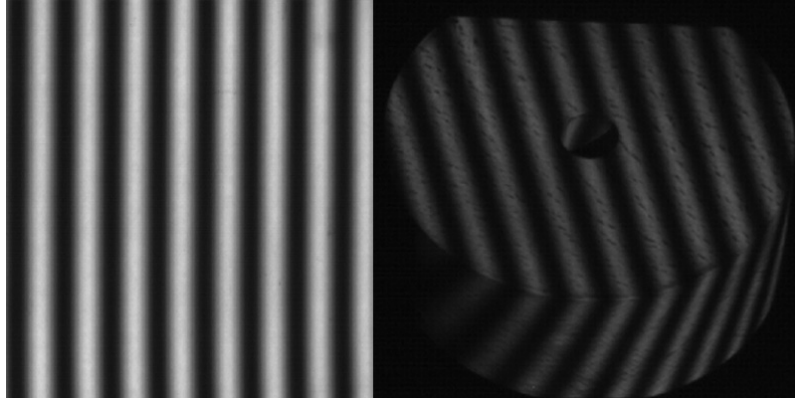
Obr. 2.2 K vysvětlení principu triangulace

Je zřejmé, že úhel, který svírá osa zdroje s osou detektoru, má za výše uvedeného předpokladu lineární vztah vzhledem ke konstantě  $k$ . Čím větší je tedy hodnota úhlu  $\alpha$ , tím větší pozorujeme změny v posuvu stopy ve směru kolmém na osu detektoru při stejné změně vzdálenosti předmětu od detektoru.

## 2.2 Phase-shifting profilometrie

Phase-shifting profilometrie patří mezi měřicí metody získávající informaci o profilu povrchu měřeného předmětu z deformace promítnuté optické struktury na měřený předmět. Řadí se tak mezi nekoherentní optické profilometrické metody. Mezi výhody této metody patří především relativně jednoduchý princip a také cena vybavení potřebného pro realizaci experimentální sestavy. Základ experimentální sestavy tvoří zdroj pro promítání optických struktur, referenční rovina, kamera a vyhodnocovací zařízení, které je dnes téměř bez výjimky realizováno počítačem.

Princip metody spočívá ve vyhodnocování snímků z maticového detektoru, který snímá objekt, na jehož povrch je zdrojem promítnuta sinusová mřížka. Sinusová mřížka je dvoj-dimenzionální optická struktura, jejíž řádkový průběh intenzity vykazuje sinusový průběh [1]. Příklad sinusové mřížky a také její deformace po promítnutí na předmět je na obr. 2.3. Snímky jsou vyhodnocovány za účelem určení fáze pro každý bod maticového detektoru. Fáze je známá z popisu harmonických signálů, kde udává, v jaké části jedné periody se daný bod průběhu nachází. Tento pojem lze však zobecnit i na částečně harmonické signály, kterým je např. řádek zdeformované sinusové mřížky.



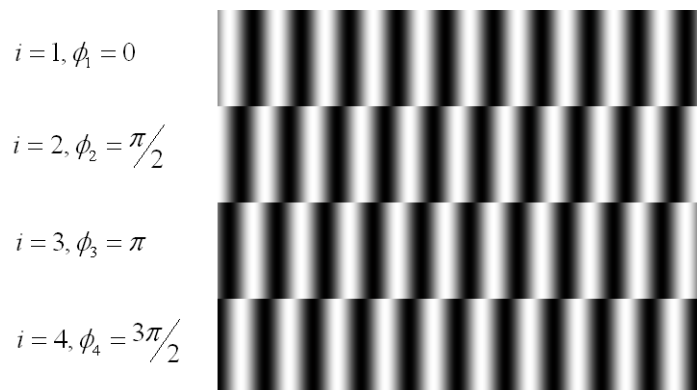
Obr. 2.3 Sinusová mřížka a její deformace nerovným povrchem předmětu

Informaci o výšce (souřadnice určená osou detektoru) předmětu v daném bodě lze získat právě za pomoci fáze zdeformovaného sinusového průběhu. Tvoří-li detektor, měřený předmět a zdroj trojúhelník, pak vzhledem k principu triangulace dochází při změnách výšky měřeného předmětu k lokálnímu posunu sinusové mřížky a tudíž také její fáze. Určíme-li nejprve fázi pro jednotlivé body snímku tzv. referenční roviny, na kterou je promítnuta sinusová mřížka, pak rozdíl fáze referenční roviny a fáze předmětu nese informaci o výšce předmětu v daném bodě vzhledem k referenční rovině, která udává nulovou výšku.

Cílem metody je tedy určit fázi pro jednotlivé body referenční roviny a měřeného předmětu. Rozdílem těchto fází dostáváme pro jednotlivé body informaci o výšce. Pro určení fáze touto metodou je nutné získat alespoň 3 snímky předmětu (resp. referenční roviny), kdy fáze promítané sinusové mřížky je pro každý snímek jiná, určená ze vztahu:

$$\phi_i = \frac{2(i-1)\pi}{N}, \quad (2.2)$$

kde  $i$  označuje číslo snímku,  $\phi_i$  označuje fázi promítané sinusové mřížky pro  $i$ -tý snímek a  $N$  udává celkový počet snímků. Posun fáze promítaných sinusových mřížek pro  $N = 4$  je ilustrován na obr. 2.4.



Obr. 2.4 Posun fáze sinusové mřížky pro jednotlivé snímky

Označíme-li funkci intenzity jednotlivých snímků  $I_i = I_i(x, y)$ , pak pro výpočet fáze lze odvodit vztah [2]:

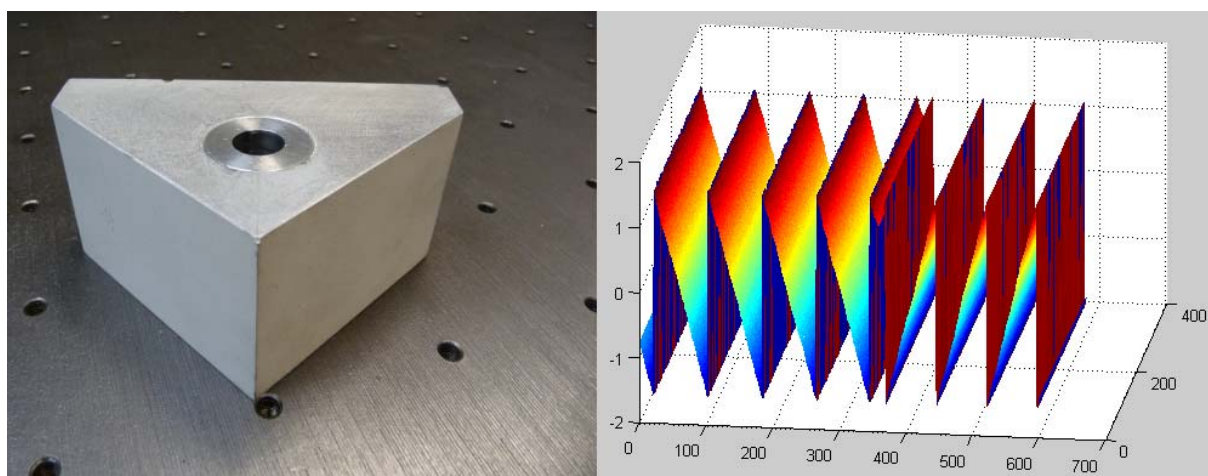
$$\phi(x, y) = \arctan \left[ \frac{\sum_{i=1}^N I_i(x, y) \sin(2\pi i/N)}{\sum_{i=1}^N I_i(x, y) \cos(2\pi i/N)} \right], \quad (2.3)$$

kde  $\phi(x, y)$  označuje funkci fáze a  $N$  označuje celkový počet snímků. Pokud zvolíme speciálně  $N = 4$ , vzorec pro výpočet fáze se vlivem funkčních hodnot goniometrických funkcí pro násobky  $\pi/2$  značně zjednoduší. V tomto speciálním případě pro výpočet fáze platí:

$$\phi(x, y) = \arctan \left[ \frac{I_1(x, y) - I_3(x, y)}{I_2(x, y) - I_4(x, y)} \right], \quad (2.4)$$

Pro fázi  $\phi(x, y)$  vypočtené z výše uvedených vztahů (2.3) a (2.4) platí, že může nabývat hodnot pouze z intervalu  $\left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle$ .

Měřením spojitého povrchu dostáváme fázi, která je obecně spojitá jen v malých oblastech. Tyto oblasti jsou od sebe odděleny skoky o  $\pi$  (resp.  $-\pi$ ). Tento jev se většinou označuje jako zabalování fáze (wrapping) a objevuje se u většiny profilometrických metod využívající promítání periodické optické struktury na předmět. Tento jev je ilustrován na obr. 2.5, kde je pomocí 3D grafu reprezentována matice fáze, která vznikla měřením trojúhelníkového hranolu.



Obr. 2.5 JeV zabalování fáze

Pro získání nezabalené fáze se na zabalenou matici fáze aplikuje tzv. rozbalovací (unwrapp) algoritmus. Tyto algoritmy bývají často složité a výpočet rozbalené fáze pomocí těchto algoritmu časově náročný. Navíc při velkém podílu šumu ve fázové matici jsou tyto algoritmy zdrojem sekundárních chyb [1]. Vhodné je tedy před aplikováním rozbalovacího algoritmu připravit matici fáze s co nejmenším počtem fázových skoků. Polovinu fázových skoků lze eliminovat jednoduchou úvahou a úpravou vzorce již při výpočtu fázové matice. Funkce arctan vrací hodnoty z intervalu  $\left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle$ , fáze sinusového průběhu však může nabývat hodnot s intervalu  $\langle -\pi, \pi \rangle$ . Pro získání fáze v celém tomto rozsahu je po výpočtu funkce arctan dále nutné vyšetřovat znaménka čitatele a jmenovatele argumentu funkce arctan ve vzorci (2.3) nebo (2.4). Tabulka 2.1 ukazuje rozšíření oboru fáze vyšetřováním znaménka čitatele a jmenovatele při výpočtu z obecného vzorce  $\phi = \arctan\left(\frac{a}{b}\right)$ .

znaménko $a$	znaménko $b$	fáze $\phi$
+	+	$\arctan\left(\frac{a}{b}\right)$
+	-	$\arctan\left(\frac{a}{b}\right) + \pi$
-	+	$\arctan\left(\frac{a}{b}\right)$
-	-	$\arctan\left(\frac{a}{b}\right) - \pi$

Tabulka. 2.1 Rozšíření rozsahu hodnot fáze

Výše již bylo uvedeno, že informaci o výškovém rozložení profilu předmětu nese rozdíl fáze referenční roviny a fáze předmětu. Ovšem jak fázová matice referenční roviny tak fázová matice předmětu jsou zabalené. Bylo by nevýhodné rozbalovat každou matici zvlášť, a až poté tyto matice odečíst za účelem získat matici fáze předmětu v přirozeném tvaru. Výhodnější je fáze odečíst ještě před aplikováním rozbalovacího algoritmu. Označme fázi předmětu  $\phi_O(x, y)$  a fázi referenční roviny  $\phi_R(x, y)$ . Pak pro jejich rozdíl obdržíme [2]:

$$\phi(x, y) = \phi_O(x, y) - \phi_R(x, y) = \arctan\left(\frac{a_m b_r - b_m a_r}{b_m b_r + a_m a_r}\right), \quad (2.5)$$

kde:

$$a_m = \sum_{i=1}^N I_i^m(x, y) \sin(2\pi i/N)$$

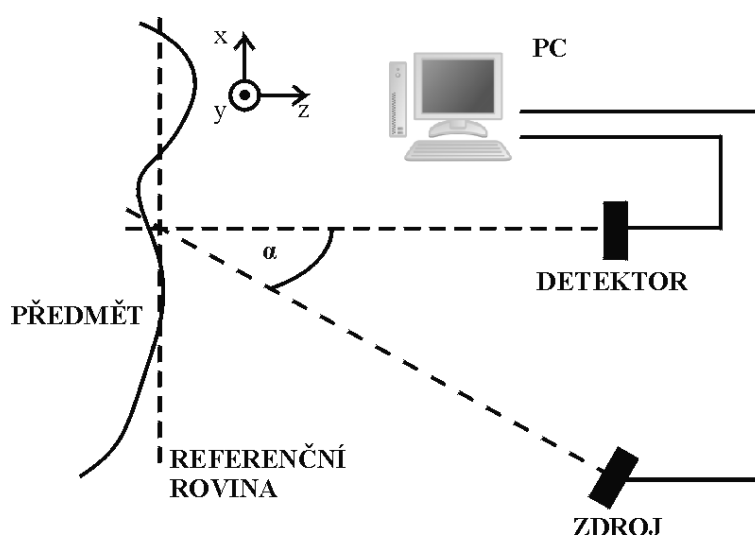
$$a_r = \sum_{i=1}^N I_i^r(x, y) \sin(2\pi i/N)$$

$$b_m = \sum_{i=1}^N I_i^m(x, y) \cos(2\pi i/N)$$

$$b_r = \sum_{i=1}^N I_i^r(x, y) \cos(2\pi i/N).$$

$I_i^m(x, y)$  označuje jednotlivé snímky objektu,  $I_i^r(x, y)$  jednotlivé snímky referenční roviny a  $N$  označuje celkový počet snímků.

Měřicí proces phase-shifting profilometrie tedy spočívá v získání jednotlivých snímků objektu  $I_i^m(x, y)$  a referenční roviny  $I_i^r(x, y)$  při promítnuté sinusové mřížce se správnou fází dle vztahu (2.2). Fázi vypočítanou ze vztahu (2.5) nesoucí informaci o profilu objektu je dále třeba rozbalit vhodným algoritmem. Při použití maticového detektoru pro měření jednotlivých snímků je fáze vyjádřena dvourozměrnou maticí. Pro vyjádření profilu v jednotkách délky je nutné provést změnu měřítka jednotlivých os vzhledem ke kalibraci sestavy. Tento problém bude podrobněji popsán současně s popisem automatizace. Typické schéma měřicí sestavy pro phase-shifting profilometrii je na obr. 2.6, kde zdroj promítá pod úhlem vzhledem k ose detektoru sinusové mřížky na předmět či referenční rovinu. Předmětem či rovinou zdeformovaná mřížka je snímána detektorem. Snímky pořízené detektorem jsou pak zpracovány a vyhodnoceny počítačem.



Obr. 2.6 Schéma měřicí sestavy pro phase-shifting profilometrii

### **3. Programové prostředí MATLAB<sup>®</sup>**

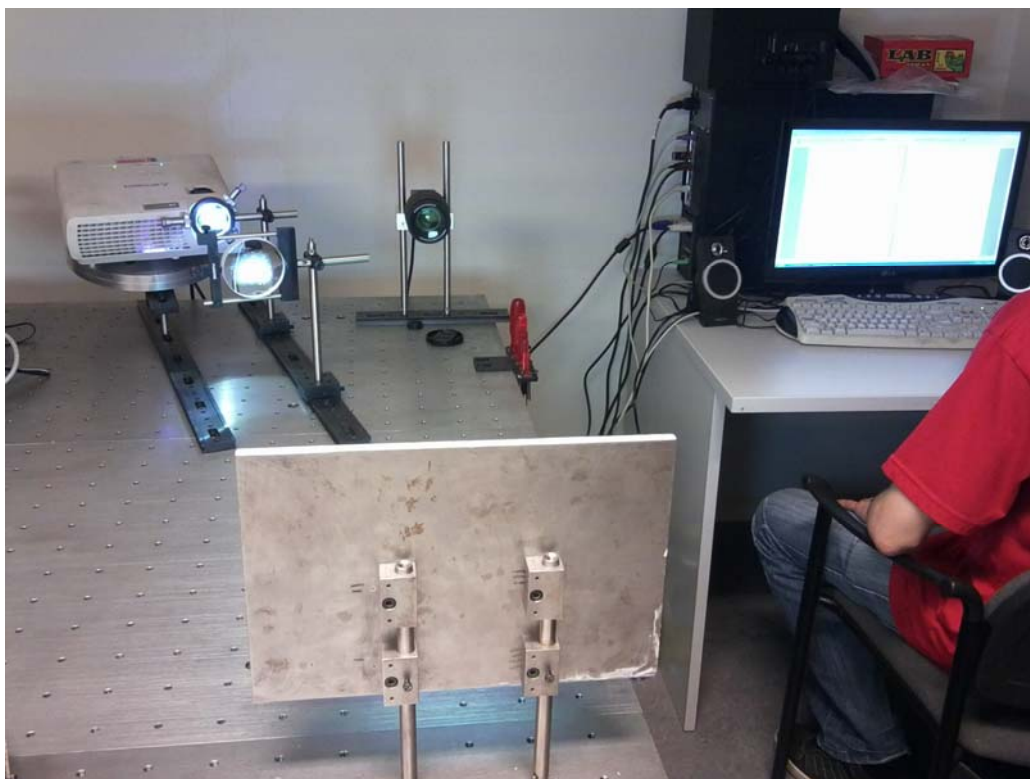
Pro vývoj algoritmů pro automatizaci měření byl zvolen program MATLAB<sup>®</sup>. Jedná se o programové prostředí určené pro nejrůznější matematické výpočty, zpracování a prezentace dat atd. Program je také možno rozšířit o řadu doplňků např. pro zpracování dat z externí kamery (Image Acquisition Toolbox), které tak rozšiřují oblast použití. Součástí programu je nepřeberné množství matematických funkcí, jako jsou např. diskrétní Fourierovy transformace, otáčení matic o definovaný úhel atd. Uživatel se tedy může soustředit přímo na vývoj složitějších algoritmů, a neztrácet čas psaním a odlaďováním těchto ne vždy rutinních, ale již lety známých funkcí. Prostředí samozřejmě dovoluje vytváření nových funkcí. Jazyk programového prostředí obsahuje klasické programovací prvky typické pro klasické programovací jazyky (rozhodovací struktury, cykly atd.). Syntax tohoto jazyka vychází z programovacího jazyka Fortran, který byl v minulosti navržen firmou IBM pro vědecké aplikace.

Programové prostředí MATLAB<sup>®</sup> dále také umožňuje vytváření tzv. GUI (Graphical User Interface) aplikací, do češtiny často překládáno jako grafické uživatelské rozhraní. Tato aplikace sestává z uživatelem naprogramovaných funkcí, které jsou navázány na různé prvky (tlačítka, textová pole, obrazová pole atd.) klasického formuláře systému Windows. Spouštění těchto funkcí je pak generováno událostmi jednotlivých prvků formuláře. Cílem této práce je vyvinout jednotlivé algoritmy pro automatizaci měření při PSP, a tyto implementovat do výše zmiňovaného grafického uživatelského prostředí.

#### **3.1 Experimentální sestava pro PSP a její ovládání**

Pro vývoj a testování algoritmů byla realizována experimentální sestava pro měření 3D profilu metodou phase-shifting profilometrie. Realizovaná sestava se skládá z dataprojektoru, kamery, odnímatelné referenční roviny a počítače. Fotka sestavy postavené na optickém stole je na obr. 3.1. Dataprojektor je typu DLP od firmy Mitsubishi, a to s rozlišením 1024 x 768 pixelů. Kamera je produktem firmy AVT, typ MARLIN F-131B. Jedná se o černobílou CMOS kameru s rozlišením 1280 x 1024 pixelů. K počítači je připojitelná přes FireWire rozhraní. Dále také disponuje grafickou kartou se dvěma VGA výstupy, aby bylo možné k počítači připojit dataprojektor za současného připojení standardního monitoru pro zobrazování uživatelského řídicího programu. Činnost kamery a

dataprojektoru je ovládána z programového prostředí MATLAB. Základem téměř všech v práci uváděných algoritmů je nastavování promítaného obrazce dataprojektorem a získávání snímků z kamery. Proto budou v následujících podkapitolách tyto činnosti stručně popsány.



*Obr. 3.1 Experimentální sestava pro PSP postavená na optickém stole*

### **3.1.1 Ovládání dataprojektoru z prostředí MATLAB**

Použitý dataprojektor je připojen k VGA výstupu grafické karty a v systému Windows je nastaven jako rozšířená plocha původního monitoru. Pro promítání uživatelem definovaného obrazce na dataprojektor pomocí prostředí MATLAB je nejprve nutné tento obrazec nadefinovat. Obrazec je reprezentován dvojrozměrnou maticí, kde hodnota prvku matice udává intenzitu daného bodu obrazce. Použitý dataprojektor je schopen zobrazit 256 úrovní šedé, je tedy vhodné generované obrazce vytvářet také v tomto rozsahu a to od 0 do 255, kdy vyšší hodnota odpovídá vyšší intenzitě. Také rozlišení obrazce by mělo být totožné s rozlišením dataprojektoru tzn. 1024 x 768.

Pro zobrazení obrazce slouží v prostředí MATLAB příkaz `imshow(A, [0, 255])`, kde `A` značí matici obrazce a čísla v hranatých závorkách určujeme rozsah hodnot obrazce. Pokud se v matici vyskytuje číslo mimo tento rozsah, tak je nahrazeno bližší mezní hodnotou tohoto rozsahu. Pokud je definice rozsahu při použití příkazu vynechána, je použit rozsah



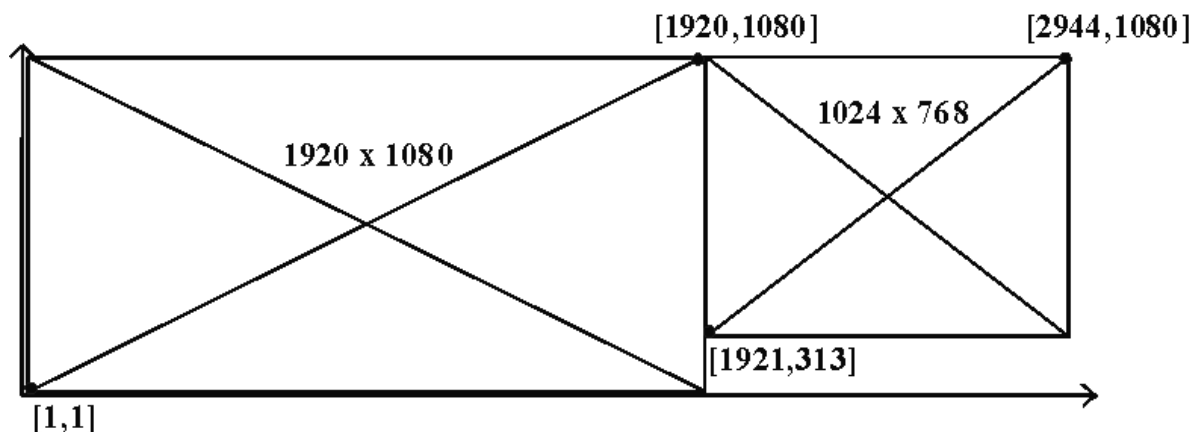
[0,1]. Po použití výše uvedeného příkazu se objeví nové okno, ve kterém bude zobrazen obrazec určený maticí A. Pro zobrazení tohoto obrazce na dataprojektor je nutné provést sérii příkazů, které nastaví parametry nového okna tak, aby přesně pasovalo na rozlišení dataprojektoru. Celkově pak kód pro zobrazení obrazce definovaného maticí A na dataprojektor vypadá následovně:

```
1: imshow(A, [0, 255]);
2: set(gca, 'Position', [0 0 1 1]);
3: set(gcf, 'MenuBar', 'none', 'NumberTitle', 'off');
4: set(gcf, 'Position', [1921 313 1024 768]);
```

Příkaz na druhém řádku způsobí, že okraje obrazce budou totožné s okraji okna (standardně je mezi okrajem okna a okrajem obrazce určitá mezera). Příkaz na třetím řádku z okna odstraní veškeré prvky související pouze s nastavením námi otevřeného okna. Příkaz na posledním řádku již jen nastaví polohu a velikost zobrazované oblasti tak, aby vyplnila plochu zobrazovanou dataprojektorem. Hodnoty v hranatých jsou závislé na konkrétním rozlišení použitého monitoru a dataprojektoru. Pro tento případ, kdy má monitor rozlišení 1920 x 1080 pixelů a dataprojektor rozlišení 1024 x 768 pixelů, se k daným hodnotám dojde následovně:

- první hodnota udává horizontální souřadnici prvního pixelu plochy, který již náleží dataprojektoru, tzn.  $1920 + 1 = 1921$ , kdy horizontální souřadnici pixelů číslujeme zleva doprava
- druhá hodnota udává vertikální souřadnici prvního pixelu plochy, který již náleží dataprojektoru, tzn.  $1080 - 768 + 1 = 313$ , kdy vertikální souřadnici pixelů číslujeme zdola nahoru
- poslední dvě hodnoty definují šířku a výšku obrazce a jsou tedy totožné s rozlišením dataprojektoru

Pro lepší představu o číslování pixelů v prostředí MATLAB a pozicích jednotlivých ploch monitoru a projektoru je na obr. 3.2 zobrazeno umístění jednotlivých ploch i se souřadnicovou soustavou.



Obr. 3.2 Rozmístění a soustava souřadnic ploch monitoru a dataprojektoru

Uvedený způsob zobrazování obrázků na dataprojektor v prostředí MATLAB jistě není ten nejelegantnější, ale ukázal se jako funkční. Vývoj grafického objektu, který by se mnohem komplexněji staral o obraz dataprojektoru, by byl časově velmi náročný a ve výsledku by nijak dále vyvíjené algoritmy nezjednodušil či nezměnil. Proto je u všech algoritmů v této práci použito pro promítání obrázků na dataprojektor právě výše zmíněného bloku příkazů.

### 3.1.2 Ovládání kamery z prostředí MATLAB

Pro získávání snímků či videosekvencí z externích zařízení je prostředí MATLAB vybaveno doplňkem Image Acquisition Toolbox. Ten obsahuje mnoho funkcí pro ovládání a nastavování připojené kamery. Před prvním použitím kamery v prostředí MATLAB je nutné nainstalovat požadovaný ovladač pro komunikaci s kamerou. Speciálně pro FireWire kamery byl schválen standard DCAM, který sjednocuje způsob komunikace kamery s počítačem. Většina FireWire kamer tento standard splňuje a je tedy možné s kamerou komunikovat pomocí ovladače DCAM. Použitá kamera MARLIN F-131B také splňuje tento standard. Po úspěšném nainstalování ovladače je již možno pracovat s kamerou v prostředí MATLAB.

Práce s kamerou v prostředí MATLAB začíná vždy vytvořením video-objektu. To je provedeno pomocí příkazu `videoinput('ovladač')`, kde proměnná `ovladač` specifikuje, pomocí kterého ovladače se bude s kamerou komunikovat. Pro případ FireWire kamery to bude ovladač DCAM. Kód pro inicializaci video-objektu bude vypadat následovně:

```
video_object = videoinput('dcam');
```

Po úspěšném vytvoření objektu je již možné zachytávat snímky, či nastavovat různé parametry kamery. Nejčastěji je kamera použita pro získání snímku. Pro tento účel slouží příkaz `getsnapshot(video_object)`. Tento příkaz vytvoří matici s rozměrem odpovídajícím rozlišení kamery, kde hodnoty matice odpovídají intenzitě jednotlivých pixelů při zachycení snímku. Kód pro uložení snímku z kamery do proměnné A bude vypadat takto:

```
A = getsnapshot(video_object);
```

Pro nastavování parametrů kamery jako je např. délka závěrky či jas, je nutné vytvořit objekt typu *videosource*, který se vytváří pomocí příkazu `getselectedsource(video_object)`. Oproti objektu typu *videoinput* slouží tento objekt ke správě specifitějších vlastností a nastavení, které nejsou společné pro různé kamery. Po vytvoření objektu je možné měnit parametry pomocí tzv. *object-inspektoru*, který je vyvolán příkazem `inspect(video_source)`. Tento příkaz otevře dialogové okno, kde uživatel vidí aktuální nastavení daného záznamového zařízení a dále může toto nastavení měnit. Pro přístup k parametrům pomocí kódu se využívá klasická objektová konstrukce `objekt.parametr`. V následujícím kódu je ukázáno nastavení závěrky kamery na manuální mód včetně vytváření objektů:

```
video_object = videoinput('dcam');  
video_source = getselectedsource(video_object);  
video_source.ShutterMode = 'manual';  
video_source.Shutter = 3000;
```

Mezi nejdůležitější parametry kamery patří:

- *ShutterMode* – udává, jestli bude závěrka automatická nebo pevně dána hodnotou parametru *Shutter*
- *Shutter* – číslo od 1 do 4095, které udává relativní pevně definovaný čas závěrky, platné pouze pokud je *ShutterMode* nastaven na 'manual'
- *Brightness* – číslo od 0 do 127 udávající jas obrazu

Výše uvedené postupy pro inicializaci, nastavení a pořizování snímků kamery jsou plně dostačující pro veškeré algoritmy uvedené v této práci. Po ukončení práce s kamerou je vhodné objekty související s kamerou vymazat. Vymazání objektů `video_object` a `video_source` je uvedeno v následujícím kódu:

```
delete(video_source);  
clear(video_source);  
delete(video_object);  
clear(video_object);
```

## 4. Vývoj algoritmů pro automatizaci měření

Tato kapitola tvoří hlavní část diplomové práce. Budou zde popsány jednotlivé úkony spojené s měřením a nastavením experimentální sestavy pro PSP. Jednotlivé činnosti, které budou předmětem automatizace, budou v odpovídajících podkapitolách nejprve stručně popsány. Dále budou zmíněny různé možnosti řešení automatizace. Pro zvolenou možnost pak bude popsán vytvořený algoritmus. Automatizace se bude týkat následujících činností:

- korekce přenosové funkce dataprojektoru
- kalibrace příčných os (osy x a y)
- proces měření metodou PSP
- kalibrace podélné osy (osy z)

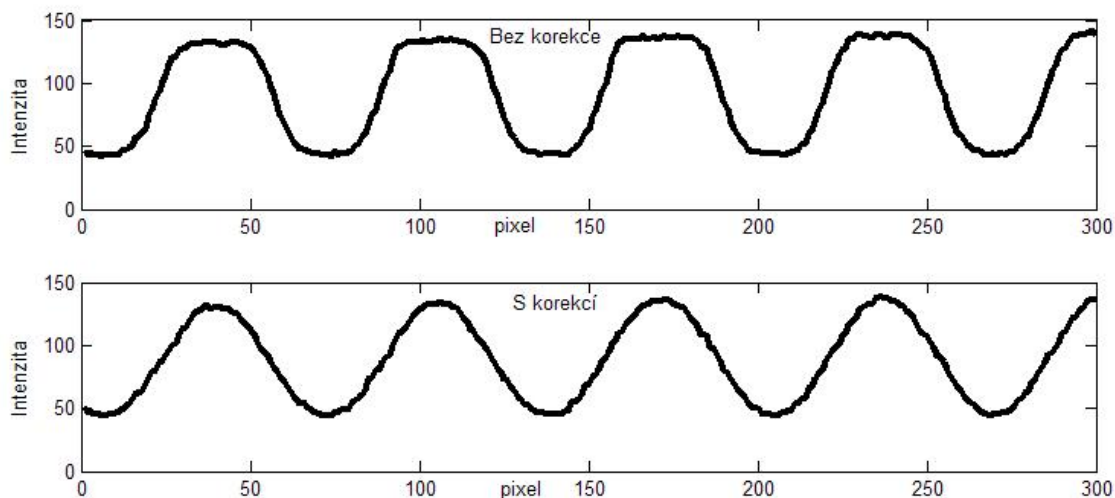
### 4.1 Korekce přenosové funkce dataprojektoru

V realizaci experimentální sestavy pro PSP je pro promítání sinusových mřížek použit dataprojektor připojený ke grafické kartě počítače. Dataprojektor byl zvolen především vzhledem k jednoduchému ovládání a velké variabilitě promítaných obrazců. Z teorie PSP plyne, že promítané mřížky by měly být ideálně sinusové, jinými slovy fáze promítané mřížky by měla růst lineárně. Nelineární změny fáze totiž tato metoda vnímá jako odchylky od rovinnosti a tedy po promítnutí neideální mřížky na rovný předmět by tato metoda podala nerelevantní výsledek.

Již z principu dataprojektoru nemohou být promítané mřížky ideálně sinusové. Obraz promítaný dataprojektorem je totiž složen z konečného počtu pixelů. Promítaný sinusový průběh je tedy pouze diskrétní aproximací spojitého sinusového průběhu. Ukazuje se ovšem, že při vhodně zvolené periodě sinusové mřížky vzhledem k rozlišení dataprojektoru nemá tato diskretizace zásadní vliv na přesnost metody.

Mnohem větší vliv na přesnost má nelineárnost přenosové funkce dataprojektoru. Přenosovou funkcí dataprojektoru rozumíme vztah mezi vstupní hodnotou intenzity definovanou prostřednictvím čísla v rozsahu od 0 do 255 a výstupní intenzitou bodu promítnutého dataprojektorem na stínítko. S ohledem na metodu PSP se nelineárnost přenosové funkce projevuje značným zkreslením promítaných sinusových mřížek. Na obr. 4.1 jsou vykresleny řádkové průběhy snímku promítnuté sinusové mřížky bez korekce a s korekcí přenosové funkce dataprojektoru. Ve výsledku se nelineárnost dataprojektoru a tedy

zkreslenost promítané sinusové mřížky projeví zvlněním naměřeného profilu. Tato skutečnost značně omezuje přesnost metody.



Obr. 4.1 Řádkové průběhy snímku promítnuté sinusové mřížky bez korekce a s korekcí přenosové funkce dataprojektoru

Možností, jak potlačit nelineárnost přenosové funkce dataprojektoru je cíleně zkreslit vygenerovaný obraz posílaný na dataprojektor tak, aby se po promítnutí toto zkreslení vyrušilo se zkreslením dataprojektoru. Pro zjištění způsobu, jak zkreslovat generované obrazy se sinusovou mřížkou uvažujme promítání sinusové mřížky:

$$I_{pix} = \frac{255}{2} \left[ \sin\left(\frac{2\pi \cdot x}{\lambda}\right) + 1 \right], \quad (4.1)$$

kde  $I_{pix}$  značí číselnou hodnotu intenzity generovaného obrazu a  $\lambda$  je perioda sinusové mřížky v pixelech. Po promítnutí dataprojektorem máme na výstupu intenzitu:

$$I = I_0 f(I_{pix}), \quad (4.2)$$

kde  $f(x)$  označuje přenosovou funkci dataprojektoru a  $I_0$  označuje konstantu udávající vztah mezi číselnou reprezentací intenzity generovaného obrazu a skutečnou intenzitou po promítnutí. Jelikož je funkce  $f(x)$  obecně nelineární, neobdržíme po promítnutí požadovanou sinusovou mřížku. Pokud ovšem před promítnutím zkreslíme sinusový průběh

jinou nelineární funkcí  $g(x)$  je možné na výstupu obdržet sinusový průběh. Z rovnice (4.3) je patrné, že daná situace nastane, pokud  $g(x)$  je inverzní k funkci  $f(x)$ :

$$I = I_0 I_{pix} = I_0 f(g(I_{pix})) \Rightarrow g(x) = f^{-1}(x), \quad (4.3)$$

kde  $f^{-1}(x)$  označuje inverzní funkci k funkci  $f(x)$ . Platnost rovnice (4.3) plyne s vlastnosti inverzní funkce:

$$f^{-1}(f(x)) = x, \forall x \in D(f).$$

Korekce přenosové funkce tedy spočívá v určení funkce  $g(x) = f^{-1}(x)$ . Při promítání sinusových mřížek jsou pak obrazy ještě před promítnutím cíleně zkresleny přepočítáním všech hodnot matice obrazce pomocí funkce  $g(x)$ .

#### 4.1.1 Algoritmus korekce přenosové funkce dataprojektoru

Cílem algoritmu je automatizovaně zjistit korekční funkci  $g(x)$ , která bude při měření sloužit k cílenému zkreslování promítaných sinusových mřížek. Sinusové mřížky jsou generovány jako matice s velikostí odpovídající rozlišení dataprojektoru. Hodnoty jednotlivých bodů matice (v rozsahu od 0 do 255) pak udávají jas těchto bodů po promítnutí dataprojektorem, kdy větší hodnota odpovídá většímu jasu. Vzhledem k použití jsou po funkci  $g(x)$  požadovány následující vlastnosti:

- funkce prostá s definičním oborem zahrnující interval  $\langle 0, 255 \rangle$
- obor hodnot funkce pro vstupní hodnoty z intervalu  $\langle 0, 255 \rangle$  je podinterval intervalu  $\langle 0, 255 \rangle$
- inverzní průběh vzhledem k přenosové funkci dataprojektoru

První dvě vlastnosti plynou s faktu, že funkce je použita k přepočítávání hodnot generovaných sinusových mřížek. Vstupem je tedy vždy číslo z intervalu  $\langle 0, 255 \rangle$ . Výstupem pak zase musí být číslo ze stejného intervalu. Výstupní interval již nemusí být celý interval  $\langle 0, 255 \rangle$  a to z toho důvodu, že některé části průběhu přenosové funkce dataprojektoru nemusí být v obecném případě prosté. Hodnoty spadající do těchto částí nelze při generování mřížek použít a jsou proto z intervalu  $\langle 0, 255 \rangle$  vypuštěny. Třetí požadavek zaručí, že cílené zkreslení před promítnutím se vyruší se zkreslením od dataprojektoru.

Algoritmus korekce přenosové funkce dataprojektoru lze rozdělit do následujících částí:

- naměření přenosové funkce dataprojektoru
- určení největšího podintervalu, na kterém je přenosová funkce prostá
- lineární změna měřítka osy  $y$  tak, aby minimální vstupní hodnotě podintervalu odpovídala hodnota 0 a maximální hodnotě podintervalu hodnota 255.
- aproximace inverzního průběhu přenosové funkce polynomem

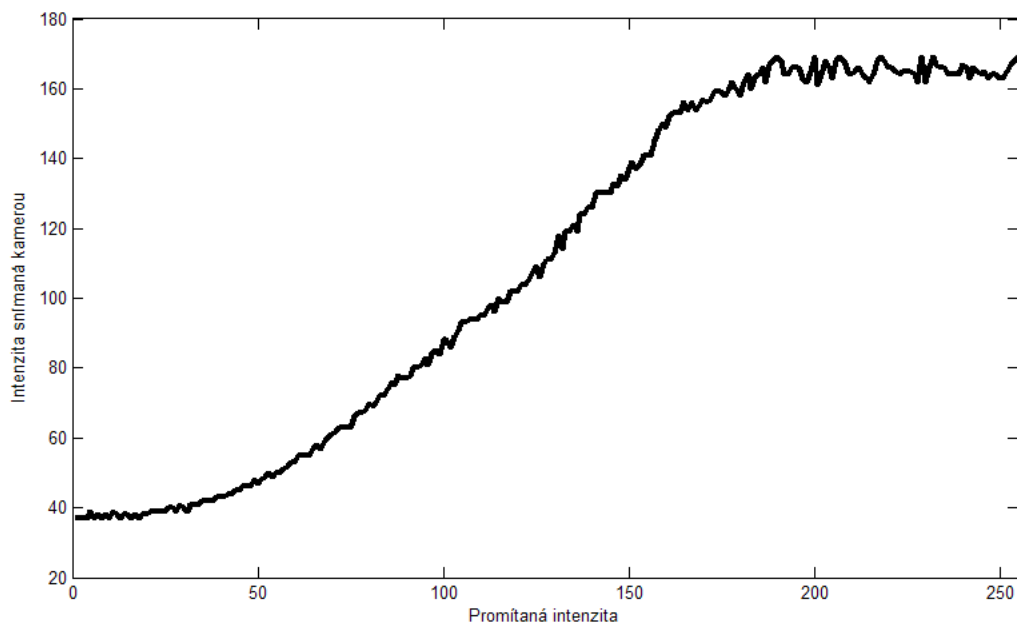
Jednotlivé části kódu jsou uvedeny v jazyku programového prostředí MATLAB. Výstupem algoritmu je v tomto případě objekt typu *cf<sub>fit</sub>* – polynomiální aproximace inverzního průběhu.

V následujícím kódu je část algoritmu, jejímž výstupem je naměřený průběh přenosové funkce dataprojektoru:

```
1: for i = 0:1:255
2:     intensity(:, :) = i;
3:     set(im, 'CData', intensity);
4:     pause(0.4);
5:     B = getsnapshot(video_object);
6:     data(i+1) = double(B(500, 650));
7: end
```

Měření probíhá postupným promítáním všech možných hodnot intenzity dataprojektorem na referenční rovinu. Pro každou hodnotu je proveden snímek referenční roviny, ze kterého je vzata hodnota intenzity bodu přibližně ve středu snímaného pole. Výsledkem je vektor *data* s 256 prvky. Programově je měření řešeno cyklem (viz kód výše) s počtem opakování rovným počtu možných intenzit promítaných dataprojektorem. Nastavení promítané intenzity je prováděno příkazy na řádcích 2 a 3. Příkazem na řádku 5 je proveden snímek. Na řádku 6 je pak do vektoru *data* uložena příslušná hodnota intenzity reprezentujícího bodu snímku. Na obr. 4.2 je vykreslen typický průběh naměřené přenosové funkce dataprojektoru.





Obr. 4.2 Typický průběh naměřené přenosové funkce dataprojektoru

Pro určení největšího prostého podintervalu je nejprve provedena pracovní aproximace průběhu přenosové funkce polynomem. Studováním diskrétní derivace polynomu je pak určen hledaný podinterval. Pod pojmem diskrétní derivace rozumíme rozdíl:

$$\delta(i) = f(i+1) - f(i), \quad (4.4)$$

kde  $f(x)$  je studovaná funkce a  $i$  je celé číslo. Začátek podintervalu je určen první hodnotou  $i$ , pro kterou je diskrétní derivace větší jako určitá mezní hodnota  $c_1$ , s tím že diskrétní derivaci určujeme postupně od  $k=0$ . Konec podintervalu je pak podobně určen první následující hodnotou  $k$ , pro kterou diskrétní derivace klesne pod určitou mezní hodnotu  $c_2$ . Je intuitivní, že pro mezní hodnoty  $c_1$  a  $c_2$  by měl platit vztah  $c_1 > c_2$ . V opačném případě bude téměř vždy druhá podmínka splněna již pro následující hodnotu  $i$ . Vzhledem k tomu, že průběh přenosové funkce dataprojektoru má vždy podobný průběh, je vhodnou volbou mezních hodnot  $c_1$  a  $c_2$  zaručeno spolehlivé určení hledaného podintervalu. Diskrétní derivace není počítána přímo z naměřených dat, jelikož tyto data jsou vždy jistou měrou zatížena šumem a i malý šum má při výpočtu derivace negativní dopad. Aproximací dat polynomem je původně naměřený průběh vyhlazen. Je tak do určení mezních hodnot podintervalu vnášena jistá chyba, ovšem přesnost určení tohoto podintervalu není nijak kritická.

Algoritmus určení podintervalu je uveden v následujícím kódu:

```

1.  y = 0:1:255;
2.  k = fit(y',data','poly9');
3.  for i = 0:1:254
4.      if (k(i+1) - k(i)) > c1
5.          a = i;
6.          break;
7.      end
8.  end
9.
10. for i = (a+1):1:254
11.     if (k(i+1) - k(i)) < c2
12.         b = i;
13.         break;
14.     end
15. end

```

Příkazy na první a druhém řádku provedou aproximaci původních dat polynomem. K aproximaci je použita funkce `fit` z Curve Fitting Toolboxu programového prostředí MATLAB. Poté je blok kódu složen ze dvou cyklů s definovaným počtem opakování, kde první vyhledává začátek podintervalu počítáním diskrétní derivace z funkce  $k$  pro hodnoty indexu  $i$  od 0 do 254. Hodnota je nalezena, pokud je diskrétní derivace větší než mezní hodnota  $c_1$ . Nalezením počátku podintervalu je první cyklus přerušen a následuje hledání konce podintervalu podobným způsobem. Jako konec podintervalu je brána hodnota indexu  $i$ , pro kterou klesne diskrétní derivace pod mezní hodnotu  $c_2$ . Tato derivace je již počítána pouze pro hodnoty indexu  $i$  od počátku podintervalu do 254.

Poslední částí algoritmu korekce přenosové funkce dataprojektoru je lineární změna měřítka osy  $y$  a následná aproximace podintervalu naměřených dat polynomem. Lineární změna měřítka osy  $y$  je prováděna proto, aby finální aproximací byla získána funkce, která zobrazuje interval  $\langle 0,255 \rangle$  na podinterval stejného intervalu. Výslednou funkcí je však funkce inverzní k původnímu průběhu, a proto pokud po inverzní funkci požadujeme definiční obor  $\langle 0,255 \rangle$  pak původní funkce musí mít takový obor hodnot. Algoritmus změny měřítka a finální aproximace inverzního průběhu je uveden v následujícím kódu:

```

1: usek = data(a+1,b+1);
2: usek = ((usek - usek(1))./(usek(b-c+1) - usek(1))).*255;
3: x = a:b;
4: korekce = fit(usek',x','poly9');

```

První řádek kódu uloží do vektoru `usek` pouze podinterval všech naměřených hodnot. Na druhém řádku je provedena změna měřítka osy `y` přepočítáním všech hodnot vektoru `usek`. Na třetím řádku je vytvořen vektor `x`, který definuje `x`-ové hodnoty. Na posledním řádku je provedena aproximace polynomem. Příkaz `fit` má jako první parametr vektor `x`-ových hodnot a jako druhý parametr vektor `y`-ových hodnot. Dosazením vektoru `y`-ových hodnot za první parametr a `x`-ových hodnot za druhý parametr obdržíme aproximaci inverzního průběhu. Výstupem celého algoritmu korekce přenosové funkce dataprojektoru je proměnná `korekce`, ve které je uložen objekt typu `cfit` obsahující polynomiální aproximaci. Pro přepočítání vygenerovaných sinusových mřížek pomocí funkce pro korekci je použita intuitivní konstrukce:

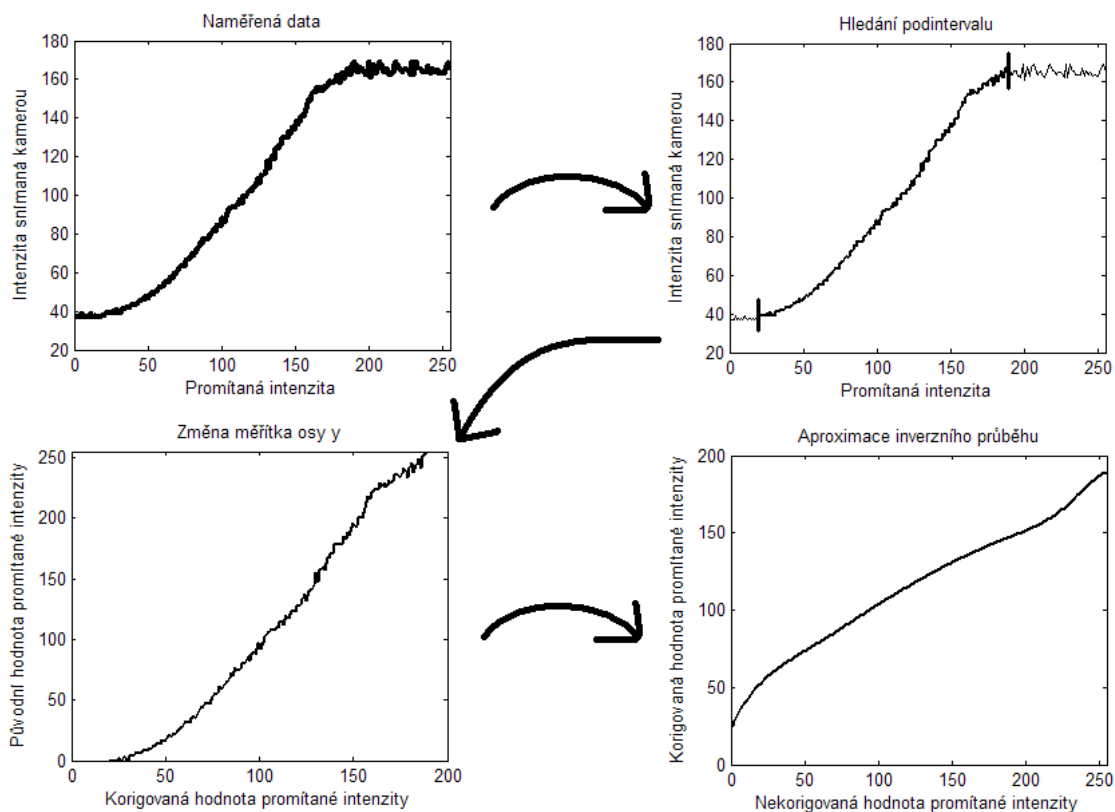
```

korigovana_sin_mrizka = korekce(puvodni_sin_mrizka);

```

kde `korigovana_sin_mrizka` je matice cíleně zkreslené sinusové mřížky, `puvodni_sin_mrizka` je matice původně vygenerované sinusové mřížky dle vztahu (4.1) a `korekce` je přepočítávací funkce určená algoritmem pro korekci přenosové funkce dataprojektoru.

Na obr. 4.3 je blokově ilustrován celý postup algoritmu korekce přenosové funkce dataprojektoru. Uvedený algoritmus představuje jednoduchý postup pro korekci nelineární přenosové funkce dataprojektoru. Vzhledem k metodě PSP je prováděna proto, aby se promítané sinusové mřížky co nejvíce blížily ideálnímu průběhu. Vliv korekce na kvalitu promítnuté sinusové mřížky je patrný z obr. 4.1. Převodní korekční funkci, která je výsledkem algoritmu lze však využít i v jiných aplikacích, kde je potřeba korigovat nelinearity přenosové funkce promítacího zařízení.



Obr. 4.3 Bloková ilustrace činnosti algoritmu korekce přenosové funkce dataprojektoru

## 4.2 Kalibrace příčných os (osy x a y)

Metoda PSP poskytuje výsledky výškového rozložení plochy povrchu předmětu ve formě matice fáze, která pro každý bod (pixel) snímacího prvku poskytuje tuto informaci. Pokud uvažujeme souřadnou soustavu podle obr. 2.6, pak při zobrazení výsledků máme na osách x a y jednotku pixel a na ose z jednotku fáze. Pro většinu aplikací je však potřeba získat výsledky v jednotkách délky. Pro tuto potřebu je nutné všechny osy kalibrovat. Tato kapitola se zabývá kalibrací příčných os, tedy os x a y. Stručně řečeno se jedná o určení vztahu mezi pixely snímacího prvku a skutečnými vzdálenostmi ve snímané scéně. Pokud tedy umístíme do snímané scény předmět, u kterého známe délku v pixelech, zajímá nás, jaké rozměry má tento předmět v jednotkách délky.

Většina způsobů kalibrace příčných os spočívá v určení velikosti předmětu v pixelech. Pokud známe skutečný rozměr tohoto předmětu, pak lze jednoduchým výpočtem určit konstantu, která bude délkové rozměry v pixelech převádět na rozměry v jednotkách délky. Vzhledem k požadavku na automatizaci je nutné zvolit způsob kalibrace, který bude přesný a

nebude vyžadovat velké nároky na obsluhu. V této kapitole budou uvedeny dva způsoby kalibrace:

- kalibrace pomocí sinusové mřížky
- kalibrace vyšetřováním obsahu obrazce

Oba způsoby kalibrace budou vyžadovat umístění kalibračního obrazce o známém délkovém parametru na referenční rovinu. Po získání snímku referenční roviny s kalibračním obrazcem, již bude pro každý způsob jiným algoritmem zjištěn rozměr v pixelech. Za předpokladu, že rozměrový parametr kalibračního obrazce v jednotkách délky je známý, lze jednoduchým výpočtem určit převodní konstantu:

$$K[\text{mm/pix}] = \frac{L_{\text{mm}}[\text{mm}]}{L_{\text{pix}}[\text{pix}]}, \quad (4.5)$$

kde  $K$  je převodní konstanta,  $L_{\text{mm}}$  je rozměr v jednotkách mm a  $L_{\text{pix}}$  je rozměr v jednotkách pixelů. Při znalosti konstanty  $K$  lze jakýkoliv rozměr v pixelech přepočítat na rozměr v jednotkách délky:

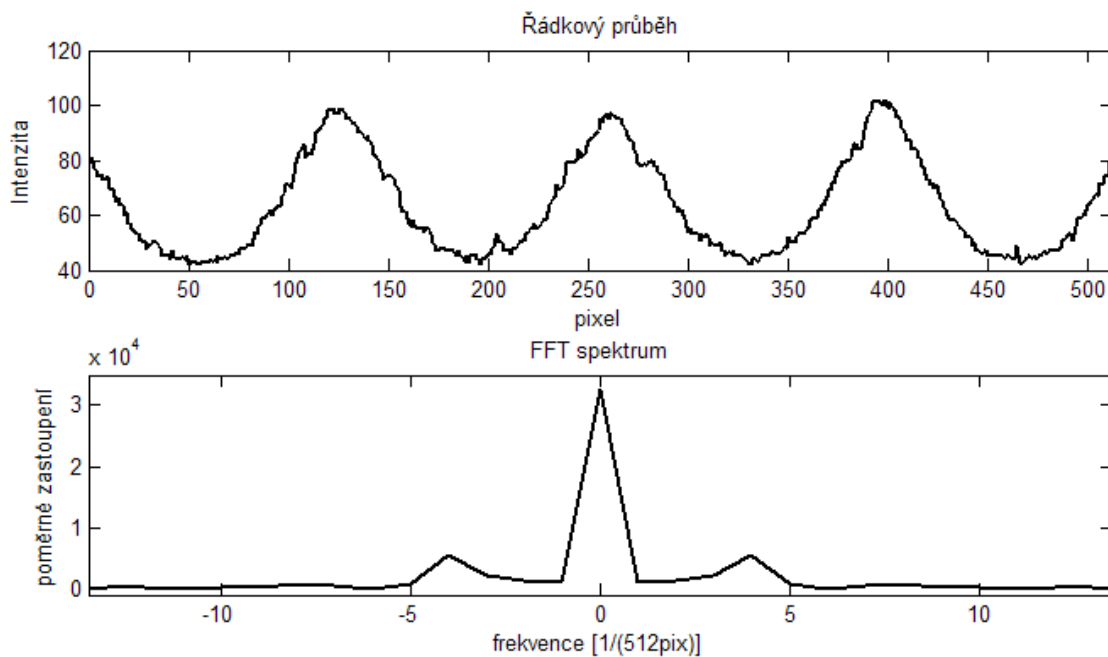
$$L_{\text{mm}} = K \cdot L_{\text{pix}} \quad (4.6)$$

### 4.2.1 Kalibrace pomocí sinusové mřížky

První uvedený způsob kalibrace je pomocí sinusové mřížky. Princip kalibrace spočívá v zjišťování periody sinusové mřížky pomocí dvojrozměrné Fourierovy transformace. Pro jednoduchost bude základní myšlenka vysvětlena na jednorozměrné Fourierově transformaci. Dále budou uvedeny důvody přechodu k dvourozměrné Fourierově transformaci a uveden konečný algoritmus kalibrace. Popis Fourierovy transformace je nad rámec této práce, základní informace o jednorozměrné i dvourozměrné Fourierově transformaci včetně jejich diskrétních adaptací lze nalézt v [3, 4, 5].

Cílem kalibračního algoritmu je určit převodní konstantu  $K$ , pomocí které bude možno dle vztahu (4.6) přepočítávat pixelové rozměry na rozměry v jednotkách délky. Toho je u tohoto způsobu kalibrace docíleno určením periody sinusové mřížky jak v jednotkách délky, tak v pixelech. Určení periody v jednotkách délky je po vytvoření kalibračního obrazce provedeno jednou, pomocí klasických délkových měřidel. Určení periody v pixelech je práce kalibračního algoritmu.

Princip zjištění periody sinusové mřížky v pixelech spočívá ve vyšetřování spektra vyfocené sinusové mřížky získaného diskretní verzí Fourierovy transformace. Důvodem, proč je jako kalibrační obrazec zvolena sinusová mřížka je ten, že spektrum sinusového průběhu má speciální vlastnosti a umožňuje jednoduché algoritmické určení požadovaných parametrů. Na obr. 4.4 je zobrazen průběh jednoho řádku vyfocené sinusové mřížky a také detail diskretního Fourierova spektra získaného pomocí algoritmu FFT (Fast Fourier Transform) [3]. Ve spektru z obr. 4.4 je kromě stejnosměrné složky (frekvence rovna nule) patrná dvojice peaků (slovem peak označujeme dominantní lokální maximum ve spektru) odpovídající sinusovému průběhu. Pozice těchto peaků nese informaci o periodě sinusového průběhu v pixelech. Z vlastností Fourierovy transformace plyne, že spektrum je symetrické. Hodnoty nalevo od nuly jsou tedy pouze zrcadlovým obrazem hodnot z kladné části. Při vyšetřování spektra se lze tedy omezit pouze na jednu část FFT spektra.



Obr. 4.4 Řádkový průběh snímku sinusové mřížky a jeho Fourierovo spektrum

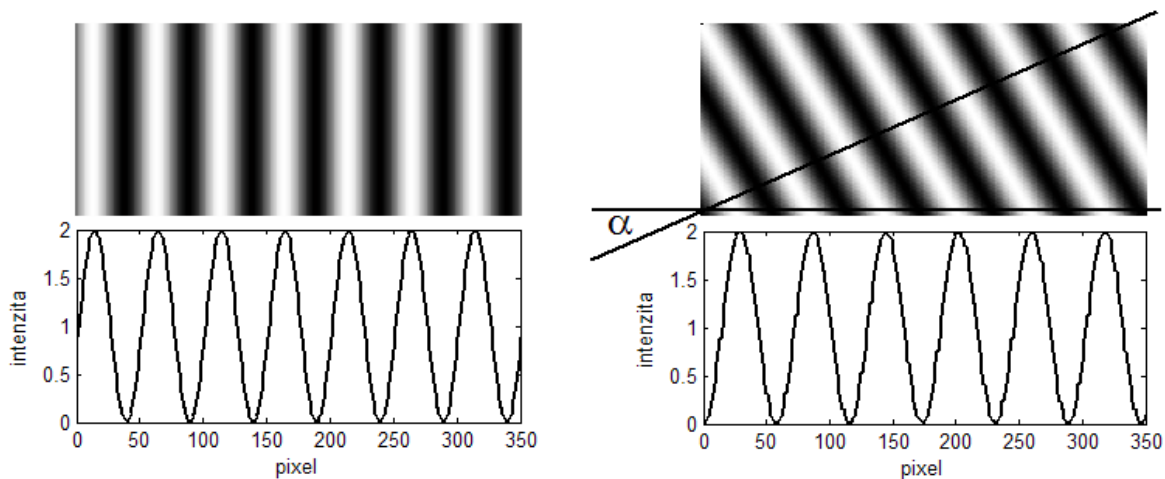
Jednou z vlastností jednorozměrné diskretní Fourierovy transformace je, že vstupem je vektor komplexních hodnot, jehož délka odpovídá celočíselným mocninám čísla 2. Výsledkem je opět vektor komplexních hodnot se stejnou délkou. Pro účely určování periody je možné se ve spektru zaměřit pouze na absolutní hodnoty komplexních čísel. V našem případě je vstupem vždy pouze signál z kamery, který je reálný, a proto lze komplexní charakter Fourierovy transformace ignorovat. Hodnoty ve spektru udávají poměrné zastoupení daných frekvencí v původním signálu. Klasická jednotka frekvence udává, kolik

period periodického signálu se vejde do jednotky osy x původního signálu. V tomto případě je jako jednotka osy x původního signálu brána velikost vektoru nesoucí vstupní hodnoty signálu. Periodu signálu v pixelech pro danou frekvenci lze tedy spočítat z následujícího vzorce:

$$p[\text{pix}] = \frac{l}{f}, \quad (4.7)$$

kde  $l$  je délka původního vektoru, ze kterého bylo FFT spektrum počítáno a  $f$  je celočíselná frekvence určená polohou ve výsledném vektoru diskrétního spektra. Určením frekvence, která odpovídá peakům sinusového průběhu lze tedy spočítat periodu sinusové mřížky a tím pádem pomocí vzorce (4.5) určit kalibrační konstantu pro osy x a y.

Pro jednorozměrný případ by uvedený způsob kalibrace vyžadoval přesné umístění kalibračního obrazce na referenční rovinu tak, aby směr určený sinusovým průběhem byl rovnoběžný se směrem řádků pixelů snímacího zařízení. V opačném případě by v řádku byl detekován sinusový průběh s delší periodou. Tento jev je patrný z obr. 4.5, kde je zobrazen případ rovnoběžnosti i nenulového vychýlení obrazce a vliv úhlu vychýlení na periodu snímaného průběhu.



Obr. 4.5 Vliv rotace kalibračního obrazce na periodu sinusového průběhu

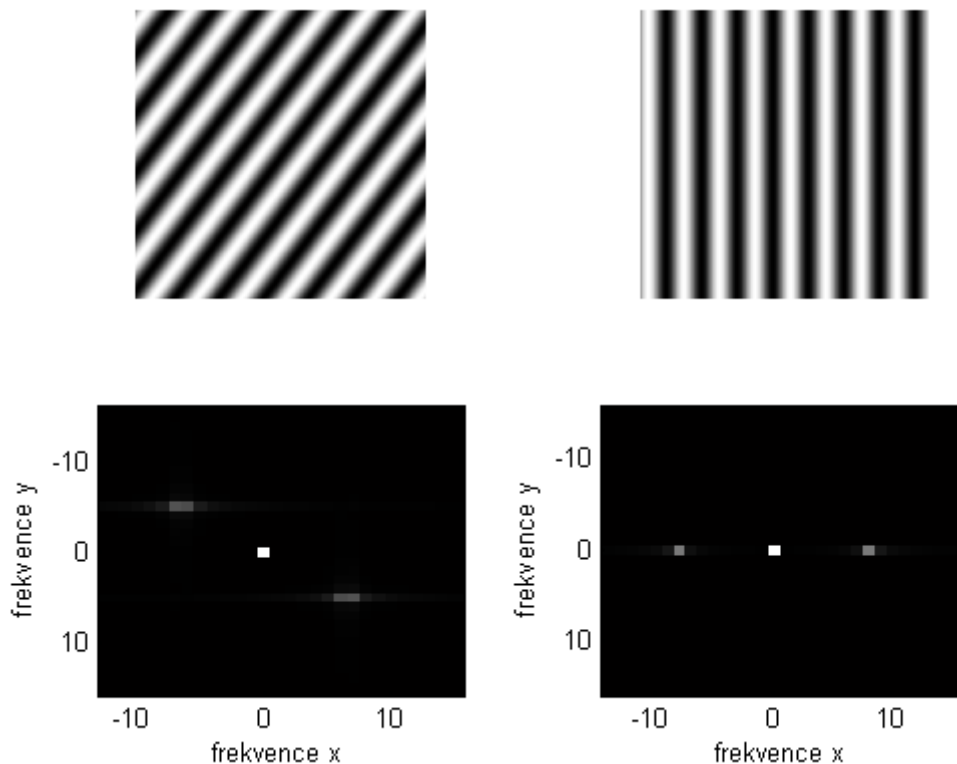
V závislosti na úhlu  $\alpha$  je v řádkovém průběhu vnímána perioda průběhu:

$$p = \frac{p_0}{\cos \alpha}, \quad (4.8)$$

kde  $p$  je perioda průběhu při natočení,  $p_0$  je perioda bez natočení a  $\alpha$  je úhel natočení. Nedokonalé umístění kalibračního obrazce by tedy vedlo ke zkreslené kalibrační konstantě a tím k negativnímu dopadu na celkovou přesnost zobrazovaných výsledků.

Přesné umístění kalibračního obrazce by bylo vzhledem k složitosti určení směru řádků kamery a kalibračního obrazce náročné. Navíc by tato operace požadovala vysoké nároky na obsluhu, kterým se ovšem automatizace měřícího procesu snaží předejít. Možným řešením je studování ne jednoho řádku, nýbrž dvojrozměrného výřezu z celkového snímku kalibračního obrazce se sinusovou mřížkou. Jednorozměrné spektrum je pak nahrazeno spektrem dvourozměrným, které je získáno pomocí dvourozměrné verze rychlého algoritmu diskrétní Fourierovy transformace (2D FFT).

2D Fourierovo spektrum sinusové mřížky má opět speciální vlastnosti. V celé matici spektra se vyskytuje dominantní peak na pozici  $[0,0]$ , který odpovídá stejnosměrné složce (nulové frekvenci). Dále se ve spektru opět objeví dvojice peaků odpovídající sinusové mřížce. Parametrem, který slouží k určení periody vyšetřované sinusové mřížky je vzdálenost peaků od středu spektra, tedy od bodu  $[0,0]$ . Pokud nedochází ke změně vzdálenosti kalibračního obrazce od kamery, pak se při jeho rotaci dvojice peaků pohybuje po kružnici se středem v počátku. Určená perioda je tedy invariantní vůči rotaci kalibračního obrazce. Na obr. 4.6 je uveden detail 2D Fourierova spektra pro dvě různé polohy sin. mřížky.



Obr. 4.6 2D diskrétní Fourierova spektra pro dvě různá natočení sin. mřížky



Algoritmus určení kalibrační konstanty tedy nejprve spočítá 2D diskrétní Fourierovo spektrum z výřezu snímku kalibračního obrazce se sin. mřížkou. Následně v tomto spektru vyhledá pozice peaků odpovídajících sinusovému průběhu. Vzhledem k tomu, že kalibrační obrazec je sinusová mřížka, je vyhledání peaků jednoduché. Ve spektru se totiž kromě peaku odpovídajícího stejnosměrné složce žádné jiné dominantní peaky nevyskytují. Jelikož je spektrum diskrétní, je také poloha peaku určena s konečnou přesností. Poloha je určena indexy prvku matice odpovídajícího maximu peaku. Tato přesnost je ovšem nedostačující. Pro zpřesnění je ovšem možno provést sub-pixelový odhad pozice peaku.

Sub-pixelový odhad pozice peaku je prováděn výpočtem těžiště z blízkého okolí maxima peaku. Přesnost sub-pixelového odhadu je již dostačující. Pro příklad výpočtu sub-pixelového odhadu pozice peaku uvažujme situaci na obr. 4.7, kde je v tabulce výřez matice spektra s maximem peaku na pozici [202 ; 231] s hodnotou 200. Poloha těžiště je spočítána pomocí vztahu:

$$i_{\text{sub}} = \frac{\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} i \cdot m(i, j)}{\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} m(i, j)}, \quad (4.9)$$

$$j_{\text{sub}} = \frac{\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} j \cdot m(i, j)}{\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} m(i, j)}, \quad (4.10)$$

kde  $m(i, j)$  je hodnota prvku matice s indexy  $i$  a  $j$ ,  $i_{\text{sub}}$  a  $j_{\text{sub}}$  určuje sub-pixelovou polohu peaku a hodnoty  $i_1, i_2, j_1, j_2$  určují blízké okolí peaku, ze kterého je těžiště počítáno. V praxi se ukazuje velikost blízkého okolí 3x3 jako dostačující. Pro případ na obr. 4.7 po dosazení do vzorců (4.9) a (4.10) vychází sub-pixelová poloha peaku na [202,02 ; 231,10].

201	2	11	4
202	31	200	61
203	3	15	5
	230	231	232

index j

Obr. 4.7 Ilustrace výpočtu odhadu sub-pixelové polohy peaku

#### 4.2.2 Algoritmus kalibrace pomocí sinusové mřížky

Cílem algoritmu kalibrace je určení kalibrační konstanty, která bude používána pro přepočítání jednotky pixel na jednotku délky. Samotný algoritmus určení kalibrační konstanty pomocí obrazce se sinusovou mřížkou je posloupností následujících činností:

- vyfocení kalibračního obrazce se sinusovou mřížkou umístěného na referenční rovině
- výpočet 2D diskretní verze (2D FFT) Fourierovy transformace z výřezu snímku o velikosti 512x512 pixelů
- nalezení pozice peaků odpovídajících sinusové mřížce ve 2D Fourierově spektru
- výpočet sub-pixelové pozice peaku určením těžiště blízkého okolí maxima peaku
- výpočet vzdálenosti sub-pixelové pozice peaku od středu spektra a následné určení periody vyfocené sin. mřížky v pixelech
- za předpokladu znalosti periody sin. mřížky v jednotkách délky výpočet kalibrační konstanty

V druhém kroku je brán pouze výřez snímku o velikosti 512x512, jelikož vstupem pro 2D diskretní Fourierovu transformaci je čtvercová matice o rozměru, který je mocninou čísla 2. Následující kód provádí první dvě části algoritmu a to vyfocení snímku a výpočet diskretního spektra z výřezu:

```

1: snimek = getsnapshot (video_object);
2: vyrez = snimek(mezy:mezy+511,mezx:mezx+511);
3: spektrum = abs(fftshift(fft2(vyrez)));

```

kde hodnoty parametry `mezx` a `mezy` udávají pozici výřezu. Samotný výpočet 2D FFT je volán příkazem `fft2` na třetím řádku výše uvedeného kódu. Příkaz `fftshift` způsobí přehození jednotlivých částí matice spektra tak aby střed spektra byl ve středu matice. Příkaz `abs` provádí pro každý prvek spektra výpočet absolutní hodnoty, hodnoty spektra jsou totiž obecně komplexní čísla a pro naše účely lze komplexní charakter zanedbat.

Jelikož jsou ve spektru dominantní pouze tři peaky, kdy poloha jednoho z nich (odpovídající stejnosměrné složce) je známa, je vyhledání pozice peaků relativně jednoduché. Dále z vlastnosti Fourierovy transformace plyne, že spektrum je středově symetrické vůči středu spektra. Stačí tedy nalézt polohu pouze jednoho peaku. Informace o poloze druhého je již redundantní a dá se určit z polohy toho prvního. Poloha peaku, který odpovídá sin. mřížce, je vyhledán jako maximum ve spektru, kde je cíleně potlačen peak ve středu spektra (ten je totiž vždy dominantní). Algoritmus hledání polohy peaku s pixelovou přesností je následující:

```

1: spektrum_pracovni = spektrum;
2: spektrum_pracovni(257,257) = 0;
3: [hodnota, radek, sloupec] = maximal(spektrum_pracovni);

```

Nejprve je na řádku č. 1 provedena pracovní kopie vypočítaného spektra. Dále je cíleně nastavena hodnota stejnosměrné složky spektra (pozice [257,257] v matici spektra) na nulu, aby bylo možné pozici hledaného peaku nalézt jako maximální hodnotu v matici. Funkcí `maximal` na třetím řádku je pak do proměnných `radek` a `sloupec` uložena pozice maxima peaku odpovídajícího sin. mřížce.

Další část algoritmu tvoří výpočet sub-pixelové pozice nalezeného peaku. Za předpokladu, že je známa poloha maxima peaku, spočívá pak určení sub-pixelové polohy peaku v implementaci vzorců (4.9) a (4.10). Zde je pro uvedena část kódu pro výpočet řádkové pozice sub-pixelové polohy:

```

1:  poloha_radek = 0;
2:  suma = 0;
3:  for i = radek-1:radek+1
4:      for j = sloupec-1:sloupec+1
5:          poloha_radek = poloha_radek + (i*d(i,j));
6:          suma = suma + d(i,j);
7:      end
8:  end
9:  poloha_radek = poloha_radek /suma;

```

Uvedený kód je pouze výpočet vzorce (4.9), kdy jako blízké okolí maxima peaku je brána matice s rozměrem 3 x 3 s maximem peaku ve středu této oblasti. V poslední části algoritmu je již jen spočítána vzdálenost sub-pixelové polohy peaku od středu spektra (pozice [257,257]), dále je pak spočítána perioda sinusové mřížky v pixelech a na závěr určena kalibrační konstanta:

```

1:  freq = sqrt((257-poloharadek)^2 + (257-poloha_sloupec)^2);
2:  perioda_pix = 512/freq;
3:  kalibracni_konstanta = perioda_mm/perioda_pix;

```

Na prvním řádku kódu je spočítána vzdálenost peaku od středu spektra pomocí Pythagorovy věty. Na druhém je určena perioda sinusové mřížky v pixelech a konečně na posledním řádku je spočítána kalibrační konstanta, která je výstupem celého algoritmu kalibrace příčných os. Po uvedení dalšího způsobu kalibrace příčných os bude provedeno srovnání obou přístupů. Budou také prezentovány výsledky zkušebního měření kalibračních konstant pro obě metody.

### 4.2.3 Kalibrace vyšetřováním obsahu obrazce

Následující způsob kalibrace je teoreticky podstatně jednodušší, než předchozí způsob. Je založen na vyšetřování obsahu obrazce v pixelech. Ze znalosti obsahu obrazce např. v mm<sup>2</sup> jsem schopni určit konstantu sloužící pro přepočet obsahu v pixelech na obsah v mm<sup>2</sup>. Tato konstanta je určena ze vztahu:

$$K_s = \frac{S[\text{mm}^2]}{S[\text{pix}^2]}. \quad (4.11)$$

Pro určení kalibrační konstanty pro přepočítání z pixelů na mm dospějeme pouhým odmocněním konstanty pro přepočítání obsahu:

$$K = \sqrt{K_s} = \sqrt{\frac{S[\text{mm}^2]}{S[\text{pix}^2]}} \quad (4.12)$$

Kalibrační obrazec je na bílém podkladě vyplněn černou barvou. Černá barva zajistí jednoduché algoritmické určení, zdali je daný pixel součástí obrazce či obklopujícího okolí. Nejvhodnějším tvarem obrazce je čtverec, jelikož je u něho jednoduché naměřit délkové parametry a tím i určit obsah v jednotkách délky. Toto měření se provede opět pouze jednou po vytvoření kalibračního obrazce a tato hodnota je nadále uchovávána spolu s kalibračním obrazcem. Vyšetřování obsahu obrazce má oproti jednoduššímu měření pouhé délky výhodu v tom, že obrazec může mít na referenční rovině libovolnou polohu. Naproti tomu při určování délky by však měřený délkový rozměr měl být rovnoběžný s řádky kamery, aby nedocházelo ke zkreslení rozměru ve stejném smyslu jako u předchozího způsobu kalibrace (viz obr. 4.5).

Algoritmus kalibrace opět začíná vyfocení snímku referenční roviny, na které je umístěn kalibrační obrazec. Dále je uživatelem vybrán výřez ze snímku tak, aby ve výřezu bylo pouze bílé okolí a černý kalibrační obrazec. Tento úkon neklade na obsluhu nijak velké nároky a podstatně zjednoduší vyhodnocení takového výřezu. Okolní tmavé plochy nepatřící obrazci by značně stěžovali identifikaci původu jednotlivých pixelů a tím zbytečně ztěžovali celý algoritmus. Poté je algoritmicky určena tzv. prahovací hodnota. Pixely s větší hodnotou intenzity než je prahovací hodnota budou brány jako bílé okolí. Pixely s hodnotou menší než je prahovací hodnota budou přisuzovány kalibračnímu obrazci. Prahovací hodnota je určena jako střed mezi maximální a minimální hodnotou intenzity vyskytující se ve výřezu. Po určení původu jednotlivých pixelů jsou spočítány ty, jenž náleží kalibračnímu obrazci. Tím je určen obsah obrazce v pixelech a za předpokladu znalosti obsahu obrazce v jednotkách délky již nic nebrání výpočtu kalibrační konstanty ze vzorce (4.12).

Kód algoritmu je následující:

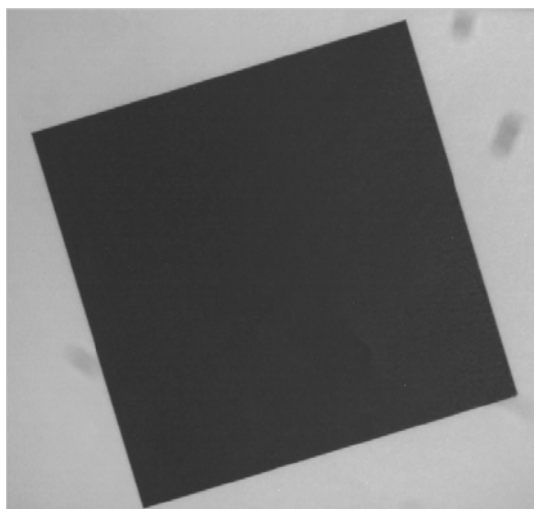
```
1: A = double(getsnapshot(video));
2: B = A (ymin:ymax,xmin:xmax);
3: min1 = min(min(B));
4: max1 = max(max(B));
5: prahovaci_hodnota = (min1 + max1)/2;
```

```

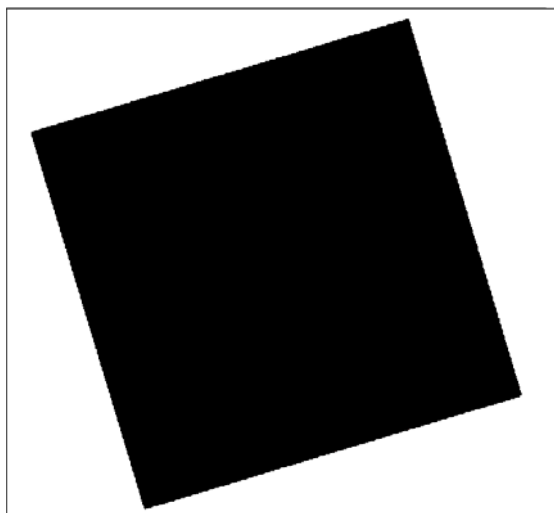
6:   suma = 0;
7:   for i =1:(size(vyrez,1))
8:       for j = 1:(size(vyrez,2))
9:           if B(i,j) < prahovaci_hodnota
10:               suma = suma + 1;
11:           end
12:       end
13:   end
14:   kalibracni_konstanta = sqrt(obsah/suma);

```

Na řádcích 1 až 5 je provedeno vyfocení snímku, ořez za pomoci dat získaných z interakce s obsluhou (proměnné  $y_{min}$ ,  $y_{max}$ ,  $x_{min}$ ,  $x_{max}$ ) a výpočet prahovací hodnoty. Dvojicí cyklů na řádcích 7 až 13 je u každého pixelu zjištěno zda patří okolí nebo kalibračnímu obrazci, pokud je součástí obrazce, je inkrementována hodnota proměnné  $suma$ . Po skončení cyklů je tedy v proměnné  $suma$  obsah kalibračního obrazce v pixelech. Na posledním řádku je dle vzorce (4.12) vypočítána kalibrační konstanta, která je výstupem algoritmu. Na obr. 4.8 je typický snímek referenční roviny s kalibračním obrazcem, který je v tomto případě čtvercový. Tvar kalibračního obrazce může být teoreticky libovolný, dokonce se ani nemusí jednat o spojitou plochu. Je ovšem vhodné volit takové obrazce, aby bylo jednoduché určit jejich obsah v jednotkách délky. Na obr. 4.9 je pak ilustrován jev prahování, kdy černou barvu mají pixely, které algoritmus vyhodnotil tak, že patří kalibračnímu obrazci. Bílou barvu pak mají ostatní pixely.



*Obr. 4.8 Výřez ze snímku referenční roviny s umístěným kalibračním obrazcem*



*Obr. 4.9 Výsledek vyhodnocení původu pixelů algoritmem*

#### **4.2.4 Srovnání obou metod kalibrace příčných os**

Oba uvedené způsoby představují spolehlivé a přesné řešení kalibrace příčných os. Hlavním cílem bylo vytvořit algoritmy, které by minimalizovaly nároky na obsluhu a při tom byla zachována spolehlivost a přesnost určení kalibrační konstanty. Pro srovnání obou způsobů bylo provedeno pro každý způsob deset měření kalibrační konstanty za pomoci uvedených algoritmů. Při každém jednotlivém měření byl obrazec umístěn v náhodné orientaci na referenční rovinu. Podmínky sestavy při měření byly pro oba způsoby kalibrace totožné. Výsledky měření shrnuje tabulka 4.1. Pro orientační kontrolu správnosti naměřených hodnot, byla kalibrační konstanta určena také manuálně určením počtu pixelů, který odpovídal vzdálenosti 13 cm. Kalibrační konstanta při manuálním měření byla stanovena na:  $K = 0.11383 \text{ mm/pix}$ .

Z výsledků měření v tabulce 4.1 je patrné, že podstatně jednodušší kalibrace pomocí obsahu obrazce dává přesnější výsledky a dokonce i opakovatelnost měření je větší. I tak dosahuje kalibrace pomocí sinusové mřížky dostačujících výsledků a je tedy použitelná. Při možné volbě je ovšem lepší použít kalibraci pomocí obsahu obrazce. Dále je nutné si uvědomit, že kalibraci příčných os je nutné provést vždy, když dojde ke změně vzdálenosti referenční roviny od kamery, či ke změně zvětšení použitého objektivu kamery.

	Kalibrace pomocí sin. mřížky	Kalibrace pomocí obsahu obrazce
Číslo měření	Kalibrační konstanta [mm/pix]	Kalibrační konstanta [mm/pix]
1	0,11399	0,11351
2	0,10824	0,11362
3	0,10873	0,11365
4	0,10935	0,11354
5	0,10708	0,11361
6	0,10874	0,11357
7	0,10803	0,11302
8	0,11205	0,11336
9	0,11141	0,11339
10	0,11453	0,11342
Průměr:	0,1102	0,1135
Chyba průměru:	0,0008	0,0001

*Tabulka 4.1 Výsledky měření kalibračních konstant*

### 4.3 Proces měření metodou PSP

Cílem měření metodou phase-shifting profilometrie je získání informací o profilu měřeného předmětu. Prvotní veličinou, která nese tuto informaci je fáze určená vztahem (2.5). Pro určení této fáze je nutné naměřit celkem 8 intenzitních snímků, kdy 4 snímky odpovídají referenční rovině a další 4 měřenému předmětu. Proces určení čtveřice snímků spočívá v postupném promítání jednotlivých sinusových mřížek s definovanou fází na předmět či referenční roviny a zachycení snímků takto promítnutých mřížek.

Po výpočtu fáze pomocí vztahu (2.5) dostaneme matici fáze, která má rozměry odpovídající rozlišení použité kamery. Jak již bylo uvedeno v teorii PSP, obsahuje tato fázová matice skoky, které neodpovídají skutečným skokům v profilu měřeného předmětu, nýbrž jsou výsledkem jevu zabalování fáze. Pro získání matice fáze bez těchto skoků je třeba aplikovat nějakou variantu rozbalovacího algoritmu. Rozbalovacích algoritmů existuje celá řada. Jejich vývoj či popis je vzhledem k složitosti nad rámec této práce. Ovšem bez rozbalovacího algoritmu se tato metoda neobejde. Využit byl proto algoritmus, který je výsledkem bakalářské práce [6], která byla tomuto tématu věnována. Daný algoritmus se



nazývá flood-fill algoritmus. Po aplikaci rozbalovacího algoritmu již fázová matice svými hodnotami odpovídá tvaru skutečného profilu předmětu.

### 4.3.1 Algoritmus měřícího procesu

Cílem algoritmu je naměřit výše zmiňovaných 8 snímků a následně vypočítat fázovou matici. Dále na tuto matici aplikovat rozbalovací algoritmus. Získávání snímků spočívá v promítání sinusových mřížek. Součástí algoritmu bude tedy i vytváření sinusových mřížek. Při jejich generaci bude využito výsledků s korekce přenosové funkce dataprojektoru. Po vytvoření sinusových mřížek, budou tyto promítány na předmět či referenční rovinu a pomocí kamery budou zachyceny snímky. Po zachycení všech požadovaných snímků bude vypočítána fázová matice.

První je zmíněna část algoritmu pro generaci sinusových mřížek. Tyto mřížky jsou matice s rozměrem odpovídajícím rozlišení dataprojektoru. Řádky matice vykazují sinusový průběh. Vzhledem k použitému počtu sinusových mřížek, který se rovná čtyřem, je nejprve vygenerován pro každou ze čtyř matic jeden řádek ze sinusovým průběhem s požadovanou fází. Následně je tímto jedním řádkem vyplněna celá matice. Výše popsáním činností odpovídá následující kód, kdy je generace ukázána pouze pro jednu matici:

```
1: x = 0:1023;
2: y2 = korekce(((255)/2)*(sin((2*pi*x/p)+faze)+1));
3: B1 = zeros(768,1024);
4: B1 = meshgrid(y1);
```

V kódu je proměnnou  $p$  označena perioda generované sinusové mřížky v pixelech, proměnnou  $faze$  je označen fázový posuv dané mřížky, který je pro každou mřížku jiný (viz obr. 2.4) a proměnnou  $korekce$  je označena funkce, která je výsledkem algoritmu pro korekci přenosové funkce dataprojektoru. Konečná matice se sinusovou mřížkou je uložena do proměnné  $B1$ .

Následně jsou jednotlivé mřížky postupně promítány na předmět či referenční rovinu, kdy po promítnutí je zachycen snímek kamerou:

```

1:   for i = 1:4
2:       switch i
3:           case 1
4:               set (im,'CData',B1);
5:           case 2
6:               set (im,'CData',B2);
7:           case 3
8:               set (im,'CData',B3);
9:           case 4
10:              set (im,'CData',B4);
11:        end
12:        pause(1);
13:        switch i
14:            case 1
15:                R1 = double(getsnapshot(video_object));
16:            case 2
17:                R2 = double(getsnapshot(video_object));
18:            case 3
19:                R3 = double(getsnapshot(video_object));
20:            case 4
21:                R4 = double(getsnapshot(video_object));
22:        end
23:    end

```

Proměnnými B1 až B4 byly označeny jednotlivé matice sinusových mřížek. Snímky z kamery byly uloženy do proměnných R1 až R4. Pokud uvažujeme, že v proměnných R1 až R4 máme jednotlivé snímky referenční roviny a v proměnných I1 až I4 jednotlivé snímky předmětu, pak fázová matice bude určena následujícím kódem:

```

1:   am=I3-I1;
2:   ar=R3-R1;
3:   bm=I4-I2;
4:   br=R4-R2;
5:   Matrix=( 1i*((am.*br)-(bm.*ar)) + ((bm.*br)+(am.*ar)));

```

```
6: matice_faze=angle(Matrix);
```

Zde je místo klasického využití funkce `arcustangens` využito funkce `angle`, která je součástí matematického prostředí MATLAB. Tato funkce vrací argument komplexního čísla, tedy úhel z intervalu  $\langle -\pi, \pi \rangle$ . Pokud tedy vytvoříme komplexní číslo  $z = b + i \cdot a$ , kde  $a$  je rovnou čitateli a  $b$  jmenovateli ve vzorci  $\alpha = \arctan\left(\frac{a}{b}\right)$ , pak jsme schopni vypočítat úhel  $\alpha$  pomocí funkce `angle` s tím rozdílem, že určený úhel bude s intervalu  $\langle -\pi, \pi \rangle$  namísto intervalu  $\left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle$ , který je oborem hodnot funkce `arcustangens`.

Na závěr je již jen na matici fáze aplikován rozbalovací algoritmus:

```
rozbalena_faze = unwrap (matice_faze);
```

Kdy funkce `unwrap` je implementace flood-fill rozbalovacího algoritmu.

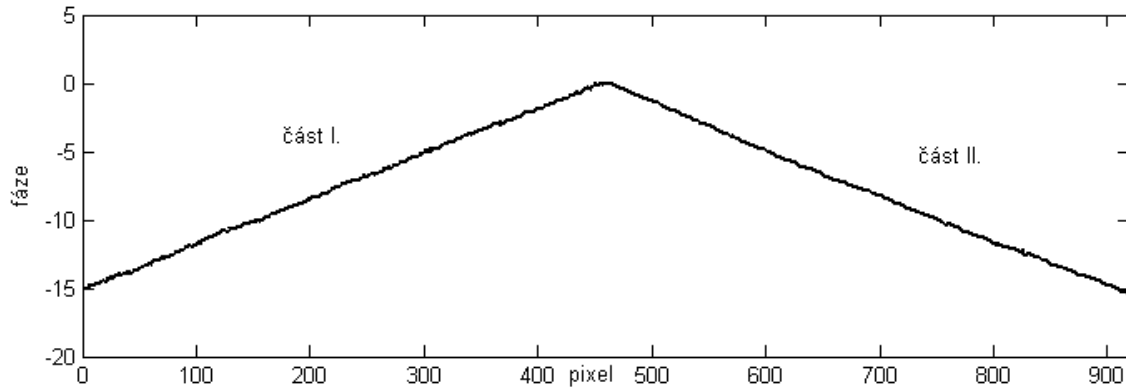
## 4.4 Kalibrace podélné osy (osy z)

Kalibrace osy z slouží k určení vztahu mezi naměřenou fází a měřítky příčných os. Pokud jsou pak příčné osy kalibrovány na jednotku délky, pak i po kalibraci osy z máme na této ose jednotku délky. Podobně jako u kalibrace příčných os, je kalibrace osy z prováděna pomocí předmětu o známém profilu. Z faktu, že víme, jaké vlastnosti má skutečný profil, jsme schopni určit konstantu, která uvede do souvislosti naměřenou fází a skutečný profil. Po naměření fáze lze pak za pomoci této kalibrační konstanty obdržet profil v požadovaných jednotkách.

Vhodným předmětem pro kalibraci výše uvedeným způsobem je trojúhelníkový hranol, v jehož jednom vrcholu je pravý úhel. Realizace takového hranolu je na obr. 4.10. Měřením takového hranolu, který je umístěn tak, aby podstava hranolu byla rovnoběžná s řádky kamery a vrchol směřoval směrem ke kameře, dostaneme matici fáze, kde řádky této matice budou tvořit dvě části přímek svírající obecný úhel. Kalibrace osy z spočívá v určení kalibrační konstanty, kterou když budou vynásobeny všechny hodnoty matice fáze, dostaneme na ose z stejnou jednotku jako na příčných osách.



Obr. 4.10 Realizace kalibračního hranolu s pravoúhlým vrcholem



Obr. 4.11 Řádek fázové matice kalibračního hranolu

Kalibrační konstanta je určena právě z té podmínky, že přímky z řádku profilu kalibračního hranolu mají svírat pravý úhel. Je tedy využito znalosti vlastností měřeného předmětu. V řádku fázové matice svírají části dvou přímek obecně libovolný úhel. Řádek fázové matice kalibračního hranolu je zobrazen v obr. 4.11. Pro každou část přímky lze určit směrnici. Pokud dojde k vynásobení všech hodnot fázové matice konstantou, směrnice přímek se změní ve stejném poměru jako hodnoty matice:

$$A'_i = KA_i, \quad (4.13)$$

kde  $K$  je kalibrační konstanta,  $A'_i$  je směrnice části přímky po vynásobení matice kalibrační konstantou a  $A_i$  je směrnice části přímky fázové matice. Pro účely výpočtu je vhodné přejít

od směrnic ke směrovým vektorům částí přímek. Směrový vektor lze pomocí směrnice určit ze vztahu [1]:

$$\vec{s}_i = (1, A_i) \quad (4.14)$$

Pokud víme, že po vynásobení kalibrační konstantou má být úhel svíraný částmi přímek pravý, tak z vlastnosti skalárního součinu pro směrové vektory platí:

$$\vec{s}'_1 \cdot \vec{s}'_2 = (1, A'_1) \cdot (1, A'_2) = (1, KA_1) \cdot (1, KA_2) = 1 + K^2 A_1 A_2 = 0. \quad (4.15)$$

Vyjádřením neznámé  $K$  ze vztahu (4.15) obdržíme finální vztah pro kalibrační konstantu [1]:

$$K = \sqrt{\frac{-1}{A_1 A_2}}. \quad (4.16)$$

Pro určení kalibrační konstanty pomocí pravoúhlého hranolu tedy stačí znát směrnice částí přímek řádku fázové matice.

#### 4.4.1 Algoritmus kalibrace osy z

Kalibrace osy z poskytuje svázání jednotek příčných os ( $x$  a  $y$ ) a osy  $z$ . V takovéto soustavě má pak naměřený profil totožné proporce jako skutečný předmět. Pokud je potřeba zobrazit výsledky v jednotkách délky, je po kalibraci osy  $z$  možné změnit měřítko všech os pomocí kalibrační konstanty pro příčné osy. To je dovoleno, jelikož po kalibraci osy  $z$  má tato osa také jednotku pixel. Výsledkem kalibrace osy  $z$  je kalibrační konstanta, pomocí které jsou vynásobeny všechny hodnoty fázové matice za účelem získání stejné jednotky na všech osách.

Vstupem pro algoritmus je řádek fázové matice, která vznikla měřením správně umístěného kalibračního hranolu s pravoúhlou hranou směřující ke kameře. Z toho důvodu byl algoritmus měření uveden dříve než kalibrace osy  $z$ . Výstupem je pak kalibrační konstanta pro osu  $z$ . Tato konstanta je vypočítána ze vztahu (4.16). Cílem algoritmu je tedy určit směrnice částí přímek. Neideálně rovný tvar předmětu či šum fázové matice způsobuje, že části přímek nejsou ideálně rovné. Pro určení směrnic těchto částí přímek je tedy nutné použít

aproximační metodu, která se průběhem snaží proložit přímkou. Z parametrů této přímky je pak určena směrnice. Algoritmus určení kalibrační konstanty se skládá z následujícího kódu:

```
1: [val, pozice] = max(radek);
2: cast1 = radek(1:pozice);
3: cast2 = radek(pozice:size(radek, 2));
4: p1x = 1:size(cast1, 2);
5: primka1 = fit(p1x', cast1', 'poly1');
6: p2x = 1:size(cast2, 2);
7: primka2 = fit(p2x', cast2', 'poly1');
8: kalibracni_konst_z = sqrt((-1)/(primka1.p1*primka2.p1));
```

Příkazem na prvním řádku je zjištěna pozice v řádku, která rozděluje dvě části přímek. Ta je vyhledána jako maximální hodnota v celém řádku. Na řádcích 2 a 3 je pak každá část překopírována do samostatného vektoru. Na řádcích 4 až 7 je provedeno proložení obou částí polynomem prvního stupně, tedy přímkou. Na posledním řádku je z parametrů jednotlivých polynomů určena kalibrační konstanta.

Kalibrace osy z pomocí pravoúhlého kalibračního hranolu představuje jednoduchý a přesný způsob určení vztahu mezi naměřenou fází a skutečnými výškovými parametry měřeného předmětu. Hlavní výhodou oproti jiným způsobům je, že není nutné měřit jakékoliv geometrické parametry sestavy, jejichž měření bývá často velmi obtížné. Pro úplnost je vhodné doplnit, že kalibraci osy z je nutné provést znovu, pokud dojde k jakékoliv změně v sestavě stejně jako v případě kalibrace příčných os. Navíc, jelikož vztah mezi naměřenou fází a výškovými rozdíly je značně závislý na použité periodě promítané sinusové mřížky, je kalibraci nutné provést při každé změně této periody.

## 5. Implementace algoritmů do grafického uživatelského rozhraní

Programové prostředí MATLAB umožňuje vývoj a spouštění GUI aplikací. Pro snadné zpřístupnění algoritmů uživateli měřicí sestavy PSP byly tyto algoritmy implementovány do aplikace založené na GUI. Jednotlivé algoritmy jsou uživatelem spouštěny za pomoci ovládacích prvků formuláře, který tvoří celkový grafický vzhled aplikace. Za pomoci takové aplikace je pak možné snadnou cestou ovládat celou experimentální sestavu a využívat ji k měření.

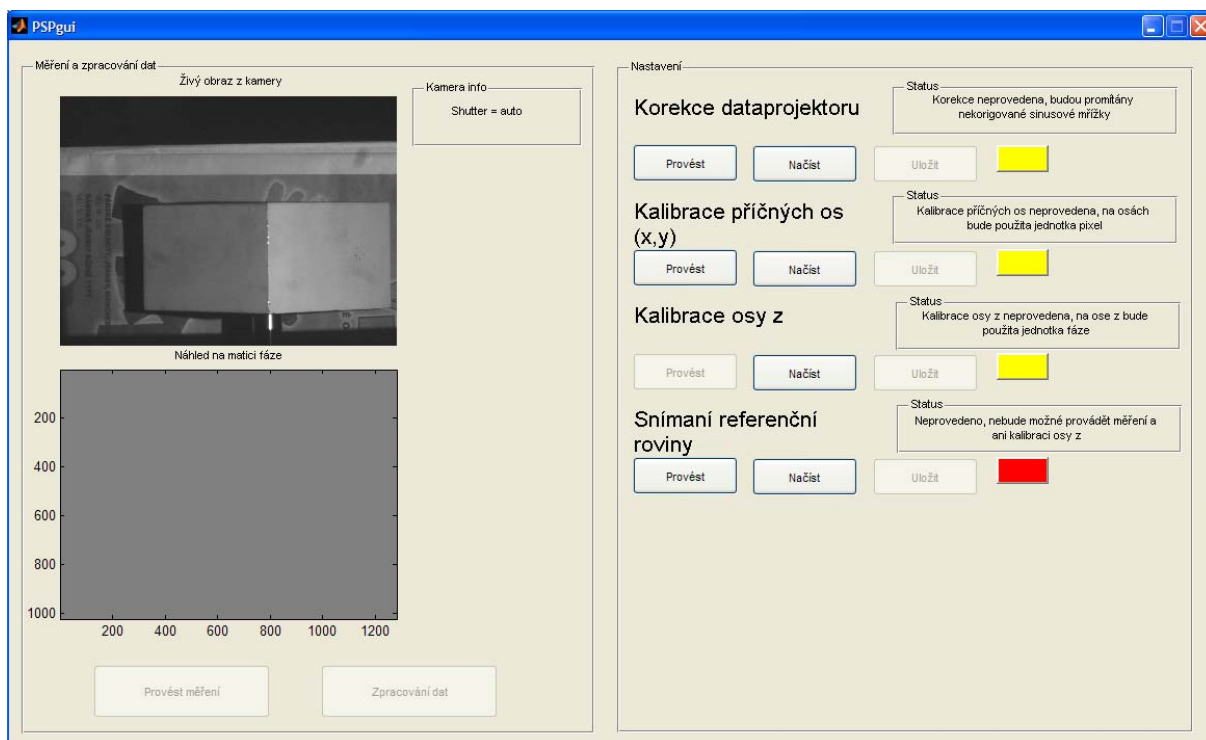
Vytváření GUI aplikace sestává ze dvou významných kroků. Nejprve je ve vývojovém prostředí vytvořen vzhled formuláře, na který se umísťují jednotlivé ovládací či zobrazovací prvky. V našem případě se jedná především o tlačítka, textová pole a grafická pole. Tlačítka slouží ke spouštění jednotlivých kroků spojených s nastavením, kalibrací či samotným měřením. Pomocí textových polí je uživatel informován o stavu jednotlivých procedur. Grafická pole pak slouží pro zobrazování naměřených dat či k zobrazování živého obrazu z kamery.

Po vytvoření formuláře a po umístění jednotlivých ovládacích či zobrazovacích prvků následuje krok zahrnující programování jednotlivých funkcí, které jsou spouštěny různými událostmi na formuláři (stisk tlačítka apod.). Ve většině případu jsou po stisku tlačítka volány malou měrou upravené algoritmy popisované v předchozích částech práce. Přidané či upravené části kódu algoritmu jej pouze přizpůsobují pro použití v grafickém uživatelském rozhraní. Jedná se tak většinou o dialogová okna pro zadání parametrů potřebných pro úspěšné dokončení algoritmu, výběr výřezu z matice hodnot či okna zobrazující průběh u algoritmů časově náročných apod.

Výše zmiňované úpravy kódu nemají žádný vliv na původní principy vyvinutých algoritmů. Vzhledem k rozsahu nebude celkový kód tvořící funkční část grafického rozhraní uváděn. Pozornost bude spíše věnována filozofii fungování celého grafického uživatelského rozhraní. Budou také popsány jednotlivé prvky formuláře s ohledem na jejich funkčnost. Tato kapitola tedy spíše tvoří návod na používání vytvořeného grafického rozhraní.

## 5.1 Popis grafického uživatelského rozhraní

Jádro celé aplikace sloužící k ovládání experimentální sestavy pro (PSP) tvoří formulář zobrazen na obr. 5.1. Ten je rozdělen do dvou rámců. První z nich s názvem *Měření a zpracování dat* obsahuje dvě grafická pole, dvě tlačítka a jedno textové pole. V prvním Grafickém poli je zobrazen živý obraz z kamery. Druhé grafické pole je před provedením měření prázdné. Po provedení měření je zde zobrazen náhled fázové matice, kdy hodnoty matice jsou reprezentovány jako body s různými stupni šedi. Dále obsahuje tento rámeček dvě tlačítka *Provést měření* a *Zpracování dat*. Tlačítkem *Provést měření* je spuštěn měřicí algoritmus, tedy sekvenční promítání sinusových mřížek a snímání scény kamerou. Toto tlačítko se stává aktivní až po provedení měření referenční roviny, jelikož snímky referenční roviny jsou nutné pro výpočet fázové matice měřeného předmětu. Dalším tlačítkem je *Zpracování dat*, které otevře další okno grafického uživatelského rozhraní, kde má uživatel možnost provádět potřebné operace s fázovou maticí (aplikace rozbalovacího algoritmu, ořez fázové matice, prezentace výsledků atd.).



Obr. 5.1 Vzhled hlavního formuláře grafického uživatelského rozhraní

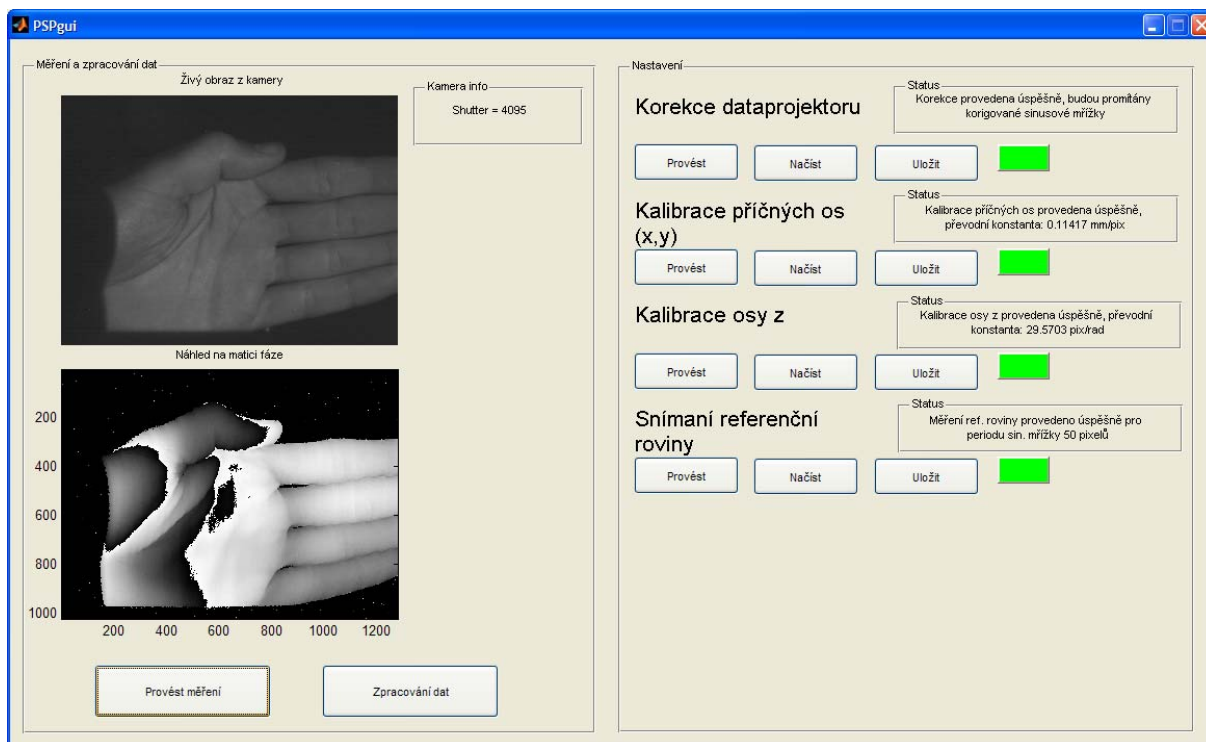
V druhém rámečku formuláře s názvem *Nastavení* jsou umístěny tlačítka pro obsluhu jednotlivých procesů týkajících se nastavení a kalibrace experimentální sestavy pro PSP.



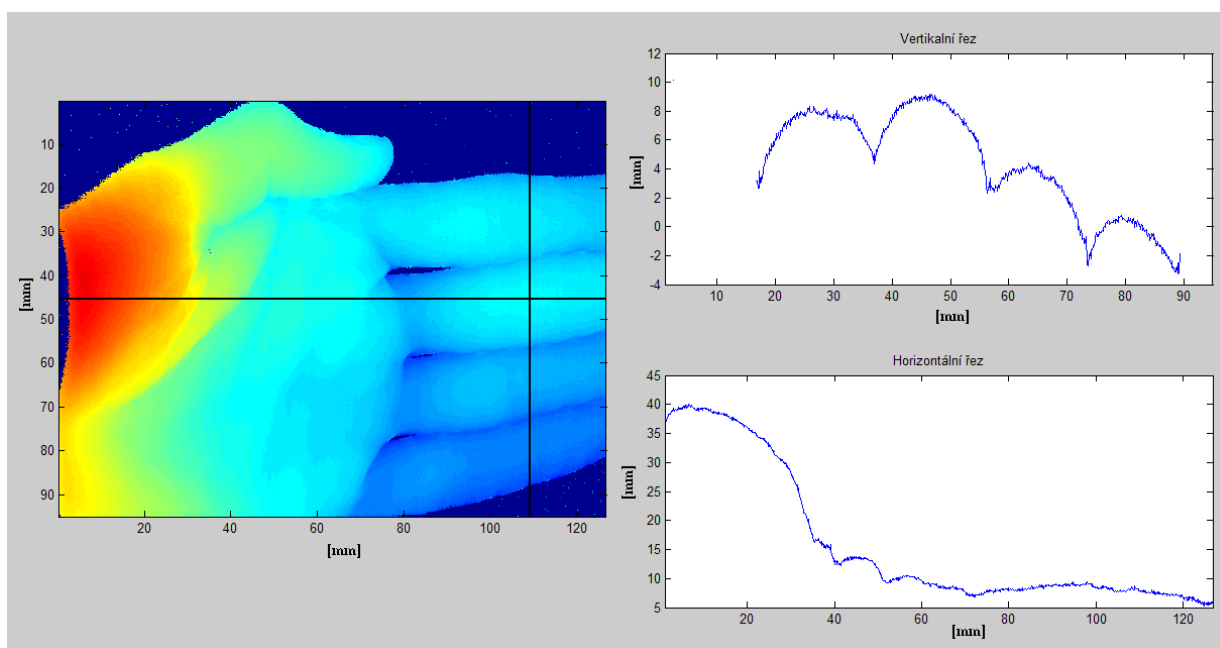
Jmenovitě jsou to korekce přenosové funkce dataprojektoru, kalibrace příčných os, kalibrace osy z a snímání referenční roviny. Pro každou činnost je možno tlačítkem *Provést* spustit algoritmus vykonávající danou operaci. Dále je také možné nahrát data již dříve provedených operací a to tlačítkem *Načíst*. Po provedení některé z činností je možné výsledky uložit pro pozdější použití tlačítkem *Uložit*. V textovém poli *Status* je u každé operace stručně popsáno v jakém stavu je daná operace tzn., jestli byla či nebyla provedena popřípadě, jaký byl výsledek. Je zde také uvedeno jaký vliv bude mít tento stav na měření.

Stav jednotlivých operací je také zjistitelný z barevných indikátorů, kdy červená barva znamená neprovedená činnost, která dokud nebude provedena či nahrána, nebude možno provádět měření. Oranžová barva indikuje neprovedenou činnost, která ovšem nezabraňuje provedení měření. Zelená barva pak představuje v pořádku provedenou operaci či úspěšně načtené výsledky již dříve provedené operace. Programový kód aplikace nastavuje aktivitu jednotlivých tlačítek v závislosti na stavu jiných operací. Některé činnosti totiž nelze provádět, dokud nebudou úspěšně provedeny činnosti jiné. Například kalibrace osy z zahrnuje měření fázové matice kalibračního hranolu. K tomu je ovšem nutné mít jednotlivé snímky referenční roviny. Tlačítko pro provedení kalibrace osy z bude tedy aktivní až po úspěšném snímání referenční roviny. Dále např. tlačítko *Uložit* je aktivní až po úspěšném provedení dané operace.

Po každém spuštění aplikace jsou všechny úkony v neprovedeném stavu. Před začátkem měření je tedy nutné provést jednotlivé operace nebo načíst výsledky již dříve provedených operací. Následně je již možné provádět měření předmětů. Vzhled formuláře po úspěšném provedení operací týkajících se nastavení a kalibrace a po měření lidské dlaně je na obr. 5.2. Na obr. 5.3 je pak zobrazeno okno pro analýzu naměřených dat, kdy pohybem myši po obrazu (kde je výchylka v ose z reprezentována pseudobarvami) jsou vybírány souřadnice pro vertikální a horizontální řez. Tyto řezy jsou pak zobrazeny ve vedlejším okně. Všechny osy mají měřítko v mm, při vytváření grafů je tedy využito výsledků z provedených kalibrací.



Obr. 5.2 Vzhled hlavního formuláře grafického uživatelského rozhraní po provedení všech úkonů týkajících se nastavení a kalibrace a po měření testovacího předmětu



Obr. 5.3 Analýza naměřených dat pomocí horizontálního a vertikálního řezu, všechny osy mají měřítko v milimetrech

Uvedená aplikace umožňuje jednoduchou formou využívat algoritmy popsané v této práci. Je tedy efektivním nástrojem pro využívání experimentální sestavy pro phase-shifting profilometrii. Pomocí několika málo operací, které se spouští zmáčknutím tlačítka se uživatel dostane až k samotnému měření 3D profilu povrchu předmětu. Aplikace ve výsledku zpřístupňuje metodu PSP i v dané problematice nekvalifikovaným osobám, což bylo cílem této práce.

## 6. Závěr

V diplomové práci je zkoumána metoda phase-shifting profilometrie z hlediska vývoje algoritmů pro automatizaci měření. V úvodu práce jsou nastíněny základní principy, na kterých je tato měřicí metoda založena. Následuje stručný popis této metody za uvedení základních rovnic pro výpočet veličin nesoucích informaci o profilu povrchu měřeného předmětu. Tyto rovnice jsou pak v upravené podobě použity při vytváření některých algoritmů.

K vývoji algoritmů bylo použito programové prostředí MATLAB, proto také tomuto prostředí byla věnována kapitola popisující základní rysy tohoto prostředí. S ohledem na experimentální sestavu, ve které figuruje kamera a dataprojektor, je uveden způsob ovládání těchto zařízení z programového prostředí MATLAB.

Hlavní část práce tvoří popis jednotlivých úkonů spojených s nastavením, kalibrací a měřením na sestavě pro PSP. Zde jsou také uvedeny principy a konkrétní řešení algoritmů provádějící dané operace. Jmenovitě se jedná o korekci intenzitního přenosu dataprojektoru, kalibraci příčných os, proces měření metodou PSP a kalibraci osy z.

Na závěr práce je popsáno grafické uživatelské rozhraní, které implementuje vyvinuté algoritmy aplikace umožňující jednoduché provádění operací spojených s měřením metodou PSP. Hlavním cílem práce bylo vytvořit právě zmiňované grafické rozhraní, a tím zpřístupnit metodu více lidem. Některé uváděné algoritmy nejsou přímo spojeny s metodou PSP a je proto možné použít i v jiných oblastech. Algoritmy mohou také sloužit jako inspirace k programovému řešení jiných, podobných problémů.

## Seznam použité literatury:

- [1] Báránek, R.: *Nastavení a kalibrace experimentální sestavy pro optickou phase-shifting profilometrii*. Bakalářská práce, PřF UP Olomouc (2009).
- [2] Su, H.-J., Li, J.-L., Su, X.-Y.: *Phase algorithm without the influence of carrier frequency*. Optical Engineering, Vol. 36, 1799-1804 (1997).
- [3] Brigham, E. O., Morrow, R. E.: *The fast Fourier transform*. IEEE Spectrum, Vol. 4, 63 - 70 (1967).
- [4] Čížek, V.: *Diskrétní Fourierova transformace a její použití*. 1. vyd. Praha: SNTL - Nakladatelství technické literatury, (1981).
- [5] Rektorys, K., aj.: *Přehled užití matematiky*. 4. vyd. Praha: SNTL, 1139 s., s. 925-933 (1981).
- [6] Dudka, M.: *Problém rozbalování fáze*. Bakalářská práce, PřF UP Olomouc (2009).
- [7] Srinivasan, V., Liu, H. C., Halioua, M.: *Automated phase-measuring profilometry of 3-D diffuse objects*. Applied optics, Vol. 23, 3105-3108 (1984).
- [8] Malacara, D.: *Optical shop testing, Second edition*. John Wiley & Sons, Inc., New York (1992).
- [9] Asundi, A., Wensen, Z.: *Fast phase-unwrapping algorithm based on a gray scale mask and flood fill*. Applied optics, Vol. 37, 5416-5420 (1998).
- [10] *MATLAB – Documentation*. The MathWorks, Inc., c2011 [cit. 28-04-2011]. Dostupná z WWW: <<http://www.mathworks.com/help/techdoc/>>.
- [11] Matušová, J.: *Kalibrace optických 3D metod měřením prostorového etalonu*. Diplomová práce, PřF UP Olomouc (2007).