

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ZPRACOVÁNÍ OBRAZU PŘI URČOVÁNÍ TOPOGRAFICKÝCH PARAMETRŮ POVRCHŮ

IMAGE PROCESSING BY DETERMINATION OF TOPOGRAPHIC SURFACE PARAMETERS

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE
AUTHOR

BC. MARTIN BOHÁČ

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. RNDR. ING. JIŘÍ ŠŤASTNÝ, CSC.

BRNO 2009

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Martin Boháč

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Zpracování obrazu při určování topografických parametrů povrchů

v anglickém jazyce:

Image processing within determination of topographic surface parameters

Stručná charakteristika problematiky úkolu:

K moderním metodám studia topografie povrchů patří metody optické. Jejich výstupem je obecně plošné rozložení intenzity světla nesoucí informaci o topografii studovaného povrchu. Získání této informace moderními metodami a algoritmy zpracování obrazu je významnou úlohou v charakterizaci topografie povrchů.

Cíle diplomové práce:

Cílem projektu je návrh metod a algoritmů pro identifikaci topografických parametrů povrchů a analýza simulačních experimentů s danými povrchy. Bude proveden návrh konkrétních algoritmů a jejich implementace ve vytvořeném simulačním programu v objektovém prostředí C++ nebo Delphi. Pro dané testovací povrchy budou prováděny simulační experimenty s následným vyhodnocením dosažených výsledků.

Seznam odborné literatury:

- [1] De Juan, Ch.: Contour Recognition Problem, [online]. 2001. Dostupné z: <www.cc.gatech.edu/classes/ay2000/cs7495_fall/participants/cnd/ps3/ps3.html>.
- [2] Ohlídal, M.: The Scattering of Light from Rough Surfaces and Its Application in Surface Roughness Measurement. Výrobné inžinierstvo, Vol.6, , No.4, 2007.
- [3] Ohlídal, M., Šír, L., Jákľ, M., Ohlídal, I.: Digital two-wavelength holographic interference microscopy for surface roughness measurement. Slovak-Czech-Polish Conference on Wave and Quantum Aspects of Contemporary Optics, Nitra, 2005.

Vedoucí diplomové práce: doc. RNDr. Ing. Jiří Šťastný, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2008/2009.

V Brně, dne 6.11.2008

L.S.

doc. RNDr. Ing. Miloš Šeda, Ph.D.
Ředitel ústavu

doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

LICENČNÍ SMLOUVA

ABSTRAKT

Práce se zabývá určením topografických parametrů náhodně drsného povrchu pomocí metody stříhové interferometrie. Jedná se o optickou metodu určení drsnosti povrchu. Základní myšlenka je založena na deformaci interferenčních proužků vzniklých interferencí dvou stejných navzájem posunutých a nakloněných monochromatických světelných vlnoploch. Interferující vlnoplocha vznikne průchodem nebo odrazem monochromatického světla od povrchu zkoumaného tělesa. Interferující vlnoplochy vytváří obraz s deformovanými interferenčními proužky, který nese informaci o charakteru povrchu. Tato informace se z obrazu dá získat pomocí algoritmů zpracování obrazu. Tato diplomová práce byla zpracována v rámci vědecko-výzkumného záměru MSM 0021630529 Inteligentní systémy v automatizaci.

ABSTRACT

This work deal with determination topographic parameters of a randomly rough surface by the help of method of shearing interferometry. It is a optical method for determination surface roughness. The basic idea is based of on deformation interference strips which are made by interference of the same mutually translated monochrome luminous wavefronts. The wavefront is created after transit or reflection monochrome lights from the surface of a studied sample. The wavefronts creates picture with deformed interference strips , which carries information about character of the surface. This information can be profited by algorithms of image processing from the picture . The thesis was developed in research project MSM 0021630529 Intelligent Systems in Automation.

KLÍČOVÁ SLOVA

metoda stříhové interferometrie, náhodně drsný povrch, reprezentace digitalizovaného obrazu, předzpracování obrazu, hledání hran obrazu, hledání objektů obrazu

KEYWORDS

method of shearing interferometry, randomly rough surface, representation of digital image, image preprocessing, edge detection in image, detection of objects in image

Obsah:

	Zadání závěrečné práce.....	5
	Licenční smlouva.....	7
	Abstrakt.....	9
1	Úvod.....	13
2	Optické metody při určování topografických parametrů povrchů.....	15
2.1	Charakteristiky náhodně drsného povrchu používané v optických metodách.....	16
2.2	Vyšetřování povrchu pomocí úhlové korelace polí koherenční zrnitosti.....	17
2.3	Vyšetřování povrchu pomocí rozdělení intenzity rozptýleného laserového světla.....	18
2.4	Vyšetřování povrchu pomocí střihové interferometrie.....	18
2.4.1	Vlastnosti studovaného náhodně drsného povrchu.....	19
2.4.2	Základní princip střihové interferometrie.....	19
2.4.3	Závěrem k metodě střihové interferometrie.....	25
3	Algoritmy zpracování obrazu při určování topografických parametrů obrazu.....	27
3.1	Reprezentace digitalizovaného obrazu.....	27
3.2	Předzpracování obrazu.....	28
3.2.1	Jasová a kontrastní korekce.....	28
3.2.2	Převod na stupně šedi.....	29
3.2.3	Ekvalizace histogramu.....	29
3.2.4	Filtrace obrazu.....	30
3.2.5	Ostření obrazu.....	32
3.3	Segmentace obrazu.....	32
3.3.1	Prahování.....	33
3.3.2	Barvení.....	34
3.3.3	Hledání hran.....	35
3.3.4	Sledování hranice a propojování hran.....	39
3.3.5	Vyplňování objektů.....	41
3.4	Popis objektů.....	42
4	Algoritmy pro vytváření náhodně drsného povrchu a simulaci interferenčního mikroskopu.....	43
4.1	Vytvoření náhodně drsného povrchu pomocí fraktálů.....	43
4.2	Simulace interferenčního mikroskopu.....	45
5	Konkrétní návrh algoritmů zpracování obrazu při určování parametrů povrchu pomocí metody střihové interferometrie.....	47
5.1	Předzpracování interferogramu.....	48
5.1.1	Filtrace interferogramu.....	48
5.1.2	Hledání středů temných a světlých proužků interferogramu.....	49
5.2	Hledání jednopixelových linií.....	51
5.2.1	Bipolarizace interferogramu.....	52
5.2.2	Separace interferogramu.....	53
5.2.3	Nalezení jednopixelových linií.....	55
5.3	Statistické zpracování jednopixelových linií.....	57
6	Popis prostředí a struktury softwaru pro určování topografických parametrů povrchů.....	59
6.1	Struktura softwaru.....	59
6.2	Seznámení s uživatelským prostředím softwaru.....	64
7	Závěr.....	69
	Seznam použité literatury.....	71

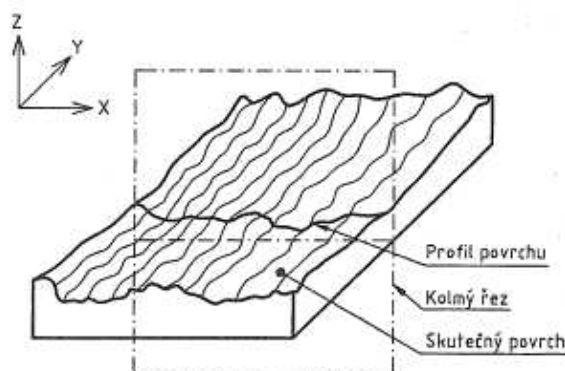
1 ÚVOD

Studování topografických parametrů povrchu je velice důležité odvětví strojírenského průmyslu. Drsnost povrchu má totiž velký vliv na fyzikální a chemické vlastnosti pevných látek. Potřeba mít nějaké nástroje pro popis drsnosti povrchů sebou přinesla zavádění různých veličin, které nějakým způsobem studovaný povrch charakterizují. Může jít například o rozdělení výšek povrchu, odchylek sklonů, autokorelační vzdálenost atd.

Kontrola kvality povrchů ve strojírenském průmyslu přináší zvýšení estetičnosti produktů, životnosti výrobků a odhalení vadných výrobků dříve než přijdou na trh, kde by mohly způsobit škody na majetku nebo dokonce ztráty na lidských životech. Z těchto důvodů se vlastnosti povrchu kontrolují v mnoha oblastech jako je například automobilový průmysl (kontrola povrchu před nanášením barvy na karoserii), oblast tribologie (snížení opotřebení povrchů), mikroelektronický průmysl (výroba integrovaných obvodů), bioinženýrství (prodloužení životnosti kloubů, snadnější přijetí umělých orgánů tělesnou tkání) a mnoho dalších.

Možností jak kvalitu povrchu ohodnotit je mnoho. Někdy se používá subjektivní hodnocení zaměstnanců, jindy se zas uplatní objektivní metody. Ty se dělí na dvě skupiny: dotykové profilometry a optické metody. První skupina vede k 2D vyhodnocení povrchu, což není ideální, přesto jde v současnosti o nejpoužívanější metodu. Ta druhá již vystihuje 3D charakter struktury povrchu a s pomocí využití dnešní komerčně dosažitelné výpočetní techniky je relativně rychlá a levná.

Právě na optické metody je tato práce zaměřena. Jejich výstupem bývá obraz různých jasových úrovní, který charakterizuje zkoumaný povrch. Tento obraz se převádí do digitální podoby a pak se zpracovává pomocí výpočetní techniky. Tato práce se zaměří především na metodu stříhové interferometrie a na návrh a realizaci algoritmů zpracování digitalizovaného obrazu vedoucích k určení topografických parametrů povrchu. Nalezneme zde popis některých metod předzpracování obrazu pro odstranění lokálních chyb a pro úpravu obrazu zjednodušující následující zpracování. Ukážeme si několik řešení samotného zpracování obrazu a také konkrétní návrh zpracování obrazu vedoucí k získání topografických parametrů povrchu.



Obr.1.1 Profil povrchu a souřadnicový systém [14]

2 OPTICKÉ METODY PŘI URČOVÁNÍ TOPOGRAFICKÝCH PARAMETRŮ POVRCHŮ

Optické metody pro určování topografických parametrů nejsou žádnou novinkou. Již ve 30. letech 20. století se začaly využívat v průmyslovém odvětví. Jejich nasazení v praxi přináší 4 hlavní výhody oproti používání klasického profilometru. Jsou založené na bezkontaktním principu, v praxi to znamená žádné opotřebování povrchu při zjišťování jeho struktury. Druhou výhodnou vlastností je možnost studovat povrch o téměř libovolných rozměrech. Další velikou předností je jejich relativně rychlé vyhodnocení výsledku, ten je k dispozici téměř okamžitě. Za čtvrtou hlavní výhodu můžeme považovat jejich širokou škálu použitelnosti. Dají se využít i pro povrchy, které nelze pomocí profilometru analyzovat z důvodu jejich velkých rozměrových rozdílů profilu.

V průmyslové praxi se však ve větší míře používají kontaktní profilometry. Pro využití tohoto nástroje jsou na rozdíl od optických přístrojů definovány ISO normy. K tomu charakteristiky získané oběma způsoby bývají odlišné a v mnoha případech se nadají porovnat. Proto je problém optické přístroje dostat na trh, ve kterém má tak výsostně postavení kontaktní profilometr. Optické metody mají také své úskalí. Je totiž důležité, aby zkoumaný povrch splňoval určité vlastnosti potřebné k interakci světla. Velkou nevýhodou optických metod oproti dotykovým je také jejich menší tolerance vůči nečistotám. Vyžadují velmi čistý povrch, oproti tomu snímací hrot kontaktního profilometru dokáže drobné nečistoty odsunout a nevádí mu olejové skvrny. [1], [4]

Přehled výhod a nevýhod kontaktních i bezkontaktních metod je uveden v tab. 1 a v tab. 2.

Výhody optických přístrojů	Nevýhody optických přístrojů
-bezkontaktnost a nedestruktivnost	-nepřímé měření parametrů
-okamžité výsledky	-složitější interpretace výsledků
-schopnost kontinuální kontroly a řízení kvality povrchu	-ztížená korelovatelnost na normované parametry
-opakovatelnost měření	
-možnost měření z větší vzdálenosti	
-velká rychlost snímání	

Tab. 1 - Výhody a nevýhody optických metod [4]

Výhody dotykových přístrojů	Nevýhody dotykových přístrojů
-měří strojírenské povrchy	-destrukce povrchů u měkkých materiálů
-velká rozlišovací schopnost	-pouze 2D měření
-přímé měření parametrů topografie povrchu	-relativně pomalý pohyb hrotu po měřeném povrchu
-možná kalibrace výsledků z nepřímých metod vzhledem k přímému měření	-chybějící kontinuální kontrola a řízení kvality povrchu
-překoná zbytky materiálu a olejový film	-nemožnost dálkového řízení
-jednoznačně definované ISO standardy a povrchy	

Tab. 2 – Výhody a nevýhody mechanických metod [4]

Dále se budeme již zabývat pouze optickými koherentními metodami. V této kapitole si přiblížíme základní charakteristiky získané optickými metodami, pak si ukážeme základní principy optických koherentních metod a nakonec se podíváme na metodu stříhové interferometrie trochu podrobněji.

2.1 Charakteristiky náhodně drsného povrchu používané v optických metodách

V této podkapitole se budeme zabývat charakteristikami popisující vlastnosti náhodně drsného povrchu, se kterým se setkáváme téměř na každém reálném používaném tělese. Náhodná drsnost má vliv na fyzikální a chemické vlastnosti pevných látek, proto je velice důležité mít k dispozici veličiny, které ji dokáží popsat. Budeme zkoumat pouze povrchy, jenž vznikly izotropními, stacionárními stochastickými procesy. To znamená, že veličiny popisující tento povrch nezávisí na poloze ani směru.

K základním veličinám charakterizující náhodně drsný povrch patří: střední kvadratická odchylka výšek nerovností, střední kvadratická odchylka sklonů, autokorelační vzdálenost, autokorelační funkce drsného povrchu, rozdělení hustoty pravděpodobnosti výšek nerovností, rozdělení hustoty pravděpodobnosti parciálních derivací náhodné funkce popisující texturu povrchu a normovaná autokorelační funkce drsného povrchu. V následujících několika řádcích si některé z těchto charakteristik trochu přiblížíme.

Střední kvadratická odchylka výšek nerovností σ je definovaná vztahem:

$$\sigma = \sqrt{M_z} \quad , \quad (2.1)$$

kde

$$M_z = \int_{-\infty}^{\infty} (z - \langle z \rangle)^2 w(z) dz \quad , \quad (2.2)$$

kde z je výška nerovnosti povrchu, $\langle z \rangle$ je střední hodnota výšek nerovností a $w(z)$ je rozdělení hustoty pravděpodobnosti veličiny z . Jedná-li se o rovinný povrch, pak lze volit takový souřadnicový systém, že platí: $\langle z \rangle = 0$.

Střední kvadratická odchylka sklonů $\tan \beta_0$ je definována vztahem:

$$\tan \beta = \sqrt{M_z'} \quad , \quad (2.3)$$

kde

$$M_z' = \int_{-\infty}^{\infty} (z')^2 w(z') dz' \quad , \quad (2.4)$$

kde z' je parciální derivace náhodné funkce $\zeta(x,y)$ charakterizující texturu náhodně drsného povrchu a $w(z')$ je rozdělení hustoty pravděpodobnosti parciální derivace funkce $\zeta(x,y)$. Vyšetřujeme-li povrch s Gaussovým rozdělením výškových nerovností, lze výše uvedený vztah zjednodušit na rovnici:

$$\tan \beta = \sqrt{(2)} \sigma / T \quad , \quad (2.5)$$

kde T je autokorelační vzdálenost.

Autokorelační funkce $B(\tau)$ je definována vztahem:

$$B(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} z_1 z_2 w(z_1, z_2, \tau) dz_1 dz_2, \quad (2.6)$$

kde z_1 a z_2 jsou výšky drsného povrchu ve dvou bodech vzdálených od sebe ve střední rovině o hodnotu τ a $w(z_1, z_2, \tau)$ je dvourozměrné rozdělení hustoty pravděpodobnosti náhodné funkce $\zeta(x, y)$. U povrchů mající Gaussův charakter, lze vztah (2.6) zjednodušit na rovnici:

$$B(\tau) = \sigma^2 \exp(-\tau/T^2), \quad (2.7)$$

kde τ je vzdálenost dvou bodů ve střední rovině povrchu (τ je proměnná).

Autokorelační vzdálenost T je definována jako vzdálenost τ pro níž platí:

$$B(\tau) = \sigma^2 / e \quad (2.8)$$

Normovaná autokorelační funkce (též nazývaná jako autokorelační koeficient) je definována vztahem:

$$C(\tau) = \left(\frac{B(\tau)}{\sigma^2} \right). \quad (2.9)$$

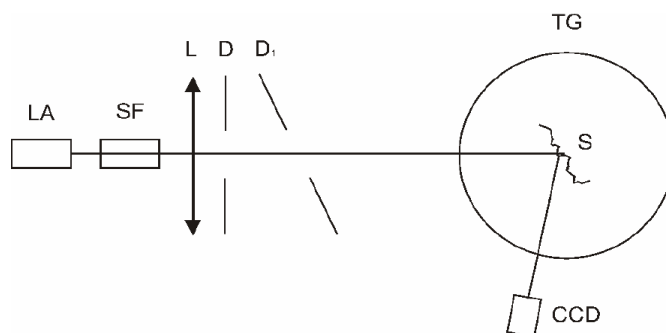
Tímto jsme definovali základní charakteristiky používané pro optické metody. Bližší popis naleznete v pracích [1], [3].

2.2 Vyšetřování povrchu pomocí úhlové korelace polí koherenční zrnitosti

U této metody si uvedeme jen základní princip. Podrobnější vysvětlení je obsaženo v pracích [1], [8], [9].

Metoda využívá jevu zvaného koherenční zrnitost. Jde o trojrozměrný interferenční jev. Ozáření dostatečně drsného povrchu dostatečně koherentním zářením zapříčiní vznik trojrozměrného náhodného rozdělení intenzity záření kolem povrchu. Toto rozdělení intenzity záření se nazývá pole koherenční zrnitosti.

Poté se provede záznam řezu koherenční zrnitosti na holografickou desku, který je označován jako specklegram. Z něho je možné po analýze získat informace o statistických vlastnostech povrchu.

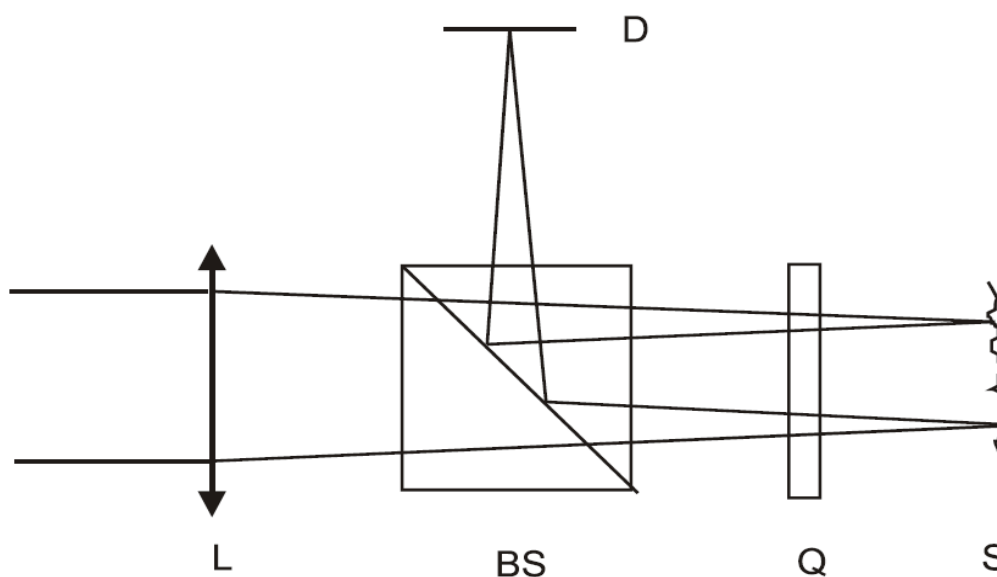


Obr 2.1. Schéma experimentálního uspořádání metody „Úhlová korelace polí koherenční zrnitosti:“
LA - He-Ne laser, SF - prostorový filtr, L - spojná čočka, D a D_1 - clony vymezující laserový svazek, S - studovaný povrch, TG - otočný stolek goniometru, CCD - CCD kamera [1]

2.3 Vyšetřování povrchu pomocí rozdělení intenzity rozptýleného laserového světla

Jedná se o způsob spadající do kategorie fotometrických metod. Principem je rozdělení intenzity rozptýleného laserového světla ve fourierovské rovině studovaných povrchů.

Velikost amplitudy rozptýlené laserové vlny těsně nad povrchem tvoří jeho světelný otisk. Wienerové spektrum amplitud této rozptýlené vlny je tvořeno intenzitou vlny ve fourierovské rovině studovaného povrchu. Autokorelační funkce odpovídá inverzní Fourierově transformaci Wienerova spektra. Tedy z naměřené intenzity ve fourierovské rovině se spočítá rozdělení střední intenzity. Pomocí počítače se provede její inverzní fourierovská transformace a dostaneme experimentální odhad autokorelační funkce amplitudy vlny nad povrchem. Z této funkce lze pomocí metody nejmenších čtverců získat námi požadované charakteristiky povrchu. Velkou výhodou této metody je získání výsledků již po jednom měření. [1], [7]



Obr.2.2 Schéma experimentálního uspořádání metody „Měření drsnosti povrchu z rozdělení intenzity rozptýleného laserového světla“ : L – spojná čočka, BS – dělič svazku, Q – čtvrtvlnná destička, S – studovaný povrch, D – detektor (fotodioda) [1]

2.4 Vyšetřování povrchu pomocí stříhové interferometrie

Nyní se podíváme poněkud podrobněji na tuto optickou metodu. Ještě podrobnější popis naleznete v pracích [5], [6]. Metoda stříhové interferometrie patří do skupiny interferenčních metod, které jsou založené na tvaru interferenčních proužků. Pomocí této metody jsme schopni určit střední kvadratickou odchylku výšek nerovností σ , střední kvadratickou odchylku sklonů povrchu σ_1 , autokorelační vzdálenost T , autokorelační koeficient $C(\tau)$, autokorelační funkci sklonů povrchu $C_1(\tau)$, autokorelační funkci povrchu $B(\tau)$ a autokorelační funkci derivací $B_1(\tau)$. Tuto metodu můžeme využít jak pro vzorky neprůhledné (pomocí Zeiss Peraval Interphako interferenčního mikroskopu), tak pro průhledné (pomocí Zeiss Epival Interphako interferenčního mikroskopu). Výsledky dosahované touto metodou se velmi blíží výsledkům získaných pomocí kontaktního profilometru. Velkou výhodou této metody je spolehlivost a relativní jednoduchost ve srovnání s jinými optickými metodami. [1], [2]

V této kapitole si nejdříve upřesníme předpoklady, které musí být splněny pro užití této metody, poté přejdeme k samotnému principu měření charakteristik a na závěr vše shrneme.

2.4.1 Vlastnosti studovaného náhodně drsného povrchu

Níže je uveden přehled předpokladů specifikující model náhodně (statisticky) drsného povrchu, jenž by měl být splněn při studiu drsného (lokálně hladkého) transparentního nebo odrazného, rovinného nebo zakřiveného povrchu při použití metody stříhové interference. Tyto předpoklady vychází z [2], [10], [11], [12].

Vymezení modelu náhodně drsného povrchu:

- (i) Pro derivaci funkce $Z(x,y)$ popisující střední hodnotu náhodně drsného povrchu musí platit:

$$\left(\frac{\partial Z(x,y)}{\partial x}\right)^2 + \left(\frac{\partial Z(x,y)}{\partial y}\right)^2 \ll 1, \quad (2.10)$$

pro každý bod (x,y) , kde x a y jsou souřadnice Kartézského souřadnicového systému.

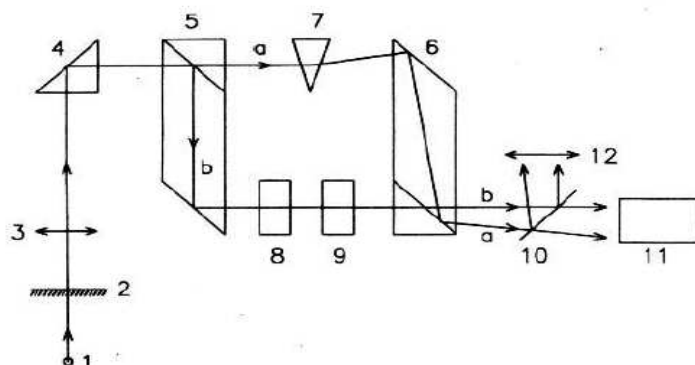
- (ii) Lineární rozměry osvětlené části povrchu jsou mnohem větší než vlnová délka λ dopadajícího světla.
- (iii) Drsnost povrchu je vygenerována nějakým stacionárním izotropním normálním (Gaussovým) náhodným procesem.
- (iv) Střední hodnota náhodné funkce $\zeta[x,y,Z(x,y)]$ popisující drsnost povrchu je nula v každém bodu povrchu.
- (v) Drsný povrch je místně hladký, tedy v každém bodě povrchu se tečná rovina jen mírně odchyluje od tohoto povrchu na oblasti lineárních rozměrů, jenž jsou mnohem větší než λ .
- (vi) Spády nerovností na povrchu musí být natolik malé, aby mnohonásobné odrazy a stíny mezi těmito nerovnostmi nemusely být uvažovány.
- (vii) Materiál tvořící medium po obou stranách drsného povrchu musí být z optického hlediska izotropní a homogenní.
- (viii) Jde-li o odrazující povrch (tedy světlo odražené), pak pro střední kvadratickou odchylku nerovností výšek povrchu σ musí platit: $\sigma < \lambda$. V případě transparentních povrchů (tedy světlo prošlé) musí platit: $\sigma < \lambda$, $\sigma \sim \lambda$, $\sigma > \lambda$ nebo s použitím imersní stříhové interferometrie $\sigma \gg \lambda$.

2.4.2 Základní princip stříhové interferometrie

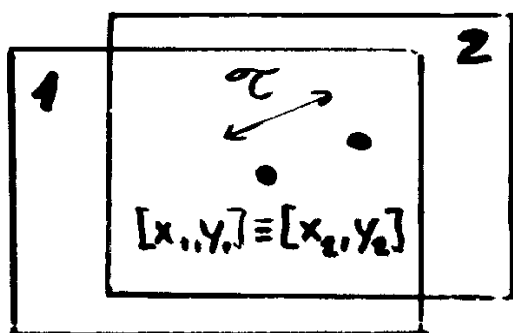
Metoda využívá komerčně dostupných interferenčních mikroskopů. V laboratoři na Ústavu fyzikálního inženýrství Vysokého učení technického v Brně byl použit interferenční mikroskop Zeiss Peraval Interphako pro propustné povrchy transparentních látek a interferenční mikroskop Zeiss Epival Interphako pro odrazné povrchy transparentních i netransparentních látek.

Schéma Zeiss Peraval Interphako (ZPI) interferenčního mikroskopu pro světlo prošlé je zobrazeno na obrázku 2.3, kde jednobarevné světlo tvořené wolframovou lampou s filtrem pro vlnovou délku 578 nm (1) projde skrz studovaný vzorek (2). Vznikne „světelný otisk“ povrchu, který nese informaci o drsnosti ve své fázi. Světelná monochromatická vlna pokračuje směrem k čočce (3) a k pomocnému hranolu (4). Odtud vstupuje do dvou-paprskového interferometru (Mach-Zehnerova interferometru), kde se rozdělí na dvě vlny. Interferometr je tvořen dvěma optickými hranoly (5, 6) a

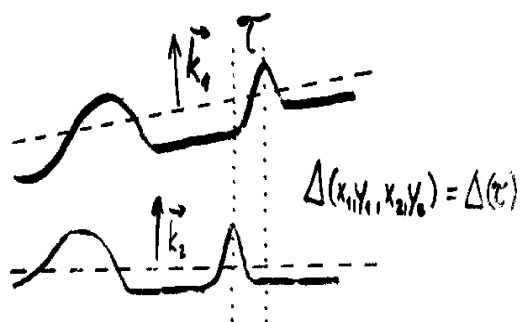
třemi klíny (7, 8, 9). Velikost úhlu klínu 7 se dá měnit. Pomocí klínů 7-9 můžeme získat dvě vlnoplochy odpovídající dvěma paprskům a a b. Vlnoplochy jsou vzájemně stranově posunuté a vychýlené viz obrázek 2.4. Vznikne mezi nimi jistý příčný stříh. Velikost tohoto stříhu τ je vzdálenost odpovídajících si bodů vlnoploch. Velikost stříhu τ je možno měnit pomocí interferenčního mikroskopu. Vlnoplochy navzájem interferují a vzniká interferogram se světlými a tmavými proužky jako na obrázku 2.5. Vzniklý interferogram lze souběžně sledovat v okuláru 12 a zaznamenávat pomocí CCD kamery 11. [1], [2]



Obr. 2.3 Schematické uspořádání Zeiss Peraval Interphako (ZPI) interferenčního mikroskopu: 1, zdroj světla ($\lambda=578$ nm); 2, vzorek; 3, čočka objektivu; 4, pravouhlý hranol; 5, 6, optické hranoly; 7, klín s nastavitelným úhlem; 8, 9, klíny interferometru; 10, dělič paprsku; 11, CCD kamera; 12, okulár [1]

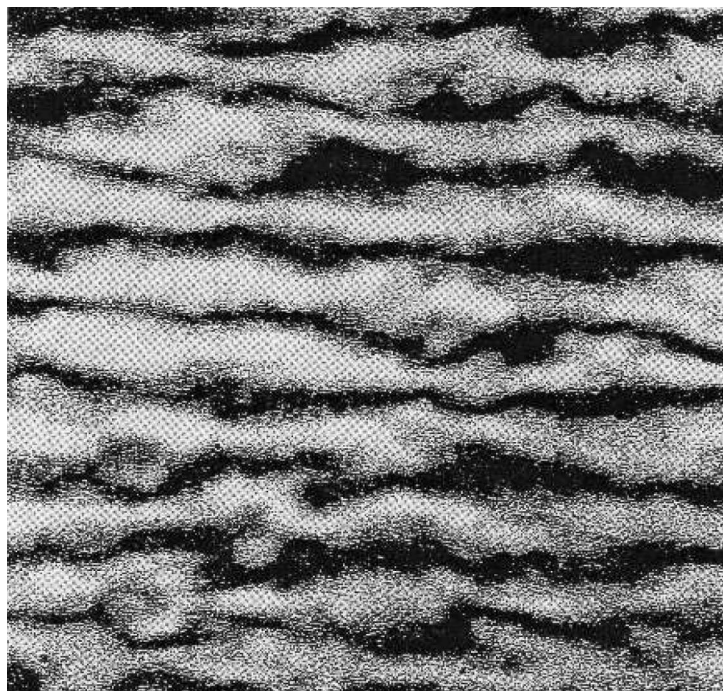


a)



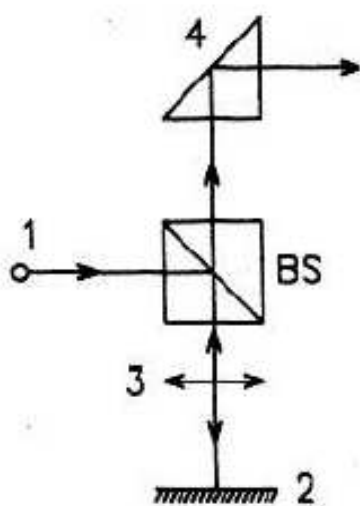
b)

Obr. 2.4 a) Schematický „čelní pohled“ na výstupu ZPI nebo ZEI; b) Schematický „boční pohled“ na výstupu ZPI nebo ZEI [2]



Obr.2.5 Interferogram vzorků skleněných matnic připravených jednostranným leptáním skleněných destiček ve směsi HF a HO v poměru 2:3.[2]

Zeiss Epival Interphako interferenční mikroskop, který se používá pro metodu ve světle odraženém u neprůhledných materiálů se odlišuje od ZPI mikroskopu jen v oblasti sousedící s vyšetřovaným povrchem jak je znázorněno na Obr. 2.6.



Obr. 2.6 Schematické uspořádání části Zeiss Epival interphako (ZEI) interferenčního mikroskopu mezi vzorkem a dvou-paprskovým interferometrem: 1, zdroj světla ($\lambda=578$ nm); 2, vzorek; 3, čočka objektivu; BS, optický hranol; 4, pravoúhlý hranol (zbytek ZEI mikroskopu je identický s ZPI mikroskopem) [2].

Jak již bylo řečeno, výstupem celé procedury je záznam interferogramu pomocí CCD kamery. Interferogram je tvořen tmavými a světlými proužky jako na obrázku 2.5. Odtud je také vidět, že interferenční proužky jsou tvarově deformované. Matematickým rozbořem v pracích [1], [10], [11] bylo dokázáno, že deformace proužků vystihuje veličina h o rovnici:

$$h = \xi(x, y) - \xi(x + \tau, y), \quad (2.11)$$

kde $\xi(x, y)$ popisuje výšku povrchu v bodě $[x, y]$, je-li stříh proveden podél osy x .

Najdeme-li středy temných a světlých proužků, dostaneme linie podobné jako je tomu na obrázku 2.7. Ve výše uvedených pracích jsou odvozeny také vztahy (2.12) a (2.13) mezi veličinami h , q a p , kde q je střední vzdálenost mezi středy sousedních proužků a p je místní odchylka linie (představující střed interferenčního proužku) od své průměrné hodnoty, n_0 je index lomu vnějšího prostředí (nad povrchem) a n je index lomu průhledného studovaného vzorku. Na obrázku 2.8 je význam těchto veličin schématicky znázorněn.

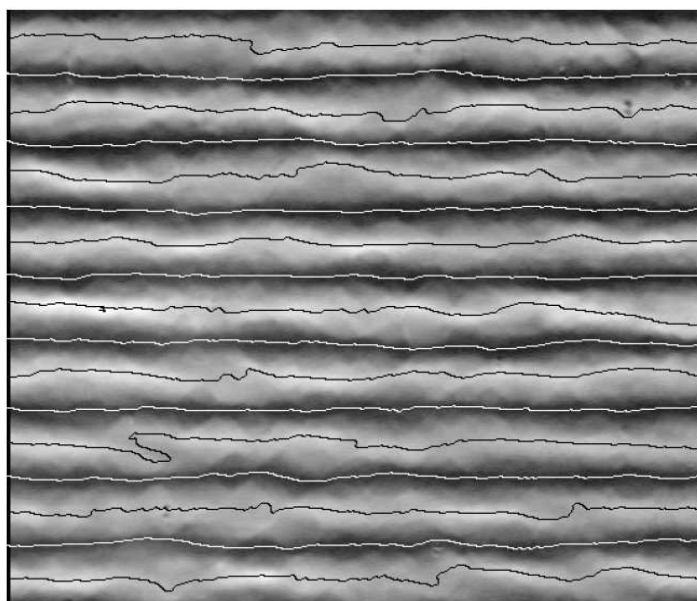
$$h = \frac{\lambda}{n_0 - n} \frac{p}{q} \quad \text{pro světlo prošlé} \quad (2.12)$$

$$h = \frac{\lambda}{2n_0} \frac{p}{q} \quad \text{pro světlo odražené} \quad (2.13)$$

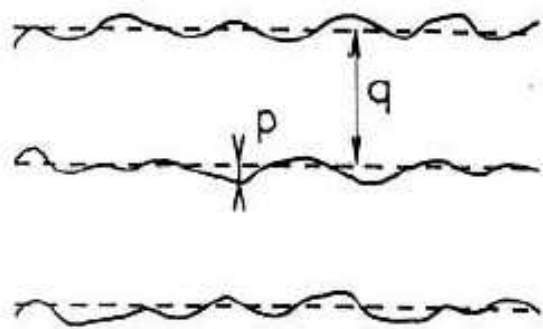
Ze vztahu (2.12) a (2.13) je vidět, jak lze snadno přepočítat hodnoty veličiny p_i charakterizující středy proužků na veličinu h_i . Z vypočítaných hodnot h_i lze pomocí rovnice (2.14) získat střední kvadratickou odchylku σ_h .

$$\sigma_h = \sqrt{\frac{\sum h_i^2}{M}}, \quad (2.14)$$

kde suma nabývá hodnot od jedné do M a M je počet všech bodů linií představující středy temných a světlých proužků interferogramu pro určité τ .



Obr. 2.7 Interferogram s vyznačenými středy temných a světlých proužků [1]



Obr. 2.8 Schematické znázornění významu veličin p a q . Plná čára představuje středy interferenčních proužků a čárkovaná linie jejich střední hodnotu [13].

Vytvoříme-li pomocí interferenčního mikroskopu několik interferogramů pro různé hodnoty τ , získáme závislost $\sigma_h(\tau)$. Ta je potřebná k určení statistických charakteristik zkoumaného povrchu. Dále pokračujeme podle postupů uvedených ve [1], [2], [13]. Obecně platí rovnice:

$$\sigma_h(\tau) = \sigma \sqrt{2(1 - C(\tau))} \quad , \quad (2.15)$$

kde σ je střední kvadratická odchylka funkce $\zeta(x,y)$ a $C(\tau)$ je autokorelační koeficient této funkce.

Vytvořená závislost $\sigma_h(\tau)$ ze souboru interferogramů o různých hodnotách τ může buď vypadat jako na obrázku 2.9 nebo 2.10.

První případ nastane, když metodou stříhové interferometrie nelze získat závislost $\sigma_h(\tau)$ až do oblasti saturace. To je způsobeno tím, že se kontrast interferenčních proužků silně redukuje pro vyšší hodnoty τ . Musíme tedy tvar funkce $C(\tau)$ postulovat. Tomu nejlépe vyhovuje Gaussův tvar:

$$C(\tau) = \exp\left(-\frac{\tau^2}{T^2}\right) \quad , \quad (2.16)$$

kde T je autokorelační vzdálenost.

Poté použijeme metodu nejmenších čtverců na rovnici (2.15), kde využijeme závislosti (2.16) pro zjištění hodnoty T a σ .

Nastane-li případ jako na obrázku 2.10 (metodou stříhové interferometrie můžeme získat závislost $\sigma_h(\tau)$ až do oblasti saturace), nalezneme jistou hodnotu stříhu τ^* , kde pro všechny $\tau > \tau^*$ již závislost $\sigma_h(\tau)$ téměř neroste. Ze vztahu (2.15) a zjištěného průběhu závislosti $\sigma_h(\tau)$ je zřejmé, že pro $\tau > \tau^*$ platí:

$$C(\tau) = 0 \quad (2.17)$$

To tedy znamená, že pro $\tau > \tau^*$ ze vztahu (2.15) plyne:

$$\sigma = \frac{\sigma_h(\tau)}{\sqrt{2}} \quad . \quad (2.18)$$

Známe-li hodnotu σ , můžeme určit průběh funkce $C(\tau)$ takto:

$$C(\tau) = 1 - \frac{\sigma_h^2(\tau)}{2\sigma^2} \quad (2.19)$$

Z průběhu $C(\tau)$ lze zjistit velikost autokorelační vzdálenosti T , pomocí její definice (2.8) a vztahu (2.9).

Nyní již známe velikosti veličin σ , T a závislost $C(\tau)$ pro oba případy možných závislostí $\sigma_h(\tau)$. Dále již postup pro určení dalších charakteristik je pro obě varianty stejný. Určíme $B(\tau)$ ze vztahu (2.9), pak můžeme určit také autokorelační funkci derivací $B_1(\tau)$ funkce $\zeta(x,y)$ takto:

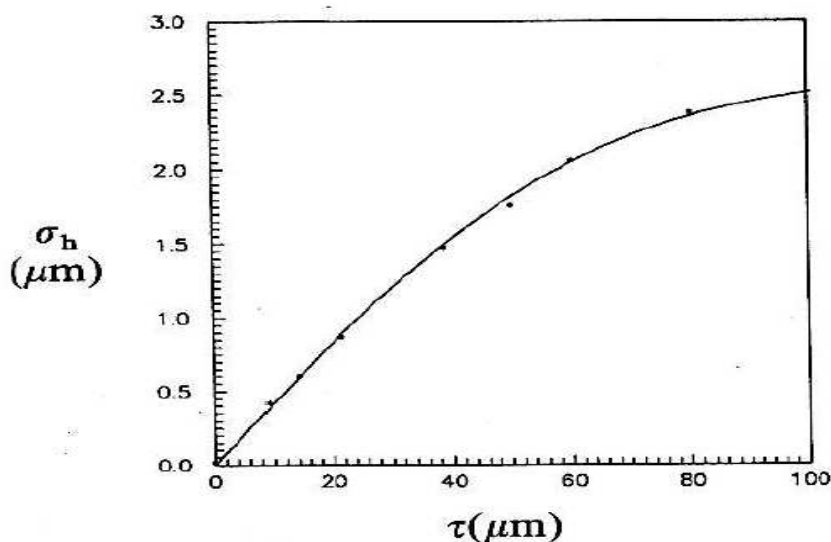
$$B_1(\tau) = \frac{-d^2 B(\tau)}{d\tau^2} \quad (2.20)$$

Z $B_1(\tau)$ určíme střední kvadratickou odchylku sklonů povrchu σ_1 rovnicí:

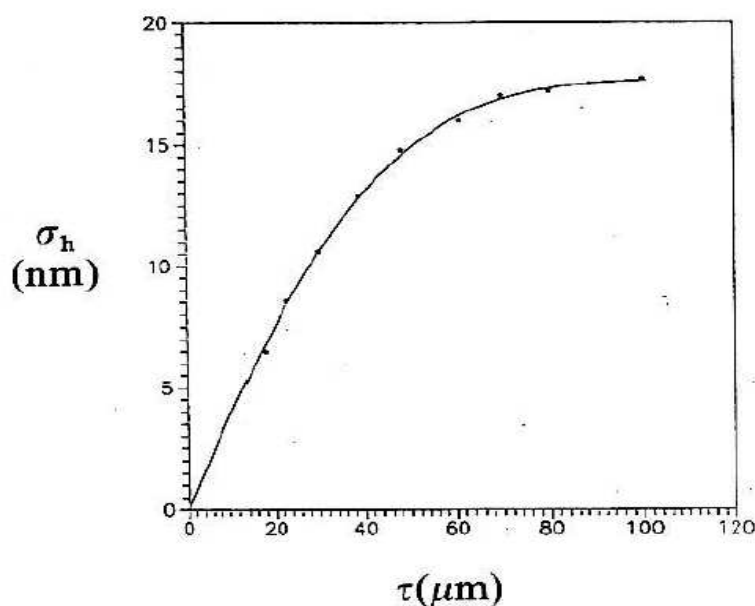
$$\sigma_1 = \sqrt{B_1(0)} \quad (2.21)$$

Nakonec určíme autokorelační funkci sklonů povrchu $C_1(\tau)$ rovnicí:

$$C_1(\tau) = \frac{B_1(\tau)}{\sigma_1^2} \quad (2.22)$$



Obr. 2.9 Závislost velikosti veličiny σ_h na velikosti stříhu τ pro vzorky skleněných matnic připravených jednostranným leptáním skleněných destiček ve směsi HF a HO₂ v poměru 2:3.



Obr 2.10 Závislost velikosti veličiny σ_h na velikosti stříhu τ pro vzorky parabolických světelných reflektorů.

2.4.3 Závěrem k metodě stříhové interferometrie

Z předchozí podkapitoly je zřejmé, že metoda stříhové interferometrie může být použita, jestliže výstupní interferogram z interferenčního mikroskopu je tvořen deformovanými proužky pro něž platí $p_i < q$. Tato nerovnost bývá závislá na velikosti střední kvadratické odchylky funkce $\zeta(x,y)$ a na autokorelační vzdálenosti zkoumaného povrchu, tedy na σ a T . Horní hranice použitelnosti této

metody pro σ je $\sigma_{max} \approx \frac{\lambda}{6\sqrt{2}n_0}$ pro odražené světlo a $\sigma_{max} \approx \frac{\lambda}{3\sqrt{2}|n_0 - n|}$ pro světlo prošlé.

Spodní hranice σ_{min} je závislá na použité stavbě interferenčního mikroskopu a na vlastnostech jeho jednotlivých částí. Horní hranice T_{max} použitelnosti metody je dána zvětšením použitého mikroskopu, dolní hranice T_{min} je dána vlastnostmi interferenčního mikroskopu obdobně jako σ_{min} . [2]

Metoda stříhové interferometrie je určena pro zjišťování základních statistických charakteristik drsnosti průhledných i neprůhledných povrchů a zakřivených ploch vzniklých stacionárním izotropním normálním (Gaussovým) náhodným procesem. Metoda je založena na průchodu nebo odrazu monochromatického světla zkoumaným vzorkem a následném zpracování tohoto světla interferenčním mikroskopem. Ten monochromatickou vlnu rozdělí na dvě vlny a pak je nechá interferovat. Interferující vlny jsou navzájem odkloněné a posunuté o velikost stříhu τ , jenž se může měnit. Výstupem tohoto mikroskopu je tzv. interferogram zaznamenaný na CCD kameru. Interferogram obsahuje interferenční proužky, ty se dále zpracovávají tak, aby se našli jejich středy, které tvoří jednopixelové linie. Z tvarů těchto linií se vypočte veličina σ_h . Takto se získají různé hodnoty σ_h z interferogramů vzniklých různým nastavením velikosti stříhu τ pomocí interferenčního mikroskopu. Ze závislosti σ_h na τ se dají prostřednictvím známých postupů získat požadované charakteristiky zkoumaného povrchu.

Na úplný závěr této kapitoly bych chtěl shrnout hlavní výhody této metody [1]:

- (i) metoda je nedestruktivní
- (ii) pro tuto metodu se využívají jednoduchá komerční zařízení
- (iii) metoda je relativně jednoduchá oproti jiným optickým metodám

- (iv) metoda může být použita pro získání základních statistických veličin drsnosti jak pro rovinné , tak pro zakřivené plochy
- (v) metoda je založena na nekomplikovaném hledání středů interferenčních proužků a to je výhoda oproti metodám založeným na kontrastu proužků, protože měření kontrastu je komplikován mnoha faktory
- (vi) pro efektivní způsob zpracování se dají využít metody výpočetní techniky
- (vii) metoda nevyžaduje žádné předpoklady o vlastnostech povrchu

3 ALGORITMY ZPRACOVÁNÍ OBRAZU PŘI URČOVÁNÍ TOPOGRAFICKÝCH PARAMETRŮ OBRAZU

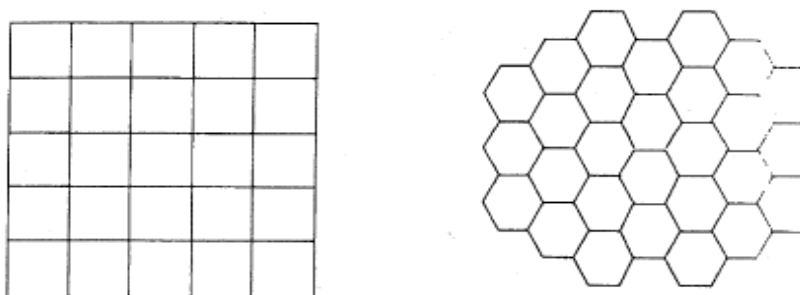
Výstupem interferenčního mikroskopu je interferogram s temnými a světlými proužky, jak je vidět na obrázku 2.5. Ten je zaznamenán na CCD kameru. K jeho přenosu do PC se používá 8-bitový video adaptér. Interferogram se v PC uloží jako černobílý bitmapový obrázek ve 256 odstínech šedi. Tím získáme digitalizovaný obraz zaznamenaný v PC a můžeme ho dále zpracovávat pomocí počítačových programů (např. SOFO DIPS 4.0).

Jedním z úkolů této práce je navrhnout a realizovat postup jak získat statistické údaje ze středů interferenčních proužků interferogramu pomocí známých algoritmů pro zpracování obrazu. V této kapitole si ukážeme některé postupy pro práci s obrazem, které by se mohly hodit pro určení topografických parametrů povrchu pomocí metody stříhové interferometrie.

3.1 Reprezentace digitalizovaného obrazu

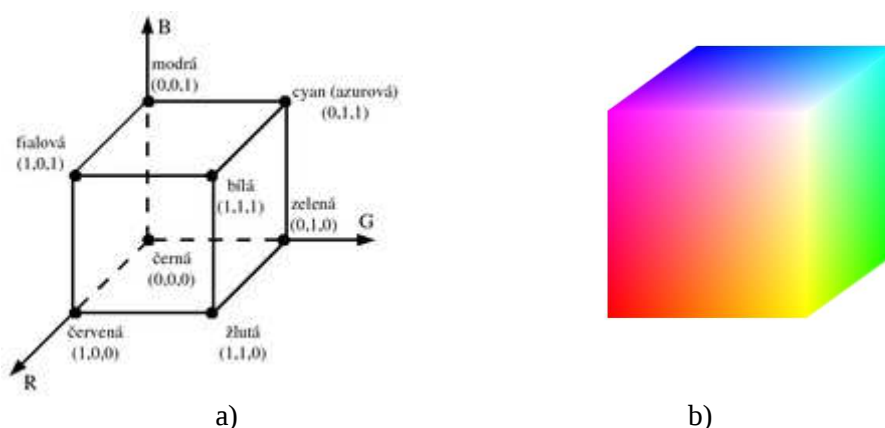
Abychom mohli zpracovat dvojrozměrný obraz pomocí počítače, musíme ho nejdříve získat (např. CCD kamerou), po té digitalizovat (skenerem, pomocí převodního adaptéru...) a uložit do počítače. Pro digitalizaci obrazu musíme použít nějaký vhodný matematický model, kterým bychom obraz v počítači reprezentovali. Jednou z možností je použít tzv. obrazovou funkci $f(i,j)$, která reprezentuje námi sledovanou veličinu obrazu v bodě $[i,j]$. Při zpracování obrazu nabývá obrazová funkce hodnot charakterizující barvu v příslušném bodě. Barvu reprezentujeme zvoleným modelem (např.: RGB, CMY, HSV, HLS...). Obrazovou funkci však nemůžeme reprezentovat v počítači jako spojitou, proto musíme sáhnout k diskretizaci funkce, jinak by nebylo možné obraz digitalizovat. To znamená přiřadit každému bodu diskrétní obrazové funkce $f(i,j)$ hodnotu charakterizující okolí bodu $[i,j]$ (vzorkování a kvantování) reálného obrazu. Bod diskrétní obrazové funkce budeme nazývat pixelem. [16], [18], [19]

Digitalizace sebou přináší také otázku volby velikosti vzorkovacího intervalu (velikost okolí bodu). K vyřešení tohoto problému se používá Shannonova věta o vzorkování. Velikost vzorkovacího intervalu musí být menší nebo roven polovině velikosti nejmenších detailů obrazu. Druhou otázkou digitalizace je volba uspořádání vzorkovacích bodů na ploše (volba vzorkovací mřížky). Používá se mřížka čtvercová nebo hexagonální (Obr. 3.1). Při zpracovávání obrazu je dobré hodnoty diskrétní obrazové funkce reprezentovat v počítači maticí s údaji o barvě jednotlivých pixelů. [15], [16]



Obr. 3.1. Čtvercová a hexagonální mřížka [15]

Pro reprezentaci barvy budeme používat RGB model. Lidské oko vnímá barevný vjem pomocí dráždění 3 barevných receptorů. Tyto receptory jsou u většiny lidí nejcitlivější na podráždění světelného paprsku o vlnových délkách 630 nm (červená barva), 530 nm (zelená barva) a 450 nm (modrá barva). Z těchto barev je také odvozen název modelu (Red-Green-Blue). V RGB modelu každá barva je reprezentovaná třemi složkami (r,g,b) odpovídající intenzitě jednotlivých barev, jak je vidět z jednotkové krychle na obrázku 3.2. Černá odpovídá souřadnicím (0,0,0), bílá (1,1,1). Ostatní základní barvy jsou ve vrcholech krychle ležících na osách. Ve zbývajících rozích jsou doplňkové barvy k základním barvám. Odstíny šedi odpovídají tělesové uhlopříčce z bodu (0,0,0) do (1,1,1). Intenzity barev se pohybují v rozmezí $<0,1>$, ale při počítačovém zpracování se používají hodnoty celých čísel v intervalu $<0,255>$. [18], [19]



Obr. 3.2 Rozložení barev v RGB modelu [19]

3.2 Předzpracování obrazu

Ještě než se pustíme do samotného zpracování digitalizovaného obrazu, provádí se tzv. předzpracování obrazu. Po předchozím kroku máme tedy k dispozici matici s prvky reprezentující barvu jednotlivých pixelů původního obrazu (tzv. ikonická úroveň reprezentace obrazu). Cílem metod uvedených v této kapitole je upravit tuto matici pro potřeby dalšího zpracování, potlačit šum a zkreslení způsobené digitalizací a přenosem obrazu, odstranit zkreslení vzniklé snímacím zařízením nebo potlačit či zvýraznit rysy důležité k dalšímu zpracování. Často se během tohoto kroku zpracování obrazu upravují matice obrazové funkce pro snadnější vyhledání hran v obrazu. Hranou se rozumí dva sousední body obrazu s náhlou změnou obrazové funkce (s vysokou hodnotou velikosti gradientu). [15], [16], [17]

3.2.1 Jasová a kontrastní korekce

Nerovnoměrné osvětlení povrchu nebo nestejná citlivost jednotlivých snímacích bodů CCD kamery mohou způsobit jisté zkreslení digitalizovaného obrazu. Ke zmírnění tohoto nežádoucího účinku se používají jasové transformace. Ta spočívá ve zvýšení všech tří složek RGB modelu o stejnou hodnotu ve všech pixelech obrazu. Je-li výsledná velikost složky větší než 255, nahradí se právě touto hodnotou.

Dále se může provést i korekce kontrastu a to stejným způsobem jako korekce jasu, ale místo přičítání stejné hodnoty ke všem složkám použijeme operaci násobení. [15], [17]

3.2.2 Převod na stupně šedi

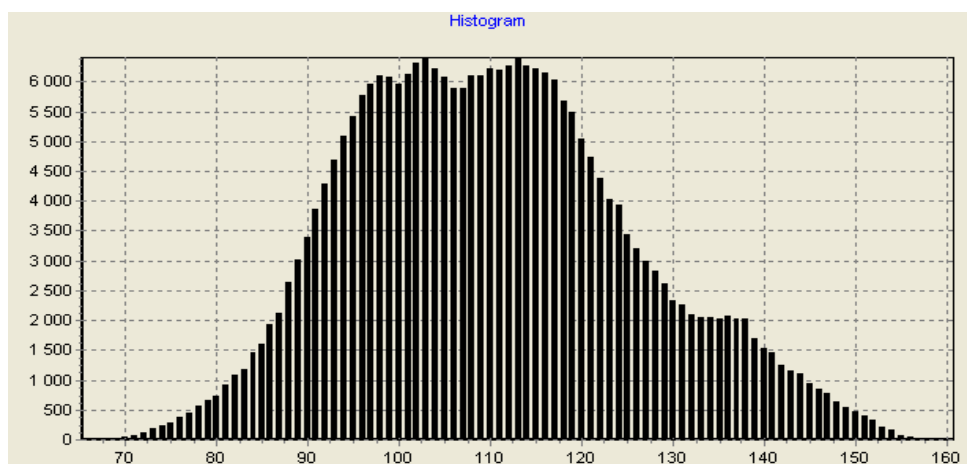
Reprezentace každého pixelu zkoumaného obrazu pomocí tří složek RGB modelu je nepraktická z hlediska dalšího zpracování a zabírá zbytečně mnoho paměti. Nejčastěji se tedy studovaný obraz převede na obraz ve 256 odstínech šedi podle vzorce :

$$I = 0,299r + 0,587g + 0,114b, \quad (3.1)$$

, kde r , g a b jsou jednotlivé složky základních barev RGB modelu a I je výsledná intenzita bílé (s rostoucí intenzitou se sledovaný bod jeví jasněji). Použití jasu je velmi výhodné, protože se dá změřit pomocí čidel (z čidel můžeme získat již černobílý obraz bez použití vzorce (3.1)) a vystihuje podstatu vnímání obrazu člověkem. V dalším textu bude obrazová funkce nabývat hodnot jasu místo trojice barev RGB modelu. [16], [17], [18]

3.2.3 Ekvalizace histogramu

Histogram je jedna z nejdůležitějších vlastností obrazu. Dává nám přehled o tom, jak často se jednotlivé hodnoty jasu v obraze vyskytují. Jde o vektor H se složkami $(j_k, j_{k+1}, j_{k+2}, \dots, j_{k+n})$, kde n udává počet jasových úrovní vyskytujících se v obraze a k hodnotu té nejnižší z nich. Hodnota složky j_i udává počet bodů v obraze s jasnem rovným i . Příklad histogramu je na obrázku 3.3, ze kterého vyčteme například četnost výskytu hodnoty jasu rovné 90, která činí necelé 3500.



Obr. 3.3. Ukázka histogramu

Metoda ekvalizace histogramu se používá k vyvážení jasu v obraze pro přeexpanované fotografie. Pro ně totiž hodnoty jasu v pravé části histogramu převažují. Cílem této metody je vytvořit tzv. ekvalizovaný histogram přesunem některých jasových úrovní do levé části histogramu. Tím se zvýší kontrast obrázku a vystoupí některé detaily. Vznikne tedy histogram, kde četnost jednotlivých jasových úrovní je přibližně stejná.

Jde nám o nalezení funkce, která by zobrazila interval jasů $\langle p_0; p_n \rangle$ do nového intervalu $\langle q_0; q_n \rangle$, kde p_0 je nejnižší a p_n je nejvyšší hodnota jasu vyskytujících se v původním obraze s histogramem $H(p)$ a $\langle q_0; q_n \rangle$ představuje interval všech možných jasových úrovní dané obrazové funkce umožňující daná barevná hloubka obrazu.

Výstupem této metody bude rovnoměrně rozložený histogram $G(q_0, q_1, q_2, \dots, q_n)$ po celém výstupním intervalu $\langle q_0; q_n \rangle$. Velikost jednotlivých složek vektoru $G(q_0, q_1, q_2, \dots, q_n)$ je dána vzorcem:

$$q_i = \frac{q_n - q_0}{N \cdot M} \sum_{i=p_0}^{p_i} H(i) + q_0 \quad \text{pro } i=0,1,2,\dots,n, \quad (3.2)$$

kde N a M jsou výška a šířka obrazu v pixelech a ostatní proměnné jsou totožné s předchozím výkladem. [15], [17], [20]

3.2.4 Filtrace obrazu

Digitalizovaný obraz může být zatížen nějakými lokálními poruchami, které mohou být způsobeny mnoha vlivy od nestejné citlivosti jednotlivých snímacích bodů CCD kamery přes nečistoty na vzorku až po náhodný šum způsobený přenosem z CCD kamery do PC nebo digitalizací obrazu. Právě ke snížení těchto parazitních jevů nacházející se v obraze se používají metody zvané filtrace. Jde o algoritmus, který mění hodnoty jasu jednotlivých pixelů původního obrazu, aby se dosáhlo potlačení výše zmíněných nežádoucích parazitních jevů. Během filtrace dochází k potlačení vysokých frekvencí obrazu a obraz se jeví jako vyhlazený. Nevýhodou je potlačování náhlých změn jasu (hrany). [15], [16], [17]

Mezi 4 nejčastěji používané filtry patří:

- lineární filtrace
- filtrace mediánem
- filtrace pomocí rotující masky
- filtrace Gaussovým filtrem

Lineární filtrace

Principem je použití diskretní konvoluce. Vytvoří se tzv. konvoluční jádro ve tvaru čtvercové matice s lichými rozměry. Tato matice je vždy násobena převrácenou hodnotou součtu všech prvků matice. Ukázkou konvolučního jádra je matice h v rovnici (3.3).

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.3)$$

Konvoluce upraví původní matici jasu $g(i,j)$ na matici jasu $f(i,j)$ při použití konvolučního jádra $h(i,j)$ o rozměru $N \times N$ (N je liché přirozené číslo) podle rovnice:

$$f(i, j) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} h(m, n) g(i - (N-1)/2 + m, j - (N-1)/2 + n) \quad (3.4)$$

Použijeme-li jádro z rovnice (3.3), tak se jedná o tzv. filtraci průměrováním (změna zastoupení jasu je patrná z histogramů na obrázku 3.4). Jde o velice jednoduchý filtr. Lepší vlastnosti však vznikají použitím filtru, kde je váha středu jádra největší a se vzdáleností od něj klesá. Příklady tokových filtrů typu dolní propust jsou zde:

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad h = \frac{1}{12} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

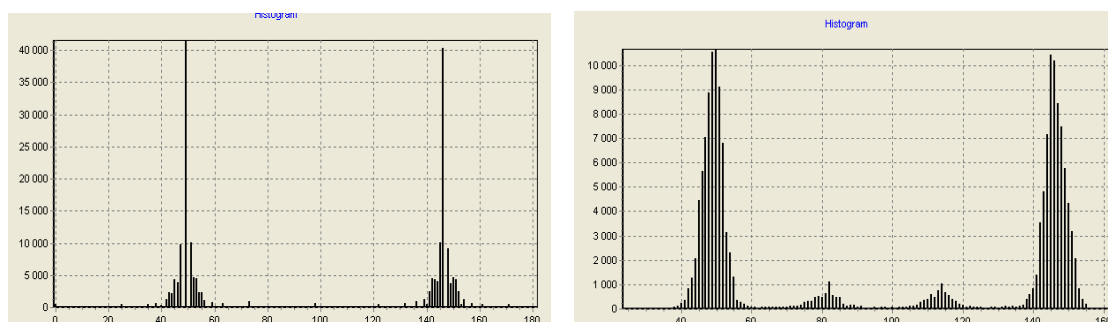
$$h = \frac{1}{112} \begin{bmatrix} 1 & 3 & 6 & 3 & 1 \\ 3 & 6 & 9 & 6 & 3 \\ 6 & 9 & 12 & 9 & 6 \\ 3 & 6 & 9 & 6 & 3 \\ 1 & 3 & 6 & 3 & 1 \end{bmatrix} \quad h = \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

Masky typu horní propust vypadají například takto:

$$h = \frac{1}{10} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad h = \frac{1}{10} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Tento druh filtrů odstraňuje složky nízkých frekvencí.

Volba použitého filtru a jeho rozměry jsou závislé na velikosti lokálních chyb a objektech obrazu. Tato metoda je velice jednoduchá, ale její nepříjemná vlastnost je rozmazávání hran. [15], [17], [20], [21]



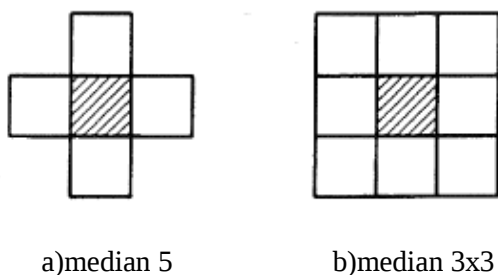
Obr. 3.4 Histogram obrazu před a po filtraci průměrováním

Filtrace mediánem

Metoda spočívá v nahrazení jasu všech pixelů obrazu mediánem jasu v okolí zpracovávaného pixelu.

Nejdříve si zvolíme tvar okolí jaké budeme používat. Nejčastěji se používá sousední 4 pole (pro vertikální a horizontální směr) nebo 8 polí (pro všech osm směrů z daného bodu), jak ukazuje obrázek 3.5. Pak pro každý bod obrazu seřadíme hodnoty obrazových funkcí v jeho okolí podle velikosti. Zařadíme do právě vytvořené posloupnosti velikost jasu zkoumaného bodu. A nakonec hodnotu obrazové funkce v tomto bodě nahradíme mediánem (prostřední hodnotou) námi vytvořené monotonní posloupnosti.

Tato metoda je představitelem nelineárních filtrací, redukuje stupeň rozmazání hran a impulsní šum. Bohužel však porušuje tenké čáry a ostré rohy. [15], [17], [20], [21]



Obr.3.5. Okolí používané pro filtraci mediánem [15]

Filtrace pomocí rotující masky

Jde o iterační metodu. Hledá ve svém okolí bod, který nejpravděpodobněji patří do stejné oblasti (má nejbližší hodnotu obrazové funkce k vyšetřovanému bodu). Z těchto dvou bodů se vypočte průměr a dosadí se do vyšetřovaného bodu. Postup se neustále opakuje dokud se obraz mění nebo dokud se neprovede určený počet iterací.

Rotující maska snižuje rozmazávání hran a má ostřicí účinky. [15], [17], [20], [21]

Filtrace Gaussovým filtrem

Pro tuto filtraci se používá konvoluce s maskou, která se skládá z hodnot daných Gaussovou funkcí:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \quad (3.5)$$

kde σ je střední kvadratická odchylka okolí bodu. Tento filtr je efektivní k potlačení šumu a vede k rozmazání obrazu. [17], [20], [21]

3.2.5 Ostření obrazu

Metoda se používá ke zvýšení strmosti hran, tedy ke zdůraznění vysokých frekvencí. Člověku potom připadá obraz více ostrý, zejména hrany vystoupí do popředí. Hodnoty obrazové funkce pro všechny body obrazu po ostření se vypočte z rovnice:

$$f(i,j) = g(i,j) - c \cdot S(i,j), \quad (3.6)$$

kde $f(i,j)$ je hodnota nové obrazové funkce v bodě $[i,j]$, $g(i,j)$ je hodnota původní obrazové funkce ve stejném bodě, c je kladný součinitel udávající sílu ostření a $S(i,j)$ je operátor udávající strmost změny obrazové funkce f . $S(i,j)$ je dána velikostí gradientu, určuje se například pomocí Laplaceova nebo Robertsova operátoru. [15], [16], [17]

3.3 Segmentace obrazu

Segmentací se rozumí nalezení objektů v obraze. Objekty rozumíme části obrazu, které jsou zajímavé z hlediska dalšího zpracování obrazu. Nejjednodušší metodou segmentace je tzv. prahování. To rozdělí obraz na několik ploch, které spolu nějakým způsobem souvisí. Tyto plochy jsou od sebe odlišeny hodnotou obrazové funkce. Vzniklé plochy nemusí být souvislé a potom tvoří soubor objektů s podobnými vlastnostmi. Tyto objekty je pro další jejich zpracování dobré odlišit hodnotou obrazové funkce (barvami). K tomu poslouží tzv. barvení.

Druhou možností segmentace je metoda hledání hran a jejich následné propojování, abychom dostali hranice objektu.

Třetí skupinou plnicí vlastnosti segmentace je tzv. narůstání oblastní. Sem patří například metoda semínkového vyplňování. [15], [16], [17], [21]

3.3.1 Prahování

Jde o velice jednoduchý a rychlý algoritmus vedoucí k rozdělení obrazu na dvě nebo více oblastí. Každá oblast bude odpovídat jedné jasové úrovni. Tyto oblasti se stejnou jasovou úrovní by spolu měly nějak souviset. Prahování lze využít u obrazů, kde se od sebe jednotlivé objekty s podobnými vlastnostmi dostatečně odlišují od ostatních objektů hodnotou obrazové funkce. Základní myšlenka této metody je velice jednoduchá. Dva pixely patří do stejné oblasti, právě když je velikost obrazové funkce (dána jasem pixelu) v těchto dvou bodech podobná. To je způsobeno tím, že většina objektů ve studovaném obraze má konstantní odrazivost nebo pohltivost povrchu.

Algoritmus pracuje tak, že se stanoví intervaly obrazové funkce pro každý objekt obrazu. Pixel do určitého objektu patří, právě když jeho hodnota obrazové funkce se nalézá ve stanoveném intervalu. Rozhraní mezi intervaly se nazývají prahy. Po té prochází každý pixel a přiřadí mu novou hodnotu obrazové funkce $f(i,j)$ takto:

$$\begin{aligned} f(i,j)=1 & \Leftrightarrow g(i,j) \text{ leží v } \langle 0;k_1 \rangle \\ f(i,j)=2 & \Leftrightarrow g(i,j) \text{ leží v } \langle k_1;k_2 \rangle \\ & \dots\dots\dots \\ f(i,j)=n & \Leftrightarrow g(i,j) \text{ leží v } \langle k_{n-1};k_n \rangle, \end{aligned}$$

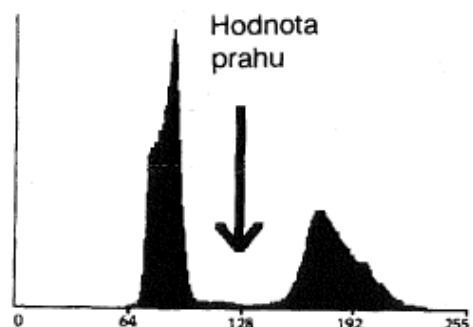
kde $g(i,j)$ jsou hodnoty původní obrazové funkce a k_i jsou prahy jednotlivých objektů.

Tímto postupem se od sebe odliší skupiny objektů s podobnými vlastnostmi, kde každá skupina má svoji specifickou hodnotu obrazové funkce. Problém je však stanovit hodnoty prahů. Ty se mohou určit interaktivně ve spolupráci s uživatelem, nebo automaticky.

Na určení prahů se používají například procentní prahování nebo analýza tvaru histogramu. Procentní prahování lze využít, pokud víme jaký poměr zaujímají jednotlivé objekty v obraze. Po té stanovíme prahy pomocí histogramu tak, aby první interval obsahoval právě tolik pixelů jako je velikost nejtmavšího objektu v obraze, druhý obsahoval tolik pixelů jako je velikost druhého nejtmavšího objektu atd.

Stanovení prahů pomocí analýzy tvaru histogramu se používá, má-li histogram podobný průběh jako je na obrázku 3.6. Nachází-li se zde tolik „kopců“ jako je objektů v obraze, pak se určí prahy mezi jednotlivými objekty jako hodnota obrazové funkce s minimální četností mezi dvěma „kopci“. Na obrázku 3.6 je tento práh vyznačen pro oddělení dvou objektů v obraze.

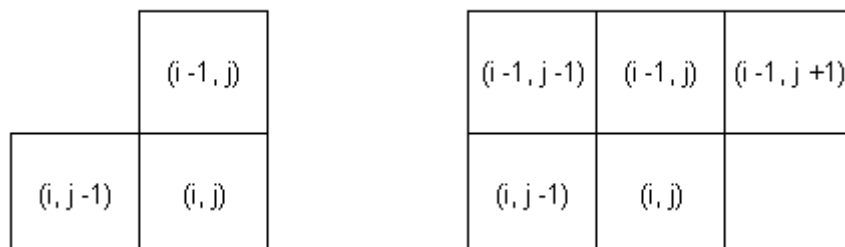
Pokud konstantní stanovení prahů nevedou k požadované segmentaci, lze volit práh dynamicky a hodnotu prahu počítat podle lokálních hodnot obrazové funkce. [15], [16], [17]



Obr. 3.6 Bimodální histogram s vyznačeným prahem [15]

3.3.2 Barvení

Tento algoritmus slouží k barevnému oddělení objektů, které po prahování patří do jedné skupiny (jejich pixely mají podobnou hodnotu obrazové funkce). Pro jednoduchost budeme předpokládat, že jsme v předchozím kroku rozdělili pixely obrazu do dvou skupin. První skupina s hodnotou obrazové funkce $g = 0$ tvoří pozadí. Druhá skupina tvoří objekty a byla jí přidělena velikost obrazové funkce $g = 1$. Za jednotlivé objekty budeme považovat souvislou oblast pixelů mající hodnotu obrazové funkce rovnou jedné. Algoritmus prochází postupně pixely po řádcích a narazí-li na nějaký s hodnotou obrazové funkce $g=1$, tak se mu přiřadí hodnota (barva) podle hodnot obrazové funkce jeho okolí. Okolí je dáno maskou. Nejčastěji se používají masky zobrazené na obrázku 3.7.

Obr. 3.7 Masky určující okolí bodu $[i,j]$ pro barvení [17]

Algoritmus se skládá ze dvou průchodů:

1. průchod – Postupně prohledáváme po řádcích obraz, dokud nenajdeme bod s hodnotou obrazové funkce $g(i,j)=1$. Tomu se přiřadí nová hodnota obrazové funkce $f(i,j)$ podle hodnot jeho sousedů, kteří se určí ze zvolené masky takto:
 - Jestliže všechny sousední body mají hodnotu obrazové funkce rovnou 0, tak se tomuto bodu přiřadí nová dosud nepřiřazená barva.
 - Existuje-li jen jeden soused s nenulovou hodnotou obrazové funkce, pak se zkoumanému bodu přiřadí barva právě tohoto souseda.
 - Existuje-li více sousedů s nenulovou hodnotou obrazové funkce, tak se zkoumanému bodu přiřadí barva libovolného nenulového souseda. Pokud existují dva nebo více sousedů (Obr. 3.8), kteří mají různou barvu, tak se tyto dvojice poznamenají do tabulky ekvivalence barev.
2. průchod – Nyní jsou všechny pixely ze skupiny objektů obarveny nějakou barvou. Některé objekty jsou však obarveny více barvami (kolize barev), proto se obraz projde ještě jednou a přebarví se podle informací v tabulce ekvivalence barev.

Tím jsme dostali obraz, kde jsou jednotlivé objekty a pozadí odlišeny barvou. To je velice výhodné pro další zpracování, protože nyní o každém pixelu budeme vědět k jakému objektu patří. [15], [17]

1		2
1		2
1	1	?

Obr. 3.8 Kolize barev. [15]

3.3.3 Hledání hran

Hrana v obraze je v podstatě vlastnost pixelu a jeho okolí. Jde o vektor, který má samozřejmě svůj směr a velikost. Směr udává úhel největšího spádu a velikost vektoru rychlost s jakou se v daném pixelu obrazová funkce mění. Psychofyziologické experimenty dokázaly, že hrany jsou velmi důležité pro člověka při rozpoznávání obrazu.

Metody detekce hran jsou založeny na výpočtu gradientu obrazové funkce. V digitalizovaném obrazu musí samozřejmě jít pouze o diskretní náhradu gradientu. Velikost gradientu obrazové funkce $g(x,y)$ se počítá z rovnice:

$$|\nabla g(x,y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} . \quad (3.7)$$

Směr gradientu obrazové funkce $g(x,y)$ je dán rovnicí:

$$\Psi = \operatorname{tg}^{-1} \left(\frac{\frac{\partial g}{\partial y}}{\frac{\partial g}{\partial x}} \right) \quad \text{pro} \quad \frac{\partial g}{\partial x} \neq 0 . \quad (3.8)$$

Při hledání hran však nepracujeme přímo s gradientem, ale využíváme tzv. hranové operátory, které aproximují parciální derivace obrazové funkce $g(i,j)$ diferencemi:

$$\Delta_i g(i,j) = g(i,j) - g(i-n,j) \quad (3.9)$$

$$\Delta_j g(i,j) = g(i,j) - g(i,j-n) , \quad (3.10)$$

kde n označuje vzdálenost mezi dvěma sousedními body v rastru. [15], [16], [17]

Pro nalezení hran pomocí operátorů lze postupovat dvěma směry:

- 1) Použijeme operátory aproximující první derivaci obrazové funkce pomocí diferencí. K tomu nám poslouží diskretní konvoluce. U operátorů invariantních vůči rotaci se používá jen jedna konvoluční maska u ostatních se používá více masek a vybere se ta maska, která zjistí největší spád obrazové funkce v daném bodě.
- 2) Použijeme operátory, které aproximují druhou derivaci obrazové funkce. Hrany se pak nacházejí v místech, kde je hodnota této aproximace nulová.

Operátory založené na aproximaci první derivace

Jak bylo řečeno dříve tyto operátory jsou založeny na diskretní konvoluci. V metodách zpracování obrazu se používají konvoluční masky čtvercového okolí O ve tvaru matic o rozměrech okolí O. Aplikací konvoluční masky $h(m,n)$ na zpracovávaný obraz vzorcem:

$$f(i, j) = \sum_{(m,n) \in O} h(m, n) g(i-m, j-n) \quad (3.11)$$

se nahradí původní obrazová funkce $g(i,j)$ novou funkcí $f(i,j)$, která je aproximací první derivace funkce $g(i,j)$. [16]

1. Robertsův operátor

Tento operátor patří mezi nejstarší, využívá okolí o rozměrech 2x2 pixely, je dost citlivý na šum. Používá dvě konvoluční masky [15], [16]:

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Velikost gradientu je dán rovnicí:

$$|\nabla g(i, j)| = |g(i, j) - g(i+1, j+1)| + |g(i, j+1) - g(i+1, j)| \quad (3.12)$$

2. Operátor Prewittové

Tento operátor odhaduje první parciální derivaci pomocí osmi konvolučních masek:

$$\begin{aligned} h_1 &= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} & h_2 &= \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} & h_3 &= \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \\ h_4 &= \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} & h_5 &= \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} & h_6 &= \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \\ h_7 &= \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} & h_8 &= \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \end{aligned}$$

Aplikujeme všechny masky na obrazovou funkci $g(i,j)$ a vybereme tu s největším gradientem. Chceme-li najít jen hrany jednoho směru, pak použijeme příslušnou konvoluční masku daného směru.

3. Sobelův operátor

Tento operátor se používá stejným způsobem jako předchozí operátor pro aproximaci první parciální derivace. Může však mít různé rozměry masek. Pro rozměr masky 3x3 vznikne opět osm masek pootočením masky 3.13 do všech osmi směrů. [15], [16]

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.13)$$

4. Laplaceův operátor

Laplaceův operátor na rozdíl od předchozích aproximuje druhou derivaci obrazové funkce. Dále se liší v tom, že neudává směr hrany, ale její velikost. Konvoluční masky se určí z jeho definice:

$$\nabla^2(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2}, \quad (3.14)$$

kde $g(x, y)$ je obrazová funkce. Dvě používané konvoluční masky pro okolí o velikosti 3x3 jsou:

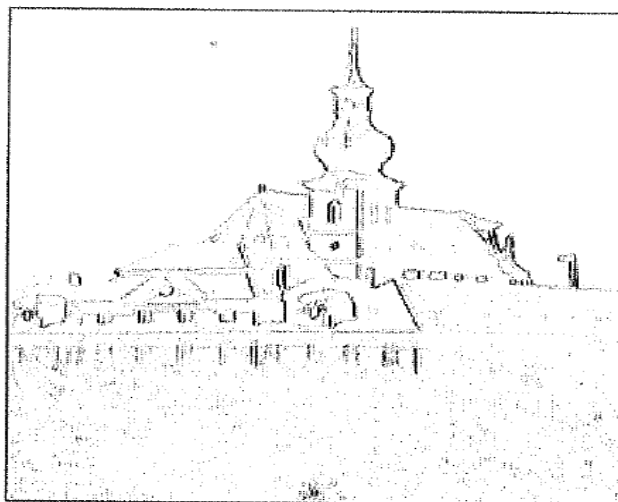
$$h_{4-soused} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad h_{8-soused} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Výhodou tohoto operátoru je necitlivost vůči otočení. Nevýhodou je velmi vysoká citlivost na šum. [15], [16]

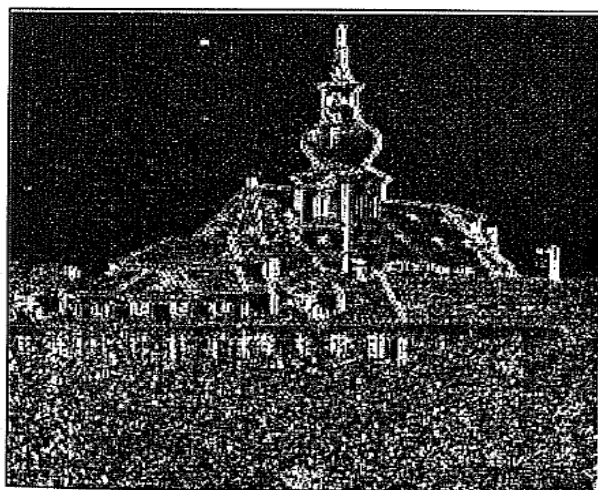
V praxi se používá mnoho dalších gradientních operátorů. Každý z nich se hodí pro jiný vstupní obraz a výběr se často určuje experimentálně. Pro srovnání jsou na obrázcích 3.9 zobrazeny postupně výchozí obraz ve 256 odstínech šedi, obraz zpracovaný Sobelovým operátorem a obraz zpracovaný Laplaceovým operátorem. Hlavní nevýhodou těchto operátorů je vysoká citlivost na šum a závislost na velikosti objektů v obraze (velikost konvoluční masky musí odpovídat velikosti významných detailů v obraze).



a) původní obraz



b) Obraz zpracovaný Sobelovým operátorem 3x3



c) Obraz zpracovaný Laplaceovým operátorem

Obr.3.9. Příklad užití konvolučních masek pro hledání hran [16]

Operátory založené na aproximaci druhé derivace

Pomocí těchto operátorů vyhledáváme hrany v místech nulové 2. derivace obrazové funkce $g(i,j)$. Využívají tedy poznatku, že druhá derivace funkce je v místě lokálního extrému rovná nule. V praxi je tento fakt velice užitečný, protože hledat průchody nulou je jednodušší než hledat maxima prvních derivací.

Druhou derivaci obrazové funkce můžeme aproximovat Laplaceovým operátorem. Problém je však v tom, že tato aproximace je velice citlivá na šum. Proto před její aplikací se musí šum vyfiltrovat. K tomu se velmi dobře hodí Gaussův filtr, který byl popsán v podkapitole 3.2.4. Díky lineárnosti operace konvoluce (značena $*$) platí:

$$\nabla^2(G * g) = (\nabla^2 G) * g = 0 \quad , \quad (3.15)$$

kde G představuje filtraci Gaussovým filtrem a g je obrazová funkce v níž vyhledáváme hrany. Z rovnice (3.15) je vidět, že můžeme nejdříve pomocí Laplaceova operátoru a rovnice 3.5 získat aproximaci druhé derivace funkce G a z ní získat konvoluční masku pro zvolenou velikost okolí σ , kterou aplikujeme na funkci g . Po-té můžeme již snadno najít hrany, jako pixely s nulovou druhou derivací. [15], [16]

Cannyho hranový detektor

Princip jeho činnosti je ve výpočtu 1. a 2. derivace. Používá se ve spolupráci s Gaussovým filtrem. Detekuje hrany metodou průchodu nulou ve směru lokálního maxima gradientu.

Je sestaven tak, aby vyhovoval třem kritériím [15], [16]:

1. Detekční kritérium – nesmí být opomenuty důležité hrany
2. Lokalizační kritérium – snaha o minimalizaci odchylky mezi nalezenou a skutečnou hranou
3. Násobnosti hran – nesmí reagovat na jednu hranu vícekrát

Pomocí Gaussova filtru vyhledáme první derivaci obrazové funkce. Ve směru maximálního spádu detekujeme hrany pomocí druhé derivace. Po té aplikujeme algoritmus pro prahování s hysterezí. [22], [23] To nám pomůže odstranit nepodstatné hrany a hysterezi.

Algoritmus pro prahování s hysterezí spočívá v určení intervalu $\langle T_0; T_1 \rangle$, kde T_0 a T_1 určují prahy pro hysterezi a nepodstatné hrany. Pak označíme všechny pixely, jejichž hodnota je větší než velikost většího prahu T_1 , a budeme je považovat za pixely budoucí hrany. Poté prohledáme obraz a pro všechny pixely v intervalu $\langle T_0; T_1 \rangle$ zjistíme, zda v jeho okolí existuje pixel budoucí hrany. Pokud ano, pak tento pixel se také označí a považuje se za součást budoucí hrany. Opakujeme prohledávání obrazu pro zjištění dalšího bodu patřícího do hrany, dokud se výstupní obraz mění.

Algoritmů pro detekci hran je celá řada. Ty uvedené v této kapitole patří k nejpoužívanějším. Volba správného algoritmu záleží na charakteru obrazu a velmi často se volí experimentálně.

3.3.4 Sledování hranice a propojování hran

Máme-li detekovány hrany, pak se na obrazu objevují jen dvě barvy a to černá a bílá, která představuje nalezené hrany. Obraz má však stále vady a to v podobě falešných hran, zbytkového šumu a chybějících částí některých hran. Proto se tyto obrazy dále zpracovávají.

Častým požadavkem je také, aby hrany měli tloušťku pouze jeden pixel (jednopixelové hrany) a byly spojitě. K tomuto účelu se obraz dále upravuje tzv. metodou sledování hranice a spojovacím algoritmem.

Metoda sledování hranice

Algoritmus se používá ke snížení tloušťky hranice na jeden pixel. Otázkou je, který z pixelů hranice se má odstranit a který zachovat. Tento algoritmus zachovává vnitřní hranici objektu. Ostatní body hranice budou odstraněny. Myšlenka je taková, že se najde vnitřní bod objektu, který leží u hraniční křivky. Tento bod se bude považovat za počáteční bod nové hranice. Poté se najde další bod nové hranice, který sousedí s počátečním bodem hranice a současně se nalézá v bezprostřední blízkosti původní hraniční křivky. Po aplikaci tohoto algoritmu vznikne nová jednopixelová hranice. Ta však může být nespojitá. Tato vada se pak odstraňuje pomocí spojovacího algoritmu.

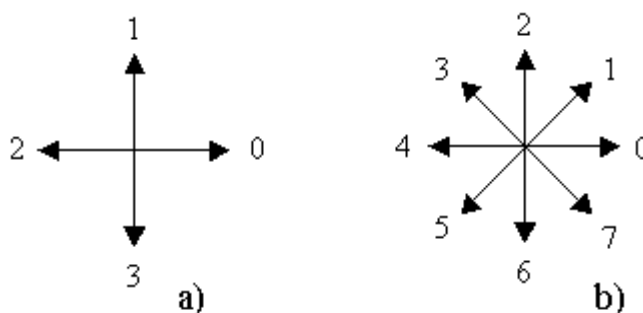
Postup pro metodu sledování hranice [15]:

1. Postupně procházíme jednotlivé pixely zleva doprava z nejspodnějšího řádku nahoru, dokud nenarazíme na první bod patřící do objektu. Tento bod označíme jako počáteční bod nové hranice a vložíme ho do proměnné `aktuální_bod`. Má tu vlastnost, že leží v nejspodnějším řádku, který obsahuje alespoň jeden pixel zkoumaného objektu, a že v tomto řádku leží nejvíce vlevo ze všech pixelů patřících do objektu.
2. Do proměnné `smer_postupu` se vloží pro 4-okolí hodnota 3 a pro 8-okolí hodnota 7. V této proměnné se uchovává směr od posledního přidaného bodu do hranice k aktuálnímu zkoumanému bodu.
3. Vyhledáme následující hraniční bod objektu. Je to ten bod, který nalezneme při prohledávání okolí 3x3 aktuálního hraničního bodu proti směru hodinových ručiček. Počáteční směr je dán rovnicí:

- $(smer_postupu + 3) \bmod 4$ pro 4-okolí
- $(smer_postupu + 6) \bmod 8$ pro 8-okolí a je-li `smer_postupu` liché číslo
- $(smer_postupu + 7) \bmod 8$ pro 8-okolí a je-li `smer_postupu` sudé číslo

Směry jsou vyznačeny na obrázku 3.10. Proměnná `smer_postupu` a `aktualni_bod` se aktualizuje.

4. Bod 3 se opakuje, dokud se v proměnné `aktulni_bod` neobjeví bod nově vznikající hranice.
5. Body 1 až 4 se opakují pro všechny objekty v obrazu.



Obr. 3.10 Směry pro metodu sledování hranice -a) 4-okolí, b)8-okolí [15]

Spojovací algoritmus

Slouží k propojení hran, které ohraničují jeden objekt. Výsledkem by měla být uzavřená křivka. Algoritmus je založen na následujících třech bodech [20]:

1. Zvolíme hodnotu δ , jenž udává maximální vzdálenost mezi hranami, které mohou být propojeny.

2. Pro všechny koncové body hran testujeme jejich vzájemnou polohu. Jestliže dva koncové body A a B mají vzdálenost menší než je hodnota δ , tak je propojíme. Vznikne tím nová hrana skládající se z hran určených koncovými body A a B a nově vloženou hranou spojující body A a B. Vznikne-li uzavřená hrana, tak se již o této hraně nebude do dalšího spojování uvažovat. Tento bod se bude opakovat tak dlouho, dokud existují koncové body hran, jejichž vzdálenost je menší než δ .
3. Dokud nebudou všechny hrany v obraze uzavřené, tak se zvýší hodnota proměnné δ a bod 2 se opakuje.

Pro propojování koncových bodů hran existují různé algoritmy, mezi nejpoužívanější patří algoritmus pomocí interpolátoru DDA a Bresenhamův algoritmus.

3.3.5 Vyplňování objektů

Další možností segmentace je pomocí vyplňování objektů. Tato metoda se snaží od sebe odlišit objekty podle určené vlastnosti hranice. Tyto metody se dají také využít na vyplňování objektů, u kterých známe jejich hraniční body. Nevýhodou metod vyplňování objektů je jejich časová náročnost, proto pokud je to možné se nahrazují metodami barvení objektů z kapitoly 3.3.2.

Algoritmy pro vyplňování objektů se dají rozdělit do dvou hlavních skupin. Jsou to algoritmy řádkové konverze a metoda semínka. V prvním případě jsou definovány hranice objektu explicitně (např.: seznamem hran). Pak se poloha vyplňovaného pixelu vůči poloze objektu určuje geometricky. Druhá skupina metod je založená na znalosti alespoň jednoho vnitřního bodu objektu a na vlastnosti charakterizující hranici objektu. V této kapitole se budeme zabývat pouze druhou metodou a to jejími dvěma modifikacemi. [18]

Semínkové vyplňování

Tento algoritmus vyžaduje, abychom znali alespoň jeden bod (semínko), který patří do ohraničeného objektu. Začneme vyplňovat objekt od tohoto bodu. Zvolíme si barvu, kterou chceme mít objekt obarven a semínko touto barvou obarvíme. Od tohoto bodu se bude semínko šířit. Algoritmus bude hledat všechny vnitřní body objektu, které lze dosáhnout ze semínkového bodu pohybem v rastrové mřížce, aniž by byla překročena hranice objektu. Tyto body obarví zvolenou barvou objektu. V mřížce se můžeme pohybovat buď jen ve 4 hlavních směrech (nahoru, dolů, vpravo, vlevo) nebo ve všech 8 směrech (4 hlavní směry + 4 směry svírající s hlavními směry 45°). První způsob obarví objekt jako souvislou oblast 4-okolí a druhý jako souvislou oblast 8-okolí, kde souvislou oblastí rozumíme všechny body mezi nimiž existuje cesta složená z horizontálních nebo vertikálních, popřípadě i v diagonálních směrů, aniž bychom překročili hraniční body objektu.

Při segmentaci je přirozenější použití 4-okolí, jelikož 8-okolí považuje dvě plochy stýkající se v rohovém bodu za jeden objekt.

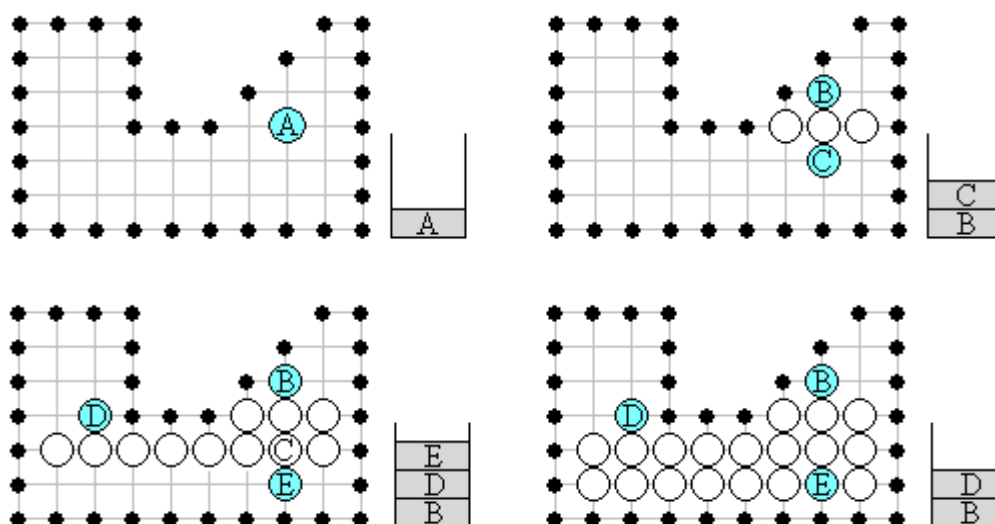
Dále by jsme měli definovat, kdy je testovaný bod vnitřní. Tři nejznámější varianty jsou:

- Hraniční vyplňování – Testovaný bod je vnitřní, má-li jinou barvu, než je námi zvolená barva hranice.
- Záplavové (lavinové) vyplňování – Testovaný bod je vnitřní, má-li námi zvolenou barvu semínka.
- Měkké vyplňování – Testovaný bod je vnitřní, má-li výrazně odlišnou barvu od námi určené barvy hranice. Tento způsob se používá například, byla-li hranice vyhlazena a nemá jednotnou barvu.

Pro šíření semínka je možno použít dva algoritmy. První se nazývá naivní algoritmus. Ten pro každý bod, který byl obarven, testuje všechny 4 respektive 8 sousedů, zda jsou vnitřními body a podle situace je obarví nebo ne. Tento postup lze realizovat pomocí rekurzivní procedury. To je však nevhodné z hlediska paměťových nároků. Lepší je použít postup pomocí prohledávání stavového prostoru (např.: prohledávání do hloubky, do šířky...), který je o něco šetrnější na paměť. Obě tyto metody jsou však dosti neefektivní, protože se každý bod testuje několikrát, zda se má obarvit.

Řádkové semínkové vyplňování.

Druhý algoritmus je daleko efektivnější. Na obrázku 3.11 je naznačen postup. Nejdříve se zvolí semínko A, to se vloží do zásobníku. Poté se prochází postupně body napravo od bodu A a testují se, zda jsou vnitřními body objektu. Pokud ano, tak se obarví. Narazíme-li na 1.bod, který není vnitřní, tak se prochází postupně body vlevo od bodu A a obarvují se, pokud jsou vnitřními body. Narazíme-li na první bod, který není vnitřní, tak obarvování tohoto řádku skončí. Vznikne tak obarvená úsečka u_1 . Bod A vyjmeeme ze zásobníku. Po té najdeme všechny souvislé oblasti vnitřních bodů objektu nad a pod touto úsečkou, které ještě nebyly obarveny. Z každé takto nalezené souvislé oblasti vybereme libovolný bod a vložíme ho do zásobníku. Na obrázku 3.11 vpravo nahoře to jsou body B a C. Nyní se vezme bod z vrcholu zásobníku a provedeme s ním stejný postup jako s bodem A. Tento postup opakujeme dokud není zásobník prázdný. [17], [18]



Obr. 3.11 Algoritmus semínkového řádkového vyplňování [25]

3.4 Popis objektů

Po segmentaci obrazu podle vybraných algoritmů z kapitoly 3.3 víme o každém pixelu, kterému objektu patří, proto můžeme každý objekt popsat a zpracovat zvlášť. Popis jednotlivých objektů je různorodý a záleží na tom, co s ním chceme dělat dál. Můžeme ho popsat pomocí číselných charakteristik, nebo kvalitativně pomocí relací mezi objekty.

4 ALGORITMY PRO VYTVÁŘENÍ NÁHODNĚ DRSNÉHO POVRCH A SIMULACI INTERFERENCEČNÍHO MIKROSKOPU

Algoritmy uvedené v předchozí kapitole, lze použít ke zpracování obrazu interferogramu. Po tomto zpracování je možné určit některé statistické charakteristiky povrchu. Problém je však v ověřování správnosti výsledků stříhové interferometrie. Jednou z možností je srovnat výsledky dosažené stříhovou interferometrií s výsledky dotykového profilometru. Takové srovnání bylo provedeno ve článku [2]. V této práci byli srovnávány hodnoty vzorku skleněných matric připravených jednostranným leptáním skleněných destiček ve směsi HF a H₂O₂ v poměru 2:3 a hodnoty vzorku parabolického světelného reflektoru.

V prvním případě se musel použít postup pomocí metody nejmenších čtverců. Po zpracování byla výsledná hodnota dosažená stříhovou interferometrií střední kvadratické odchylky výšek nerovností σ a autokorelační vzdálenost T rovna:

$$\sigma = (1,84 \pm 0,06) \mu\text{m} \quad \text{a} \quad T = (61 \pm 6) \mu\text{m}.$$

Hodnoty zjištěné pomocí profilometru typu Talystep 4 byly:

$$\sigma = (1,90 \pm 0,08) \mu\text{m} \quad \text{a} \quad T = (60 \pm 5) \mu\text{m}.$$

Shoda obou výsledků je relativně dobrá.

Pro druhý případ se použil postup se znalostí oblasti saturace. Výsledek po použití stříhové interferometrie je:

$$\sigma = (12,5 \pm 0,4) \mu\text{m} \quad \text{a} \quad T = (45, \pm 3) \mu\text{m}.$$

Profilometrem typu Talystep 4 byly zjištěny hodnoty:

$$\sigma = (13,2 \pm 0,6) \mu\text{m} \quad \text{a} \quad T = (42,4 \pm 4) \mu\text{m}.$$

Také zde nejsou vidět velké rozdíly mezi výsledky.

Jiným způsobem, jak lze ověřit správnost výsledků stříhové interferometrie, je pomocí simulací. Nejprve vygenerujeme model povrchu splňující podmínky z kapitoly 2.4.1, u něhož budeme znát přesné hodnoty výšek povrchu v každém jeho bodě. Poté provedeme simulaci interferenčního mikroskopu a dostaneme interferogram s interferenčními proužky. Tento obraz pak zpracujeme programem vytvořeným podle algoritmů z kapitoly 5. Po té můžeme srovnat statistické charakteristiky získané pomocí středů interferenčních proužků a ze znalosti hodnot výšek vygenerovaného povrchu.

4.1 Vytvoření náhodně drsného povrchu pomocí fraktálů

Vytvořený povrch pomocí simulace musí splňovat podmínky z kapitoly 2.4.1., jinak by se na něj nedala aplikovat metoda stříhové interferometrie. Takový povrch lze vygenerovat tzv. metodou plátů. Ta je založena na použití fraktálů. Její bližší popis lze najít v [26], [27]. Povrch budeme reprezentovat dvourozměrným polem, kde hodnota pole bude udávat výšku povrchu v daném bodě. Metoda je založena na postupném dělení povrchu na stále menší pláty. Začneme tím, že vyplníme rohy povrchu jeho výškou pomocí čtyř hodnot získaných Gaussovým generátorem náhodných čísel $N(\mu, \sigma)$, kde μ udává střední hodnotu a σ rozptyl tohoto rozdělení. Poté vyplníme středy jeho stran hodnotou podle vzorce:

$$v_{nove-k} = \frac{v_i + v_j}{2} + N\left(0, k \frac{\sigma}{2}\right), \quad (4.1)$$

kde v_{nove-k} je hodnota výšky povrchu ve střední vzdálenosti mezi rohy povrchu i a j , v_i a v_j jsou výšky povrchu v příslušných rozích, $N(0, k\sigma/2)$ je hodnota získaná Gaussovým generátorem a k je konstanta zvaná koeficient deformace.

Dále vyplníme střed povrchu podle vzorce:

$$v_{nove} = \frac{v_i + v_j + v_k + v_l}{4} + N\left(0, k \frac{\sigma}{2}\right), \quad (4.2)$$

kde v_{nove} je hodnota výšky povrchu v jeho středu, v_i , v_j , v_k a v_l jsou výšky povrchu v jeho rozích a $N(0, k\sigma/2)$ je hodnota získaná Gaussovým generátorem. Nyní můžeme rozdělit povrch na 4 pláty tak, že každý z nich bude obsahovat 4 vyplněné rohy. Pro každý plát vyplníme stejným způsobem body ve středu jeho stran a střed nového plátu jako jsme to udělali pro celý povrch. Aby však vznikl hladký povrch musíme s každým rozdělením plátu na další 4 podpláty snížit velikost rozptylu σ podle vzorce:

$$\sigma_{i+1}^2 = k \frac{\sigma_i^2}{2}, \quad (4.3)$$

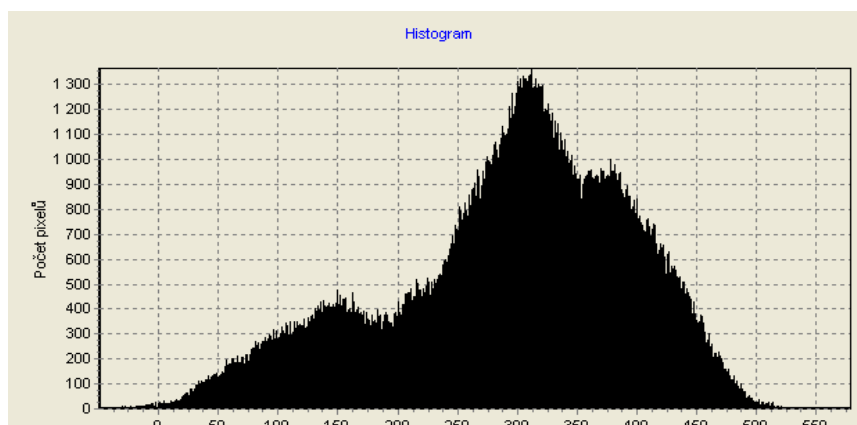
kde σ_{i+1} je nově získaný rozptyl pro Gaussův generátor, k je koeficient deformace a σ_i je rozptyl použitý pro Gaussův generátor z původního plátu. Popsaným způsobem vyplníme všechny body povrchu.

Výše popsáním způsobem dostaneme matici obsahující informaci o výškách povrchu ve všech jeho diskretních bodech. Vygenerovaný povrch tímto způsobem je znázorněn na obrázku 4.1 v odstínech šedi. Kde nejnižšímu bodu povrchu je přidělena nejtmavší a nejvyššímu bodu zase nejsvětlejší odstín šedi. Dále na obrázku 4.2 je zobrazen histogram povrchu z obrázku 4.1.



Obr. 4.1 Vygenerovaný povrch pomocí plátů v odstínech šedi s parametry: $\sigma=110 \mu\text{m}$, $k=1$

Při generování povrchu touto metodou se může použít rekurzivní volání funkce, kterou naleznete v programu na přiloženém CD k této práci. Funkce má název Vyplň a je umístěna v unitě uSimul.



Obr. 4.2 Histogram obrázku povrchu z obr. 4.1.

4.2 Simulace interferenčního mikroskopu

Pokud máme vygenerovaný povrch pomocí metody popsané výše, je třeba nějakým způsobem z něho dostat interferogram. Jelikož naše vygenerované povrchy nejsou reálné, musíme tento proces simulovat. Při tom se vychází z interference dvou vlnoploch ve vzájemném střihu, pro něž platí rovnice:

$$U(x, t) = U_1(x, t) + U_2(x, t) \quad , \quad (4.4)$$

kde U je výsledná vlna interferujících vln U_1 a U_2 v místě x a v čase t .

Z Maxwellových rovnic vyplývá:

$$J(x, t) = U(x, t) \cdot U^*(x, t) \quad , \quad (4.5)$$

kde J je intenzita v bodě x a čase t . U je rovnice vlnoplochy a U^* je rovnice k ní komplexně sdružená. Z těchto dvou rovnic se dá odvodit rovnice (4.6) pro výpočet výsledné intenzity interferujících vln v bodě x a čase t . Odvození naleznete v knize [24].

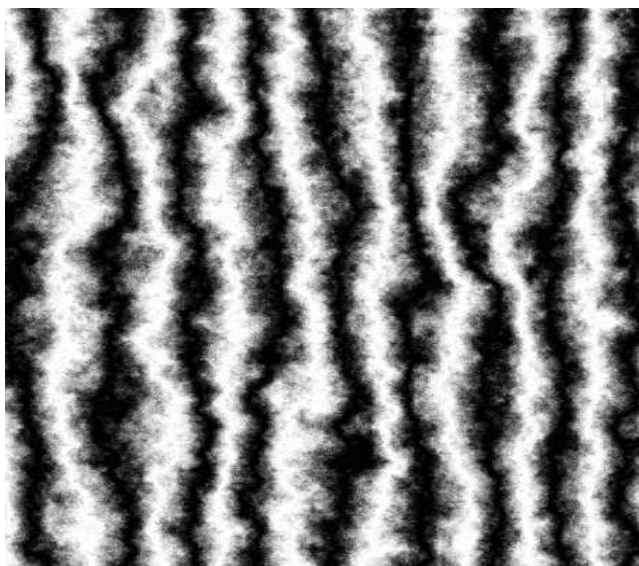
$$J(x, t) = 4J_0(x, t) \cos^2(\Delta(x, t)/2) \quad , \quad (4.6)$$

kde J_0 je intenzita vstupní vlny a Δ je dráhový rozdíl mezi interferujícími vlnoplochami. V našem případě jde o interferenci dvou stejných vlnoploch, které jsou otiskem povrchu (vlnoplocha odpovídá vygenerovanému povrchu). Aplikací vzorce (4.7) na rovnici (4.6) a volbou hodnoty $J_0 = 1/2$ dostaneme rovnici (4.8), kterou jsem použil pro simulaci interferenčních proužků v přiloženém programu.

$$\cos \frac{\alpha}{2} = \sqrt{\frac{1 + \cos \alpha}{2}} \quad (4.7)$$

$$J(x, t) = (1 + \cos \Delta(x, t)) \quad (4.8)$$

Vygenerovaný interferogram z povrchu na obrázku 4.1. je vidět na obrázku 4.3.



Obr. 4.3. Interferogram vytvořený simulací činnosti interferenčního mikroskopu z povrchu na obr. 4.1.

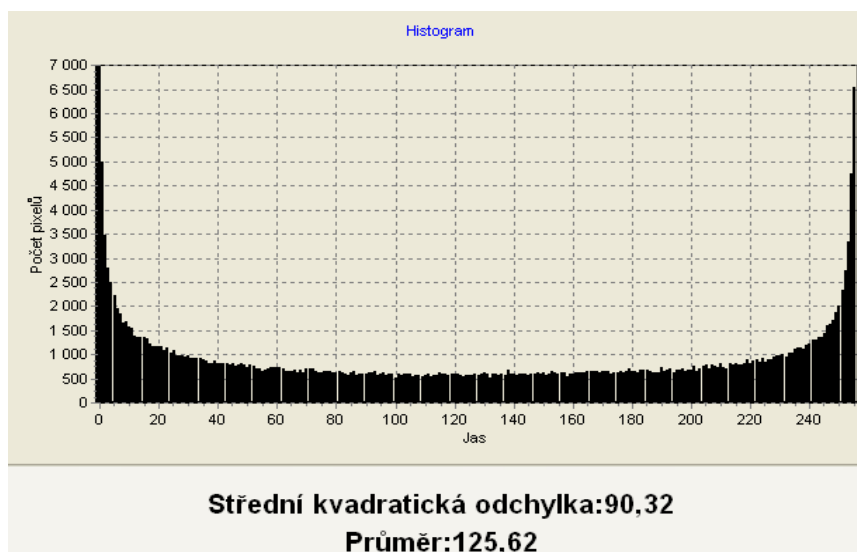
5 KONKRÉTNÍ NÁVRH ALGORITMŮ ZPRACOVÁNÍ OBRAZU PŘI URČOVÁNÍ PARAMETRŮ POVRCHU POMOCÍ METODY STŘIHOVÉ INTERFEROMETRIE

V této kapitole provedeme konkrétní návrh algoritmů použitých pro určování topografických parametrů obrazu. Vyjdeme z metody stříhové interferometrie. Jak bylo řečeno výše, výstupem interferenčního mikroskopu je interferogram s tmavými a světlými proužky podobný tomu z obrázku 2.5. Ten je zaznamenaný na CCD kameru. Odtud se přeneso do PC a uloží se jako černobílý bitmapový obrázek ve 256 odstínech šedi. Na VUT se při testování této metody použilo čtvercové vzorkování, kde 13 pixelům odpovídalo 10 μm . Získaný obrázek se upravuje v počítači pomocí vybraných algoritmu z kapitoly 3, aby se našly středy tmavých a světlých proužků. Z nich se vytvořily jednopixelové linie. Ty se pak statisticky zpracují podle podkapitoly 2.4.2. Algoritmům vedoucí k získání jednopixelových linií z digitalizovaného interferogramu a následné jejich statistické zpracování je věnována tato kapitola. Všechny algoritmy jsou popsány v objektově orientovaném programovacím jazyce Delphi 7. [2]

Nejdříve musíme najít vhodný matematický model, kterým bude interferogram v počítači reprezentován. Jelikož budeme zpracovávat černobílé bitmapové obrázky ve 256 odstínech šedi, použijeme obrazovou funkci $f(i,j)$, která bude nabývat hodnot odstínu šedi (jasu) ve složkách RGB modelu. Jak bylo řečeno v podkapitole 3.1, barva je reprezentována trojicí hodnot (r,g,b) , které v počítači nabývají celočíselných hodnot od 0 do 255. Hodnota každé složky odpovídá intenzitě jednotlivých barev. V podkapitole 3.1 bylo také řečeno, že hodnoty všech tří složek RGB model pro odstíny šedi mají stejnou velikost. Z toho vyplývá, že obrazová funkce $f(i,j)$ černobílého obrazu může být reprezentována hodnotou libovolné složky RGB modelu příslušného pixelu o souřadnicích $[i,j]$. Tímto způsobem je zvolen matematický model obrazové funkce $f(i,j)$ v přiloženém programu. Obrazová funkce tedy bude reprezentována maticí. Jednotlivé prvky matice budou odpovídat libovolné složce RGB modelu příslušného pixelu bmp obrázku. K získání složek RGB modelu pro jednotlivé pixely se dá použít funkce `ColorToRGB`, která převede barvu příslušného pixelu do trojice hodnot RGB modelu. Dále se použije funkce `GetXValue`, která vrátí hodnotu složky X RGB modelu. (X zastupuje jeden ze znaků R, G nebo B.) Hlavní část programu pro převod barevných pixelů bmp obrázku do naší požadované matice `FIData` je:

```
for I:=0 to VyskaI-1 do           //VyskaI,SirkaI....rozmary matice FIData
  for J:=0 to SirkaI-1 do
    begin
      Color:=ColorToRGB(Bmp.Canvas.Pixels[J,I]);           //Bmp ... digitalizovaný
      R:=GetRValue(Color);                                   //interferogram
      FIData[I,J]:=R;
    end;
```

Dále již nebudeme pracovat s jednotlivými pixely, ale s maticí reprezentující obrazovou funkci $f(i,j)$. Před samotným zpracováním digitalizovaného obrazu provedeme jeho předzpracování podle kapitoly 3.2. Cílem je upravit tuto matici pro potřeby dalšího zpracování. Jde především o potlačení šum a získání obrazu s proužky, jejichž hrany reprezentují středy interferenčních proužků. K potlačení šumu se použije filtrování. Pro získání požadovaných proužků použijeme funkci `arg grad`, která díky svým matematickým vlastnostem odhalí středy interferenčních proužků. Budeme-li mít obraz, v němž hrany reprezentují námi hledané objekty (středy interferenčních proužků), pak již stačí použít některý algoritmus z kapitoly 3.3, který hrany objeví. Tyto hrany pak použijeme k výpočtu veličiny σ_h podle vzorce (2.14). Každý krok prováděný s interferogramem z obrázku 4.3 bude zobrazen v příslušných podkapitolách v odstínech šedi společně s jeho histogramem a statistickými údaji.



Obr. 5.1 Histogram interferogramu z obr. 4.3.

5.1 Předzpracování interferogramu

Ještě než se pustíme do samotného zpracování digitalizovaného obrazu, provádí se tzv. předzpracování obrazu. Po předchozím kroku máme k dispozici matici s prvky reprezentující jas jednotlivých pixelů interferogramu. Cílem je odstranit šum filtrací a nalézt středy interferenčních proužků v podobě hran funkcí $\arg \text{grad}$. Šum mohl vzniknout například různou citlivostí snímacích bodů CCD kamery, nečistotami na povrchu vzorku nebo digitalizací obrazu z CCD kamery.

5.1.1 Filtrace interferogramu

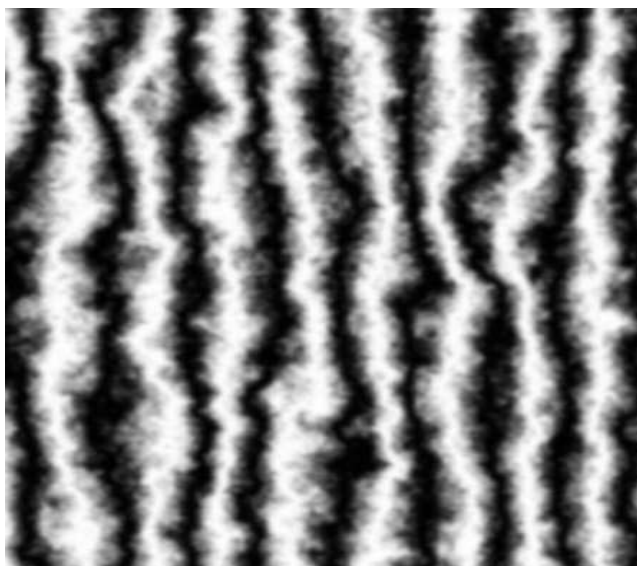
Vzhledem k tomu, že metoda vyšetřování interferogramu pomocí funkce $\arg \text{grad}$ je velmi citlivá na šum, je důležité použít filtraci. Během filtrace však dochází k potlačování hran, to však v našem případě není tak důležité, jelikož interferogramy neobsahují hrany. Důležitější věc je, že filtrace ovlivňuje následné statistické zpracování obrazu. Použitá filtrace proto musí být šetrná, aby nedošlo ke zkreslení statistických údajů. V programu, který je součástí této práce, je možnost použít jeden ze tří filtrů: lineární, medián se čtvercovým okolím 3×3 a medián 5 (Obr. 3.5).

Lineární filtr

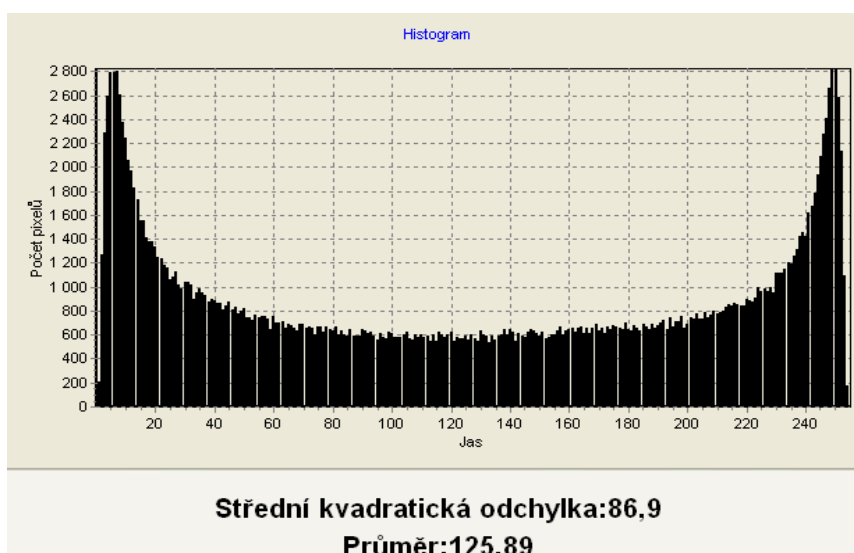
V programu byla použita konvoluční maska o rozměrech 9×9 , která aproximuje dvojrozměrné Gaussovo rozdělení:

$$h = \frac{1}{1617} \begin{bmatrix} 0 & 0 & 0 & 7 & 7 & 7 & 0 & 0 & 0 \\ 0 & 0 & 7 & 14 & 14 & 14 & 7 & 0 & 0 \\ 0 & 7 & 14 & 41 & 41 & 41 & 14 & 7 & 0 \\ 7 & 14 & 41 & 81 & 81 & 81 & 41 & 14 & 7 \\ 7 & 14 & 41 & 81 & 113 & 81 & 41 & 14 & 7 \\ 7 & 14 & 41 & 81 & 81 & 81 & 41 & 14 & 7 \\ 0 & 7 & 14 & 41 & 41 & 41 & 14 & 7 & 0 \\ 0 & 0 & 7 & 14 & 14 & 14 & 7 & 0 & 0 \\ 0 & 0 & 0 & 7 & 7 & 7 & 0 & 0 & 0 \end{bmatrix}$$

Postup použití lineárního filtru najdete v podkapitole 3.2.4. V přiloženém programu je lineární filtrace řešena funkcí s názvem *TFilterInterfer.LinearniFiltrace* v unitě *uAnalyz*.



Obr. 5.2 Interferogram z obr. 4.3 po lineární filtraci



Obr. 5.3 Histogram interferogramu z obr 5.2

Medián filtr

V programu byly použity okolí vyznačené na obrázku 3.5. Postup použití tohoto filtru je popsán v podkapitole 3.2.4. V příloženém programu jsou tyto filtrace řešeny funkcemi s názvy *TfilterInterfer.Median5* a *TfilterInterfer.Median3x3* v unitě *uAnalyz*.

5.1.2 Hledání středů temných a světlých proužků interferogramu

Po vyfiltrování interferogramu, jsme se zbavili lokálních poruch a nic nám nebrání, abychom se pustili do hledání středů interferenčních proužků. Pro tuto operaci jsem zvolil zpracování interferogramu pomocí funkce argument gradientu, která byla použita v práci [2]. Ta je založená na následující ideji. Symbol $b(x,y)$ definujeme jako argument gradientu funkce $f(x,y)$, tedy:

$$b(x,y) = \arg \text{grad } f(x,y) \quad , \quad (5.1)$$

kde symbol $f(x,y)$ značí obrazovou funkci digitalizovaného filtrovaného interferogramu.

Pro jednoduchost předpokládejme, že funkce f je spojitá, x značí pravoúhlé souřadnice kolmé na interferenční proužky a y značí pravoúhlé souřadnice rovnoběžné s interferenčními proužky.

Rovnici (5.1) můžeme podrobněji popsat těmito dvěma rovnicemi:

$$b(x, y) = \tan^{-1} \left[\left(\frac{\partial f(x, y)}{\partial y} \right) \left(\frac{\partial f(x, y)}{\partial x} \right)^{-1} \right] \quad \text{pro} \quad \frac{\partial f(x, y)}{\partial x} > 0 \quad (5.2)$$

$$b(x, y) = \pi + \tan^{-1} \left[\left(\frac{\partial f(x, y)}{\partial y} \right) \left(\frac{\partial f(x, y)}{\partial x} \right)^{-1} \right] \quad \text{pro} \quad \frac{\partial f(x, y)}{\partial x} < 0 \quad (5.3)$$

Je vidět, že funkce $b(x, y)$ není definována pro $\frac{\partial f(x, y)}{\partial x} = (0, 0)$.

Pokud se pozorně zamyslíme nad definicí funkce $b(x, y)$, tak dojdeme k závěru, že jejím výsledkem je úhel mezi vektorem gradientu funkce $f(x, y)$ a kladnou osou x . Tedy v místě, kde není funkce definována, je nespojitá a vytváří se zde hrany. Tyto místa jsou právě minima a maxima interferenčních proužků, které hledáme pro odhalení jejich středů.

Metoda hledání středů interferenčních proužků založena na použití funkce $\arg \text{grad}$ přináší však své potíže:

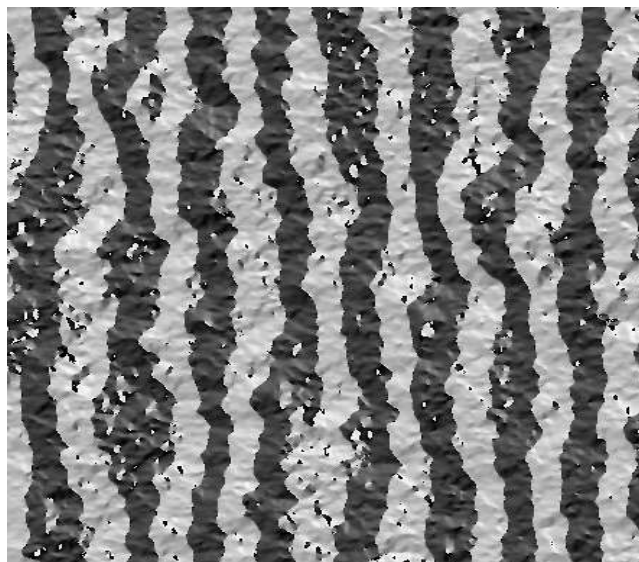
- (i) Metoda je velmi citlivá na obrazový šum, proto musí být před jejím použitím provedeno odfiltrování šumu.
- (ii) Funkce $b(x, y)$ je nespojitá také v bodech, kde přechází úhel svírající gradient funkce $f(x, y)$ s kladnou poloosou x z 2π na 0 . To by způsobovalo vznik falešných hran v těchto místech a značně by nám to znepříjemnilo hledání jednopixelových linií v interferogramu. Tento problém se dá často vyřešit změnou souřadnicového systému. Tato změna však musí být taková, aby nedošlo k odstranění nespojitosti v bodech značící středy interferenčních proužků. V našem případě postačí přičíst hodnotu $\pi/2$ k výsledné hodnotě funkce $b(x, y)$.
- (iii) Ve filtrovaném interferogramu se nachází oblasti s nulovým sklonem ($\frac{\partial f(x, y)}{\partial x} = 0$ a $\frac{\partial f(x, y)}{\partial y} = 0$). Zde není funkce $b(x, y)$ definovaná. Můžeme tedy tyto body z analýzy vynechat, nebo přistoupit k těmto bodům individuálně. Rozhodl jsem se pro druhou variantu, aby nedocházelo ke ztrátě informací z interferogramu a dopočítal jsem hodnoty v těchto bodech pomocí mediánového filtru z okolních hodnot.

Nyní již máme vyřešeny všechny detaily ohledně aplikování funkce $b(x, y)$ na filtrovaný interferogram. Postupně zleva doprava, řádek po řádku aplikujeme funkci $\arg \text{grad}$ na body matice obrazové funkce filtrovaného interferogramu. Každá nově vzniklá hodnota funkce $b(x, y)$ musí být přepočtena z rozsahu $\langle 0; 2\pi \rangle$ na interval $\langle 0; 255 \rangle$. Po provedení funkce $\arg \text{grad}$ se všemi modifikacemi vysvětlenými výše na filtrovaný interferogram na obr. 5.2 jsme získali nový interferogram zobrazený na obr. 5.4. Jeho histogram se základními statistickými údaji je na obr. 5.5.

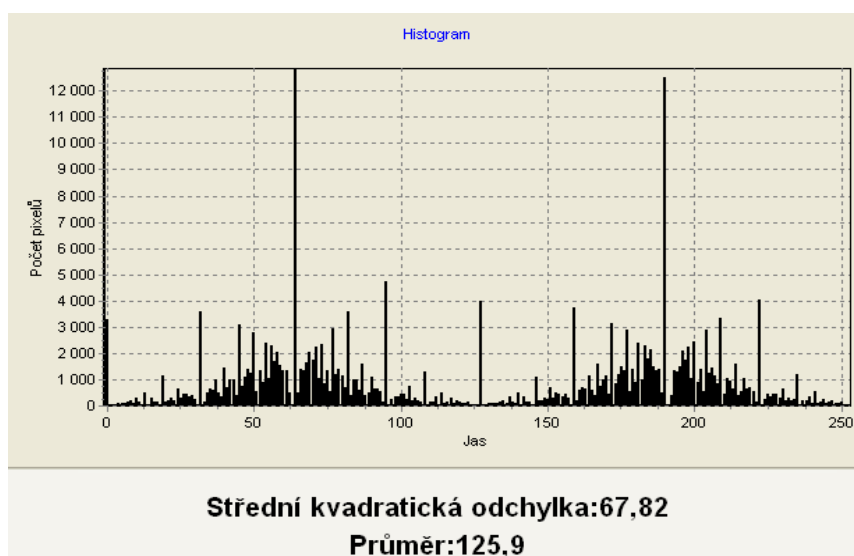
Obr. 5.4 je tvořen novými proužky, kde odstín šedi pixelů je úměrný velikosti úhlu mezi gradientem funkce $f(x, y)$ a kladnou poloosou x . Jak bylo vysvětleno výše, místa nespojitosti funkce

$b(x,y)$ tvoří středy interferenčních proužků. Ve skutečnosti funkce $b(x,y)$ a $f(x,y)$ nejsou spojité, jak jsme předpokládali, ale díky digitalizaci interferogramu zaznamenaného CCD kamerou jsou diskrétní. Proto nespojitost funkce $b(x,y)$ lze nalézt nějakým algoritmem řešící hranovou detekci z podkapitoly 3.3.

V příloženém programu je aplikace funkce `arg grad` na filtrovaný interferogram řešen funkcí s názvem `TArgGradInterfer.ArgGrad`, který najdete v unitě `uAnalyz`.



Obr. 5.4 Interferogram po aplikaci funkce `arg grad` na obr 5.2.



Obr. 5.5 Histogram interferogramu z obrázku 5.4.

5.2 Hledání jednopixelových linií

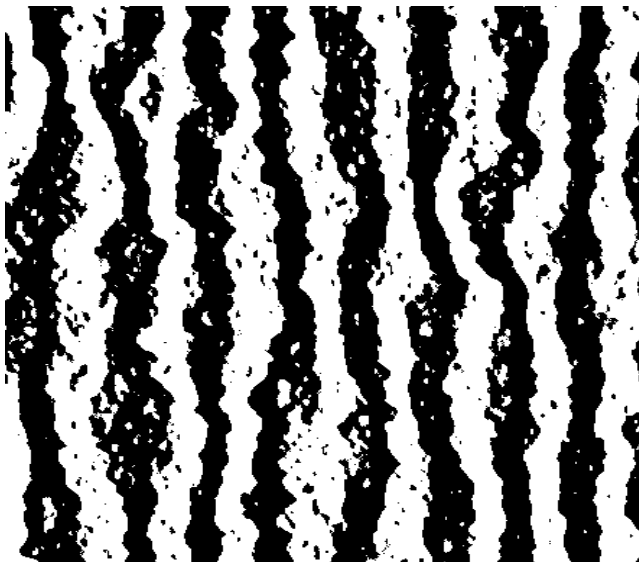
Jak již bylo řečeno v předchozí kapitole, máme nyní upravený interferogram s proužky, jejichž okraje určují pozici středů interferenčních proužků interferogramu získaného z interferenčního mikroskopu. Nyní je tedy třeba najít jednopixelové linie středů interferenčních proužků a z nich pak zjistit hodnotu σ_n . Abychom získali již zmiňované jednopixelové linie, budeme interferogram dále upravovat algoritmy uvedenými v této kapitole.

5.2.1 Bipolarizace interferogramu

Podíváme-li se na histogram interferogramu zpracovaného funkcí $\arg \text{grad}$ na obrázku 5.5, vidíme, že většina hodnot jasu je seskupena kolem dvou extrémů. Ty rozdělují hodnoty obrazové funkce na dvě skupiny. Z obrázku 5.4 je zřejmé, že jedna skupiny se ztotožňuje se světlými a druhá s tmavými proužky. Budeme tedy nyní považovat každý proužek za objekt a rozdělíme je do dvou skupin. Do první skupiny budou patřit tmavé proužky, kterým přiřadíme hodnotu obrazové funkce rovnou nule (černá barva). Druhá bude složena ze světlých proužků, těm přiřadíme hodnotu obrazové funkce 255 (bílá barva). Toto odlišení objektů provedeme pomocí algoritmu prahování, který byl popsán v podkapitole 3.3.1. Vzniklý interferogram budeme nazývat bipolarizovaným interferogramem.

Otázkou je jak stanovit práh mezi objekty. Jelikož jde jen o dva typy objektů s rozdílnými hodnotami obrazové funkce, může práh stanovit jako průměrnou hodnotu jasu všech pixelů v obrázku. Dále můžeme postupovat podle podkapitoly 3.3.1.

Bipolarizovaný interferogram, vzniklý z interferogramu na obrázku 5.4 je vidět na obr. 5.6 a jeho histogram je na obrázku 5.7.



Obr 5.6 Interferogram po bipolarizaci z obr. 5.4.

Bipolarizace je řešena funkcí *TBipoInterfer.Bipolar* v unitě uAnalyz.



Obr 5.7 Histogram bipolarizovaného interferogramu z obrázku 5.6

5.2.2 Separace interferogramu

Nyní máme interferogram skládající se z černých a bílých proužků. Hrany přechodů mezi nimi představují středy interferenčních proužků. Bipolarizovaný interferogram je téměř separovaný (jednotlivé objekty - proužky jsou od sebe odděleny obarvením). V některých bipolarizovaných interferogramech se však mohou nalézat poruchy, ty můžeme pozorovat i na obrázku 5.6. Projevují se jako malé oblasti bílých ostrůvků v černém proužku, nebo malé oblasti černých ostrůvků v bílém proužku. Aby byl interferogram zcela separovaný, musíme se těchto oblastí zbavit. Dále by bylo výhodné jednotlivé objekty patřící do jedné skupiny od sebe odlišit jasně, ale zároveň mít stále přehled o každém proužku do jaké skupiny patří (světlý nebo tmavý). V programu, který jsem sestavil, jsou od sebe barevně odlišeny jen světlé proužky, protože vyhledává jen střední linie tmavých interferenčních proužků. Světlé proužky by totiž mohli být přeexpandované a to by ovlivnilo statistické výsledky analýzy.

Odstranění poruch a odlišení bílých proužků se provede pomocí semínkového vyplňování, která byla popsána v kapitole 3.3.5. Aplikaci této metody budu nazývat separací, protože definitivně od sebe odliší jednotlivé objekty (proužky). Nově vzniklý interferogram budeme nazývat separovaným interferogramem.

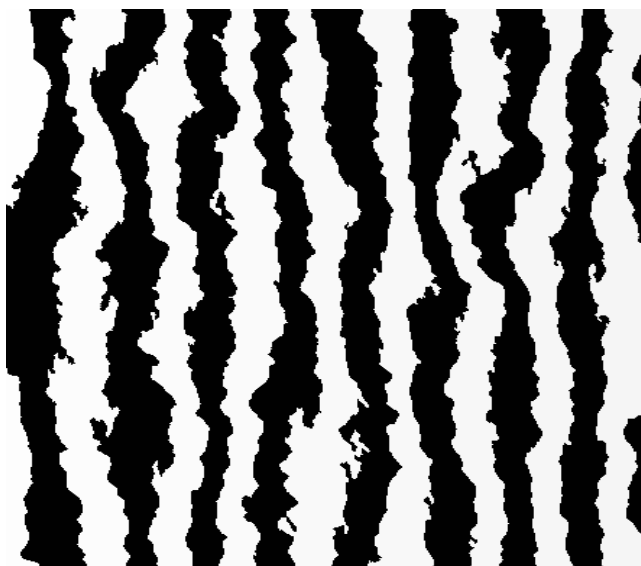
Pro účely separace byla použita záplavová modifikace semínkového vyplňování realizována naivním algoritmem pomocí prohledávání do hloubky (viz podkapitola 3.3.5).

K obarvení vnitřních oblastí všech objektů jsem použil funkci s názvem *ObarviOkoli*, která se nachází v unitě *uAnalyz*. Tato funkce obarví plochu barvy *aZBarva* do barvy *aDoBarva*. Jako semínko se použije bod o souřadnicích [*aVýška*, *aSirka*]. Je-li velikost obarvené plochy v pixelech menší než *aVelikost*, tak plocha je identifikována jako porucha a funkce přebarví plochu barvou *aJinakBarva*. Proměnné *aVyska*, *aSirka*, *aZBarva*, *aDoBarva*, *aVelikost* a *aJinakBarva* jsou funkci předány parametrem. Návrátová hodnota funkce je dána výsledkem barvení. Je-li 0, pak nevznikl nový proužek a obarvená plocha je porucha, která se musí odstranit změnou barvy na barvu jejího okolí. Je-li návratová hodnota 1, vznikl nový proužek. Je-li návratová hodnota rovna 2, pak nastala chyba a separace se musí ukončit.

Zmiňovaná funkce se použije pro odstranění černých i bílých ostrůvků. Nejdříve se prochází bod po bodu a hledají se černé pixely. Pokud se nějaký najde, aplikuje se na něj funkce *ObarviOkoli*,

která obarví souvislou černou barvou oblast odstínem šedi rovným 1 (téměř černá barva-je použita pro identifikaci již obarvených černých pixelů). Pokud je oblast menší než námi zvolená mez pak se jedná o poruchu a přebarví se na bílo, aby splynula s okolím. (Mez byla v programu stanovena na 1/256 velikosti obrazu.) Aplikací popsaného postupu na všechny černé pixely se zbavíme černých ostrůvků. Nyní musíme ještě odstranit bílé ostrůvky a odlišit světlé proužky od sebe barvou. Algoritmus prohledává postupně pixely, jestliže narazí na první bílý pixel, pak na něj aplikuje funkce *ObarviOkoli*, která obarví souvislou oblast bílých pixelů barvou $b = \text{White} - 1$. Je-li tato oblast větší než naše hranice určená pro rozpoznávání poruch, tak snížíme hodnotu b o jedničku, jinak oblast přebarvíme odstínem šedi rovným 1. Po té hledáme další bílý bod a aplikujeme na něj stejným způsobem funkci *ObarviOkoli*. Tak pokračujeme, dokud neprohledáme celý interferogram.

Nyní máme separovaný interferogram zbavený lokálních poruch v podobě ostrůvků. Máme také vyplněné jednotlivé objekty barvou. Hodnota obrazové funkce tmavých proužků nabývá hodnoty rovné 1 a světlé proužky jsou navzájem barevně odlišené. Interferogram vzniklý výše uvedeným postupem z interferogramu na obrázku 5.6 je zobrazený na obr. 5.8 a jeho histogram na obrázku 5.9.



Obr. 5.8 Interferogram 5.6 po separaci.

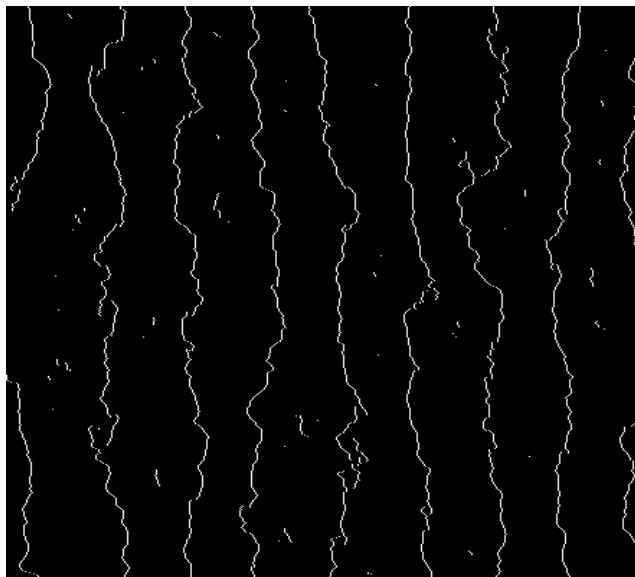


Obr. 5.9 Histogram interferogramu z obr. 5.8

Separace je řešena funkcí *TSeparInterfer.Separace* v unitě *uAnalyz*.

5.2.3 Nalezení jednopixelových linií

Nyní máme separovaný interferogram, kde hrany mezi jednotlivými proužky představují středy interferenčních proužků. Jelikož nás budou zajímat jen minima interferenčních proužků, budeme hledat jednopixelové linie, které kopírují přechod ze světlého do tmavého proužku. Můžeme použít některou metod pro detekci hran popsaných v kapitole 3.3.3 nebo použít metodu sledování hranice z kapitoly 3.3.4. Já jsem se rozhodl pro metodu detekce hran na základě změny jasu dvou sousedních pixelů ve směru kolmém k interferenčním proužkům a pro metodu sledování hranice.

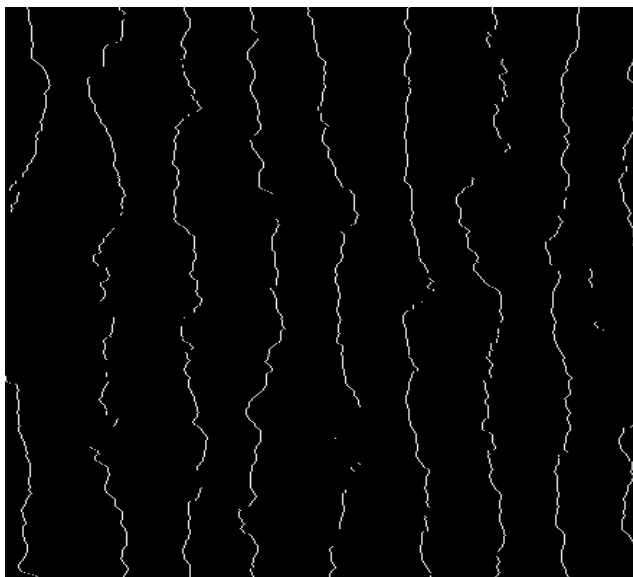


Obr 5.10 Interferogram s vyznačenými jednopixelovými liniemi získaný metodou detekce změny jasu ve směru osy *x* z interferogramu z obrázku 5.8.

První metoda pracuje na velice jednoduchém principu. Budeme postupně procházet pixely zleva doprava řádek po řádku a obarvovat je. Budeme vždy porovnávat barvu dvou sousedních pixelů ve směru osy *x*. Je-li pixel tmavší nebo stejně tmavý jako jeho pravý soused, tak tak ho obarvíme načerno (přiřadíme mu hodnotu odstínu šedi rovnou 0), v opačném případě se hodnota šedi pixelu nezmění. Mějme na paměti, že jednotlivé proužky mají jiný odstín šedi. Proto po aplikaci výše uvedeného postupu na celý interferogram v něm vzniknou hledané jednopixelové barevně odlišené linie (Obr. 5.10). Tyto linie představují minima interferenčních proužků. Obdobným způsobem by se daly najít i jejich maxima. Z výše uvedených důvodů se však tyto linie nehledají. Funkce vyhledávající minima tímto algoritmem se v příloženém programu nachází v unitě *uAnalyz* a jmenuje se *TMinInterfer.HledejMinima*.

Tento postup je velice jednoduchý a rychlý. Má však tu nevýhodu, že linie nejsou v některých místech spojitě. Dalším problémem je možná detekce krátkých čar, které nepředstavují středy temných proužků (mohou patřit k hranám charakterizující maxima interferenčních proužků nebo představují chyby vzniklé nečistotami na vzorku). I tyto vady můžeme pozorovat na obrázku 5.10. Proto je potřeba z tohoto interferogramu vypustit zdvojené části linií, ty jsou totiž potenciálním zdrojem těchto vad.

Odstranění zdvojených linií zajišťuje funkce, která se v příloženém programu jmenuje *TMinInterfer.OdstranZdvojLin*. Pro všechny barvy linií v interferogramu prochází každý řádek a pokud zjistí více než jedenásobný výskyt pixelu o stejné barvě v jednom řádku, tak všechny tyto pixely obarví načerno.

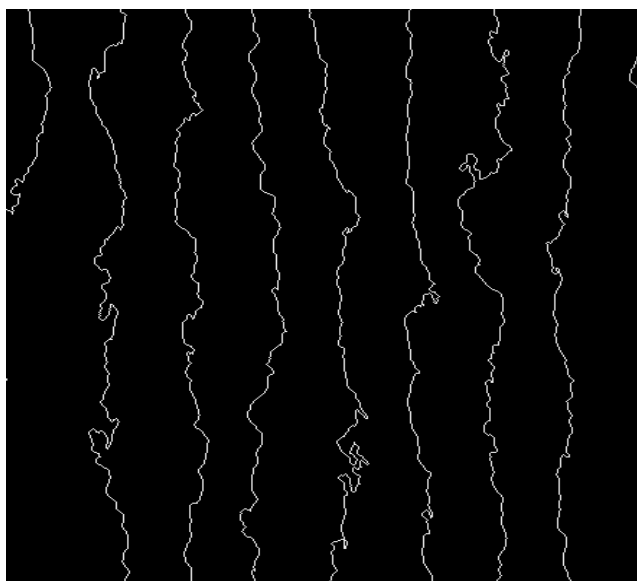


Obr 5.11. Interferogram bez zdvojenými linií získaný z obrázku 5.10

Odstranění zdvojených linií sice vede ke ztrátě informace o poloze středu interferenčního proužku, ale vzhledem k dostatečnému množství informací v celém interferogramu nedojte k výraznému ovlivnění statistického vyhodnocení interferogramu.

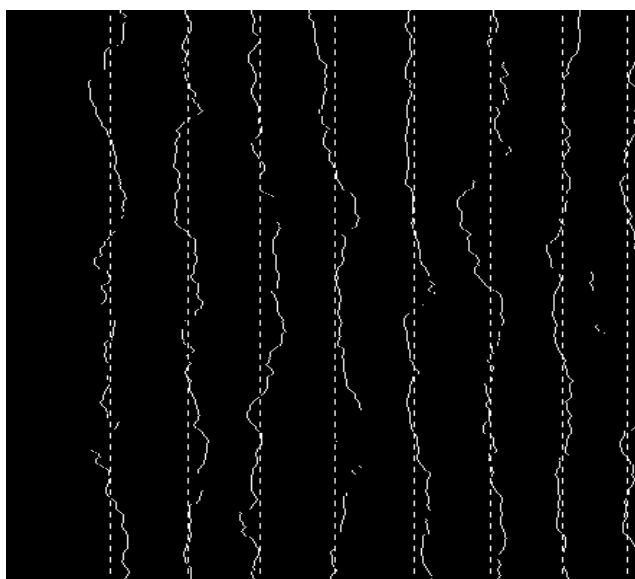
Druhý způsobem, kterým jsem použil pro získání jednopixelových linií je aplikace modifikovaného algoritmu z kapitoly 3.3.4, kde budeme považovat světlé proužky za objekty a budeme hledat jejich pravé vnitřní hranice. Ty obarvíme barvou, jenž se v interferogramu nevyskytuje. Ukončovací podmínku algoritmu musíme pozměnit. Algoritmus se ukončí pokud neexistuje pokračování hranice, nebo právě objevený bod hranice se nachází na okraji interferogramu. Jelikož hledáme pravé vnitřní hranice objektu, tak počáteční bod hranice se musí hledat zprava doleva a směr prohledávání okolí se použije ve směru hodinových ručiček. Tento algoritmus se aplikuje na všechny světlé proužky. Po té obarvíme všechny pixely, které mají jiný odstín šedi než právě objevené hranice, černou barvou. Tím dostaneme jednopixelové linie středů tmavých interferenčních proužků. Tato metoda má výhodu v tom, že přesně kopíruje hrany přechodu a tak vytvoří spojitě linie. Bylo by dobré odstranit zdvojená linie jako jsme to udělali u prvního způsobu. Jsou to totiž místa s potenciálním výskytem vady způsobené nečistotami na vzorku. Funkce používající tento algoritmus pro vyhledávání linií se nazývá *TMinInterfer.HledejMinima2* a nachází se v unitě *uAnalyz*. Při aplikaci funkce na separovaný interferogram vzniknou linie jako na obrázku 5.12.

V interferogramu se mohou také nacházet tzv. minoritní linie, které jsou natolik krátké, že statistiky z nich získané mohou být zavádějící. Je proto dobré je odstranit. Velikost hranice pro určení minoritní linie, byla v programu určena jako polovina z výšky interferogramu. Má-li vygenerovaný povrch vysokou drsnost, pak aplikací všech zatím zmíněných operací zpracování interferogramu vznikne separovaný interferogram, který může mít dva světlé proužky vzájemně spojeny úzkou štěrbinou v tmavém proužku. Pokud použijeme metodu sledování hranice, tak se projeví tato vada vyhledáváním velmi dlouhé jednopixelové linie. Tyto majoritní linie by ovlivnili statistické zpracování, proto jsou také z interferogramu vypuštěny. Velikost hranice pro určení majoritní linie jsem experimentálně určil jako 1,75 násobek výšky interferogramu. Funkce pro odstranění minoritních a majoritních linií přiloženého programu se jmenuje *TminInterfer.OdstranMinoritMajoritniLinie* a je umístěna v unitě *uAnalyz*.



Obr 5.12 Interferogram s vyznačenými jednopixelovými liniemi získanými metodou sledování hranice z interferogramu na obrázku 5.8.

Interferogram po odstranění minoritních linií z obrázku 5.11 je na obrázku 5.13. Tím jsou odhaleny středy interferenčních proužků a můžeme začít s jejich vyhodnocováním.



Obr. 5.13 Interferogram po odstranění minoritních a majoritních linií z obrázku 5.11 a vyznačení středních hodnotami linií.

5.3 Statistické zpracování jednopixelových linií

Nyní jsme již na konci úprav interferogramu. Máme linie, které charakterizují středy interferenčních proužků (Obr. 5.14). Linie jsou navzájem od sebe barevně odlišené, proto pro nás nebude problém pro každý bod určit ke které linii patří. Nastává tedy chvíle pro vyhodnocení statistických údajů vyplývajících z tvaru světlých jednopixelových linií.

Nejdříve musíme najít průměrnou vzdálenost p_{v_i} každé linie i od osy y . Linie jsou od sebe odlišené barvou, proto hodnotu p_{v_i} pro každou linii i vypočteme jako průměr x -vých souřadnic všech pixelů v interferogramu, které mají barvu linie i .

Nyní můžeme vypočítat průměrnou vzdálenost q mezi sousedními liniemi. Spočteme si vzdálenosti v_i mezi jednotlivými sousedními liniemi i a $i+1$. Teoreticky by měly být linie téměř ekvidistantní. Úpravami interferogramu však mohly některé proužky zmizet. Proto nemůžeme vypočítat hodnotu q jako průměr z hodnot v_i . Nejdříve musíme zjistit nejmenší vzdálenost mezi dvěma sousedními liniemi $v_{\min}$. Tu budeme považovat za jakousi vzdálenost blízkou ekvidistantě linií. Konečnou hodnotu ekvidistanty určíme jako průměr všech hodnot v_i , pro něž platí:

$$v_i / v_{\min} \approx 1, \quad (5.4)$$

kde v_i je vzdálenost mezi linií i a pravou sousední linií a $v_{\min}$ je nejmenší vzdálenost mezi sousedními liniemi.

Zavedeme tzv. koeficient vzdálenosti k_{v_i} pro každou linii i , který bude udávat kolik by se vešlo dalších linií mezi linií i a pravou zobrazenou sousední linií v interferogramu plus jedna. Nakonec určíme velikost parametru q a to takto:

$$\frac{\sum_{i=1}^{M-1} v_i}{\sum_{i=1}^{M-1} k_{v_i}} = q, \quad (5.5)$$

kde v_i je vzdálenost mezi i -tou linií a sousední pravou linií, k_{v_i} je koeficient vzdálenosti i -té linie a M je počet linií.

Ted' máme již vše potřebné, abychom vypočetli střední kvadratickou odchylku σ_h . Pro všechny body j patřící do některé z linie i vypočteme hodnotu místní odchylky p_j bodu j od průměrné hodnoty p_{v_i} linie i . Pak ze vztahů (2.12) nebo (2.13) vypočteme pro každé p_j hodnotu h_j . Nakonec pomocí rovnice (2.14) vypočteme střední kvadratickou odchylku σ_h .

Tím jsme získali výstupní statistickou veličinu pro interferogram o námi zvoleném střihu τ . Pak několikrát celý postup zpracování interferogramu získaného z interferenčního mikroskopu opakujeme pro různá τ a sestavíme graf závislosti $\sigma_h(\tau)$. Následují kroky popsané v kapitole 2.4.2 pro získání statistických hodnot zkoumaného povrchu.

Pokud jsme analyzovali povrch, který jsme před tím vygenerovali, můžeme výstupní hodnoty h_i interferogramu pro daný střih srovnat s hodnotami získanými přímo ze znalosti výšek povrchu. Hodnoty h_i z povrchu dostaneme pomocí definice (2.11).

6 POPIS PROSTŘEDÍ A STRUKTURY SOFTWARE PRO URČOVÁNÍ TOPOGRAFICKÝCH PARAMETRŮ POVRCHŮ

Součástí vypracování této diplomové práce je program (na přiloženém CD), který automaticky vyhledává středy temných interferenčních proužků interferogramu. K tomu používá metody popsané v kapitole 5. Po nalezení středů proužků provede statistické výpočty a výsledky sdělí uživateli. Nejdůležitější výstupní veličinou statistického zpracování je střední kvadratická odchylka σ_h . Ta se použije pro výpočet topografických parametrů zkoumaného povrchu. Pro možnost ověření správnosti výsledků byli do programu přidány funkce, které generují náhodně drsný povrch a simulují činnost interferenčního mikroskopu.

Nyní je na čase seznámit s prostředím a strukturou tohoto programu. Program byl vytvořen v objektově orientovaném vývojovém prostředí Delphi 7 (pro svoje spuštění je tedy potřeba mít toto prostředí nainstalované), který je podporován operačním systémem Windows. Součástí přiloženého programu je také soubor pZpracPovr.exe, který je spustitelný bez instalace programu Delphi 7. Tento soubor však pro svoje spuštění vyžaduje knihovnu qntintf70.dll, která je také součástí přiloženého CD.

Vytvořený program se skládá z 5 unit (uModel, uSimul, uAnalyz, fParamInter a mZpracPovr) a 2 formulářů. Unit uModel obsahuje funkce a vlastnosti potřebné k zacházení s modelem jako celkem. Unit uSimul je zaměřená na funkce a třídy, které se vztahují k části simulace (generování povrchu a získání interferogramu pomocí simulace interferenčního mikroskopu). Unit uAnalyz obsahuje funkce a třídy potřebné k analýze interferogramu (filtrace, aplikace funkce arg grad, bipolarizace separace, nalezení linií a statistické zpracování výsledků). FParamInter je pomocná unit propojená s formulářem fInfoInter, přes který může uživatel zadat některé parametry důležité pro vyhodnocování linií. Unit mZpracPovr je propojena s hlavním formulářem umožňující komunikaci programu s uživatelem. Nalezneme v ní definice akcí, při práci s formulářem.

6.1 Struktura softwaru

Unit uModel obsahuje definici základní třídy s názvem TModel, která dědí ze třídy TObject.

TModel má 7 privátních vlastností (FPovrch, FInterfer, FFilterInterfer, FArgGradInterfer, FBipoInterfer, FSeparInterfer a FMinInterfer), které charakterizují jednotlivé stupně interferogramu během analýzy. Ke každé z těchto 7 vlastností existují veřejné konstrukce read-only property, díky nimž můžeme číst informace o jednotlivých fázích analýzy. Dále obsahuje 39 veřejných metod (včetně konstruktoru a destruktoru), které zajišťují nejrůznější funkce modelu.

Všech 7 vlastností jsou typu třída, jejíž definice se nachází v unitách uSimul a uAnalyz. Názvy těchto tříd jsou: TPovrch, TInterfer, TFilterInterfer, TArgGradInterfer, TBipoInterfer, TSeparInterfer a TMinInterfer. Jednotlivé významy těchto tříd si vysvětlíme při popisu příslušných unit, do nichž tyto třídy patří.

Nyní si představíme metody třídy uModel:

Konstruktor je vždy volán při vytvoření instance třídy TModel a má za úkol vytvořit tuto třídu a přiřadit všem svým 7 vlastnostem hodnotu nil. Tímto bude vždy docíleno, že neexistující fáze interferogramu snadno poznáme, protože ukazatel na tuto fázi bude ukazovat na nil.

Destruktor má na starosti zavolat uvolnění všech 7 svých vlastností typu třída a nakonec uvolní i instanci (typu TModel), ze které je volán.

Funkce *Generuj* zajistí vytvoření instance FPovrch (typu TPovrch) (pokud již neexistuje) a volání funkce FPovrch.GenerujPovrch této instance, která zajistí vygenerování náhodně drsného

povrchu metodou plátů. Návrátová hodnota funkce je buď True (pokud nový povrch byl vygenerován) nebo False pokud se tak z nějakého důvodu nestalo. Funkce *Generuj* má 4 parametry (*aVyska*, *aSirka*, *aSigma*, *aKoefDef*), které předává funkci *FPovrch.GenerujPovrch*.

Funkce *PridejSumKPovrchu* zajistí, aby se volala funkce *FPovrch.PridejSum* (přidá šum k vygenerovanému povrchu v instanci *FPovrch*) pouze za předpokladů, že instance *FPovrch* existuje. Pokud neexistuje, tak se funkce ukončí a vygeneruje varovné hlášení. Návrátová hodnota je opět typu boolean. True se vrací, pokud došlo k přidání šumu k povrchu, jinak se vrací False. Funkce *PridejSumKPovrchu* má 2 parametry, které předávají funkci *Fpovrch.PridejSum*.

Funkce *Interferuj* zajistí vytvoření instance *FInterfer* (typu *TInterfer*) (pokud již neexistuje a pokud existuje instance *FPovrch*) a volání funkce *FInterfer.ProvedStrih* této instance, která vytvoří z povrchu v instanci *FPovrch* interferogram o zadaném střihu a zdvihu. Návrátová hodnota funkce je opět buď True (pokud vznikl interferogram) nebo False (pokud se tak z nějakého důvodu nestalo). Funkce *Interferuj* má 6 parametrů (*aStrih*, *aZdvih*, *aLambda*, *aNo*, *aN*, *aSvetlo*), které předává funkci *FInterfer.ProvedStrih*.

Funkce *NactiPovrch*, *NactiFilterInterfer*, *NactiArgGradInterfer*, *NactiBipoInterfer*, *NactiSeparInterfer* a *NactiMinInterfer* vytvoří příslušnou instanci (pokud neexistuje) a zavolá funkci této instance, která načte interferogram reprezentovaný bitmapovým obrázkem. Funkce vrací hodnotu True, pokud se příslušný interferogram nebo povrch načte, jinak vrací False. Tyto funkce mají jediný parametr typu string, který udává cestu k souboru typu bitmap.

Funkce *NactiInterfer* pracuje jako předchozí funkce, ale předá instanci také hodnoty střihu, zdvihu, vlnové délky, indexů lomů a typu interference prostřednictvím parametrů *aStrih*, *aZdvih*, *aVlnDel*, *aNo*, *aN* a *aSvetlo*.

Procedury *OdstranPovrchData*, *OdstranInterferData*, *OdstranFilterData*, *OdstranArgGradData*, *OdstranBipoData*, *OdstranSeparData*, *OdstranMinData* zavolají destruktory příslušné instance třídy a ukazateli na tuto třídu přiřadí nil.

Funkce *LinearniFiltrace*, *Median5* a *Median3x3* zkontroluje zda ukazatel *FInterfer* není roven nil, pokud ano, funkce se ukončí a vygeneruje příslušné hlášení chyby. Jinak vytvoří instanci *FFilterInterfer* (pokud neexistuje) a zavolá příslušnou funkci instance *FFilterInterfer*, která vyfiltruje daným způsobem interferogram z instance *FInterfer*. Funkce vrací True, pokud k filtraci dojde jinak vrací False.

Funkce *ArgGrad*, *Bipolarizuj*, *Separuj*, *VyznacMinima* (značení minim metodou detekce hrany ve směru x) a *VyznacMinima2* (značení minim metodou sledování hranice) zkontrolují, zda instance předchozího kroku analýzy není nilová, pokud ano, funkce se ukončí a vygeneruje příslušné hlášení chyby. Jinak vytvoří instanci (pokud neexistuje) a zavolá funkci vytvořené instance, která provede námi požadovaný krok analýzy. Funkce vrací True, pokud krok analýzy bude proveden jinak vrací False.

Funkce *OdstranZdvojLinie* a *OdstrMinorManorLinie* zkontrolují, zda ukazatel *FMinInterfer* není roven nil, pokud ano, funkce se ukončí a vygeneruje příslušné hlášení chyby. Jinak zavolá funkci *FMinInterfer.OdstranZdvojLinie* nebo *FMinInterfer.OdstranMinoritLinie*, které odstraní z interferogramu instance *FMinInterfer* zdvojené části linií nebo minoritní a manoritní linie. Funkce vrací True, pokud se krok analýzy zdaří, jinak vrací False.

Funkce *VyhodnotStatData* zkontroluje, zda ukazatel *FMinInterfer* není roven nil, pokud ano, funkce se ukončí a vygeneruje příslušné hlášení chyby. Jinak zavolá funkci *FMinInterfer.StatisZprac*, která provede statistiku interferogramu s vyznačenými liniemi středů interferenčních proužků. Funkce vrací True, pokud se statistika zdaří jinak vrací False. Funkce má 4 parametry (*aLambda*, *aNo*, *aN*,

aSvetlo), které jsou předány funkci `FMinInterfer.StatisZprac`.

Funkce `VypRozdPovrch`, `VypRozdInterfer`, `VypRozdFilterInterfer`, `VypRozdArgGradInterfer`, `VypRozdBipoInterfer`, `VypRozdSeparInterfer` a `VypRozdMinInterfer` zkontroluje, zda příslušné instance existují, pokud ne, funkce se ukončí a vygeneruje příslušné hlášení o chybě. Jinak zavolá funkci, která vypočte rozdělení dat příslušného interferogramu. Funkce vrací `True`, pokud se výpočet rozdělení zdaří jinak vrací `False`.

Funkce `VypRozdHPovrchu` zkontroluje zda ukazatel `FPovrch` není roven `nil`, pokud ano, funkce se ukončí a vygeneruje příslušné hlášení chyby. Jinak zavolá funkci `FPovrch.VypocetH`, která vypočte rozdělení hodnoty `h` povrchu pro zadaný stříh. Funkce vrací `True`, pokud se výpočet rozdělení zdaří jinak vrací `False`. Funkce má jeden parametr (`aStrih`), který se předá funkci `FPovrch.VypocetH`.

Procedura `UlozObr`, uloží obrázek `aImage` (typu `TImage`) do souboru, který se nachází na cestě `aFile` (typ `String`). `AImage` a `aFile` jsou parametry této procedury.

Unita uSimul obsahuje definice tříd `TPovrch` a `TInterfer`.

`TPovrch` je třída dědicí z `TObejct`. Obsahuje 12 privátních vlastností (`FVyska`, `FSirka`, `FSigma`, `FKoefDef`, `FPData`, `FHData`, `FRozdelH`, `FRozdelData`, `FSigmaH`, `FPrumerH`, `FSigmaDat`, `FPrumerDat`), které charakterizují povrch. Ke každé z těchto 12 vlastností existují veřejné konstrukce `only-read` property, díky nimž můžeme číst informace o studovaném povrchu. Dále obsahuje 9 veřejných metod (včetně konstruktoru a destrukturu), které zajišťují nejrůznější operace s povrchem.

Význam jednotlivých vlastností:

- *FVyska* – udává velikost výšky náhodně drsného povrchu v pixelech
- *FSirka* – udává velikost šířky náhodně drsného povrchu v pixelech
- *FSigma* – udává velikost parametru σ pro Gaussův generátor náhodných hodnot v pixelech
- *FKoefDef* – udává velikost koeficientu deformace povrchu generovaného metodou plátů
- *FPData* – jde o dynamické pole o rozměrech shodných s rozměry povrchu. Jednotlivé prvky pole udávají výšku povrchu v daném bodě v pixelech
- *FHData* – jde o dynamické pole s hodnotami $h = \text{FPData}[y, x + \text{střih}] - \text{FPData}[y, x]$
- *FRozdelH* – udává rozdělení veličiny h v pixelech
- *FRozdelDat* – udává rozdělení výšky povrchu v pixelech
- *FSigmaH* – udává střední kvadratickou odchylku veličiny h v pixelech
- *FPrumerH* – udává průměrnou hodnotu veličiny h v pixelech
- *FSigmaDat* – udává střední kvadratickou odchylku výšky povrchu v pixelech
- *FPrumerDat* – udává průměrnou výšku povrchu v pixelech

Význam metod:

- *GenerujPovrch* – Vygeneruje povrch o zadané výšce a šířce metodou plátů pomocí Gaussova rozdělení se zadanou střední kvadratickou odchylkou a koeficientem deformace. Vygenerovaný povrch se uloží do vlastnosti `FPData`.
- *PridejSum* – Přidá k povrchu (charakterizovaný polem `FPData`) šum metodou plátů pomocí Gaussova rozdělení se zadanou střední kvadratickou odchylkou a koeficientem deformace.
- *ZobrazPovrch* – Zobrazí do zvolené instance třídy `TImage` povrch (charakterizován polem `FDData`), jehož výška v bodě $[x, y]$ odpovídá hodnotě odstínu šedé v pixelu o souřadnicích $[y, x]$. Pokud maximální hodnoty výšky povrchu jsou vyšší než 255 nebo nižší než 0. Mají jednotlivé pixely odstín šedi přepočítaný, tak aby nejvyšší bod povrchu měl hodnotu 0 a nejvyšší hodnotu 255.
- *VypocetH* – Spočítá pro každý bod povrchu velikost veličiny h , jako rozdíl výšky v bodě $[x, y]$ a v bodě $[x, y + \text{střih}]$, kde velikost stříhu je předána funkci jako parametr. Výsledek je uložen do pole `FHData`.

- *RozdH* – Vypočte rozdělení veličiny h v dynamickém poli *FHData*, její střední kvadratickou odchylku a průměr. Výsledek je uložen do vlastností *FRozdelH*, *FPrumerH* a *FSigmaH*.
- *NactiPovrch* – Načte hodnoty odstínu šedi ze zvoleného souboru typu bitmap do pole *FPData*.
- *RozdelDat* – Vypočte rozdělení výšek povrchu z dynamického pole *FPData*, její střední kvadratickou odchylku a průměr. Výsledek je uložen do vlastností *FRozdelDat*, *FSigmatat* a *FPrumerDat*.
- *Konstruktor* – Vytvoří instanci třídy *TPovrch* a nastaví vlastnosti na hodnoty odpovídající neexistujícímu povrchu.
- *Destruktor* – Uvolní všechny dynamické pole a instance tříd pro něž je vlastníkem třída *TPovrch* a nakonec uvolní třídu *TPovrch*.

Všechny tyto metody jsou funkce a vracejíci hodnotu *True*, pokud splní požadovanou akci, jinak vrací *False*.

TInterfer je třída dědicí z *TObejct*. Obsahuje 11 privátních vlastností (*FIData*, *FStrih*, *FZdvih*, *FVlnDel*, *FRozdelDat*, *FSigmaDat*, *FPrumerDat*, *FWhite*, *FNo*, *FN*, *FSvetlo*), které charakterizují interferogram. Ke každé vlastnosti kromě *FWhite* existují veřejné konstrukce *only-read* property, díky nimž můžeme číst informace o studovaném interferogramu. Dále obsahuje 7 veřejných metod (včetně konstruktoru a destrukturu), které zajišťují nejrůznější operace s interferogramem.

Význam jednotlivých vlastností:

- *FIData* – jde o dynamické pole o rozměrech shodných s rozměry interferogramu. Jednotlivé prvky pole udávají hodnotu jasu interferogramu v daném bodě
- *FStrih* – udává hodnotu stříhu použitého pro interferenci v pixelech
- *FZdvih* – udává hodnotu zdvihu použitého pro interferenci v pixelech
- *FVlnDel* – udává velikost vlnové délky světla použitého pro interferenci v pixelech
- *FRozdelDat* – udává rozdělení odstínu šedi v interferogramu (*FIData*)
- *FSigmaDat* – udává střední kvadratickou odchylku jasu danou hodnotami v *FIData*
- *FPrumerDat* – udává průměrnou hodnotu jasu danou hodnotami v *FIData*
- *FWhite* – udává hodnotu jasu zastupující bílou barvu
- *FNo* – udává velikost indexu lomu vnějšího prostředí nad povrchem
- *FN* – udává velikost indexu lomu průhledného pevného tělesa
- *FSvetlo* – udává způsob vyhodnocení světelné vlny, jde-li o vlnu prošlou tělesem nebo odraženou od tělesa

Význam metod:

- *ProvedStrih* – Z daného povrchu vygeneruje interferogram o zadaném stříhu, zdvihu a vlnové délce. Ten pak uloží do dynamického pole *FIData*. Dále naplní vlastnosti *FStrih*, *FZdvih*, *FVlnDel*, *FNo*, *FN* a *FSvetlo* hodnotami užitými při stříhové interferometrii.
- *ZobrazInterfer* – Zobrazí interferogram (daný polem *FIData*) do instance třídy *TImage*. Hodnota *FIData* v bodě $[x,y]$ odpovídá hodnotě odstínu šedé v pixelu o souřadnicích $[y,x]$.
- *NactiInterfer* – Načte hodnoty odstínu šedi ze zvoleného souboru typu bitmap do pole *FIData*. Dále načte hodnoty stříhu, zdvihu, vlnové délky interferujícího světla, obou indexů lomu a způsob interference.
- *OtocO90Stup* – Otočí interferogram o 90° .
- *RozdDat* – Vypočte rozdělení jasu interferogramu z dynamického pole *FIData*, jeho střední kvadratickou odchylku a průměr. Výsledek je uložen do vlastností *FRozdelDat*, *FSigmatat* a *FPrumerDat*.

- *Konstruktor, Destruktor* – Tyto metody mají obdobné funkce jako v třídě TPovrch.

Opět všechny tyto metody jsou funkce a vracejíci hodnotu True, pokud splní požadovanou akci, jinak vrací False.

Unita uAnalyz obsahuje definice tříd TFilterInterfer, TArgGradInterfer, TBipoInterfer, TSeparInterfer a TMinInterfer.

TFilterInterfer je třída dědicí z TObjekt. Obsahuje 5 privátních vlastností (FFIData, FPouzFiltr, FRozdelDat, FSigmaDat, FPrumerDat), které charakterizují vyfiltrovaný interferogram. Ke každé vlastnosti existují veřejné konstrukce only-read property, díky nimž můžeme číst informace o studovaném vyfiltrovaném interferogramu. Dále obsahuje 8 veřejných metod (včetně konstruktoru a destrukturu), které zajišťují nejrůznější operace s filtrovaným interferogramem.

Vlastnosti mají obdobný význam jako je tomu u třídy TInterfer. Navíc je jen vlastnost *FPouzFiltr*, která udává druh použitého filtru při filtraci.

I metody mají stejný význam jako u třídy TInterfer. Jsou zde pouze navíc funkce: *LinearniFiltrace*, *Median5* a *Median3x3*. Ty vyfiltrují interferogram ze třídy TInterfer podle zvoleného filtru a výsledek uloží do dynamického pole FFIData.

TArgGradInterfer, *TBipoInterfer* a *TSeparInterfer* jsou třídy dědicí z TObjekt. Každá obsahuje 4 privátní vlastnosti, které charakterizují interferogram v dané fázi analýzy. Ke každé vlastnosti existují veřejné konstrukce only-read property, díky nimž můžeme číst informace o studovaném interferogramu. Dále obsahuje 6 veřejných metod (včetně konstruktoru a destrukturu), které zajišťují nejrůznější operace s interferogramem.

Vlastnosti i metody těchto tříd mají opět obdobný význam jako je tomu u třídy TInterfer. Funkce *ArgGrad* aplikuje na vyfiltrovaný interferogram třídy TFilterInterfer funkci arg grad. Výsledek uloží do dynamického pole FArgGradData. Funkce *Bipolar* bipolarizuje interferogram z předchozího kroku a uloží ho do dynamického pole FBipoData. Obdobně provede funkce *Separace* separaci interferogramu z předchozího kroku a uloží ho do pole FSeparData.

TMininterfer je třída dědicí z TObjekt. Obsahuje 19 privátních vlastností (FMIData, FBarvy, FMaxLin, FOdZdoLinie, FOdMinorLinie, FStatisZprac, FRozdelDat, FSigmaDat, FPrumerDat, FRozdelP, FRozdelH, FPocetLinii, FLinie, FStrVzdLinii, FSigma_P, FPrumer_P, FSigma_H, Fsigma_Povrch, FPouzMetHledMin), které charakterizují interferogram s vyznačenými středy interferenčních proužků. K 17 vlastnostem existují veřejné konstrukce only-read property, díky nimž můžeme číst informace o interferogramu s vyznačenými středy proužků a o statistických výsledcích celé analýzy. Dále obsahuje 11 veřejných metod (včetně konstruktoru a destrukturu), které zajišťují nejrůznější operace s interferogramem.

Význam jednotlivých vlastností:

- *FMIData* – jde o dynamické pole o rozměrech shodných s rozměry interferogramu s vyznačenými středy proužků. Jednotlivé prvky pole udávají hodnotu jasů interferogramu v daném bodě
- *FBarvy* – jde o pole s 256 prvky typu boolean, prvek i indikuje, zda se nachází v interferogramu pixel o hodnotě odstínu šedi rovné i
- *FMaxLin* – udává nejtmavší odstín nalezených linií
- *FOdZdoLinie* – proměnná typu boolean, která udává zda byli z interferogramu odstraněny zdvojené linie
- *FOdMinorManorLinie* – proměnná typu boolean, která udává zda byli z interferogramu odstraněny minoritní a manoritní linie
- *FStatisZprac* – proměnná typu boolean, která udává zda byl interferogram statisticky zpracován
- *FRozdelP* – udává rozdělení vzdáleností linií od svých středních hodnot v pixelech
- *FRozdelH* – udává rozdělení veličiny h získané pomocí stříhové interferometrie v

pixelech

- *FPocetLinii* – udává celkový počet linií nacházejících se v interferogramu
- *FLinie* – pole ukazatelů na třídu *TLine* (třída nesoucí vlastnosti jedné linie)
- *FStrVzdLinii* – udává střední vzdálenost mezi liniemi v pixelech
- *FSigma_P*, *FSigma_H* – udává střední kvadratickou odchylku veličiny p , h v pixelech
- *FPrumer_P* – udává průměrnou hodnotu veličiny p v pixelech
- *FSigma_Povrch* – udává střední kvadratickou odchylku povrchu v pixelech, pokud by platilo, že $\tau > \tau^*$
- *FRozdelDat*, *FSigmaDat*, *FPrumerDat* – tyto vlastnosti mají podobný význam jako v třídě *TInterfer*
- *FpouzMetHledMin* – udává způsob, kterým byly nalezeny středy temných interferenčních proužků (detekce hran ve směru osy x , sledování hranice)

Význam metod:

- *HledejMinima* – Vyhledá v separovaném interferogramu středy interferenčních proužků metodou detekce hran ve směru osy x . Výsledek se uloží v dynamickém poli *MIData*.
- *HledejMinima2* – Vyhledá v separovaném interferogramu středy interferenčních proužků metodou sledování hranice. Výsledek se uloží v dynamickém poli *MIData*.
- *OdstranZdvojLin* – Odstraní z interferogramu (uloženém v *MIData*) rozdvojené části linií.
- *OdstranMinorManorLinie* – Odstraní z interferogramu (uloženém v *MIData*) krátké a dlouhé linie.
- *StatisZprac* – Provede statistickou analýzu interferogramu s vyznačenými středy interferenčních proužků. Výsledky uloží do příslušných vlastností třídy *TMinInterfer*.
- *VykresliStrHodLin* – Zobrazí v instanci třídy typu *TImage* přímky, charakterizující střední hodnoty linií.
- *Konstruktor*, *Destruktor*, *ZobrazMinInterfer*, *NactiMinInterfer* a *RozdDat* mají obdobnou funkci, jako ve třídě *TInterfer*.

Unita *mZpracPovr* je propojena s formulářem *mZpracPovr*, který zajišťuje komunikaci mezi uživatelem a jednotlivými funkcemi a procedurami v ostatních unitách.

Unita *fParamInter* je spojena s formulářem *fInfoInter*, přes něj může uživatel zadat některé parametry důležité pro vyhodnocování linií.

V unitách *UModel*, *USimul* a *UAnalyz* se veškeré výpočty provádí v pixelech. Proto se v unitě *mZpracPovr* nachází funkce pro převod délkových rozměrů z pixelů na mikrometry eventuálně nanometry. Při převodu se používá přepočtení podle vzorkovací vzdálenosti použité při laboratorních pokusech na FSI VUT v Brně. Tento poměr vychází z faktu, že 13 pixelům odpovídá 10 μm .

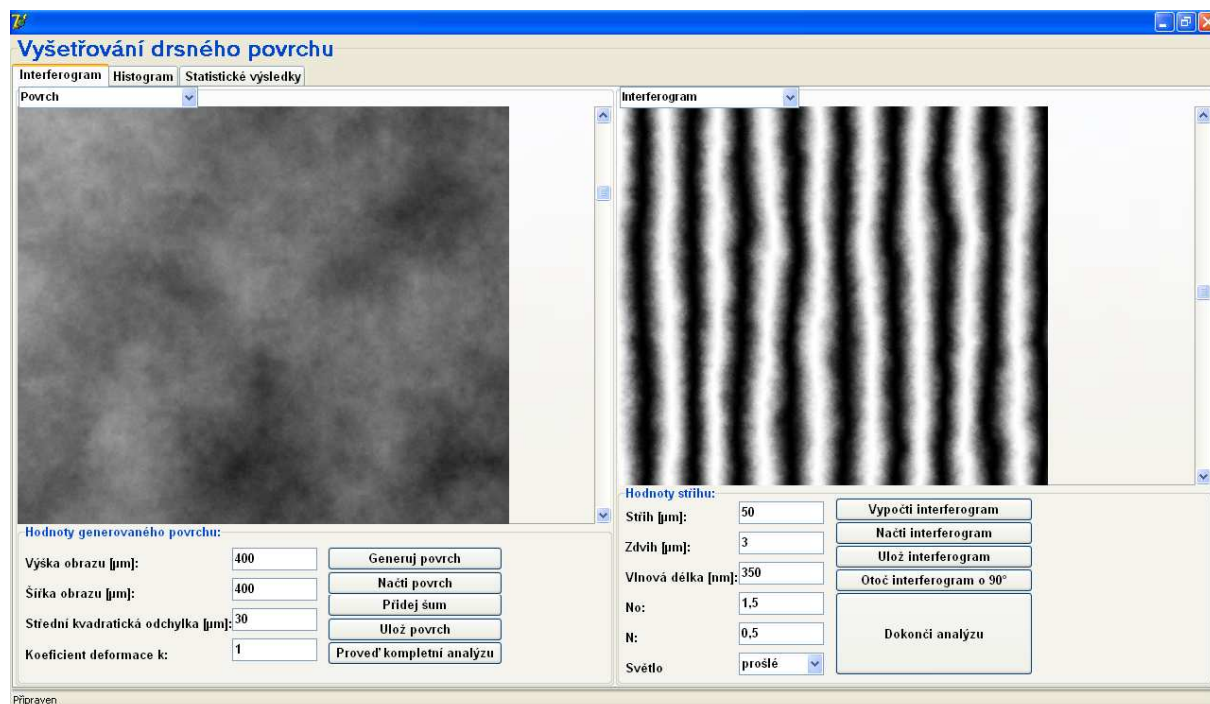
Program je vytvořen tak, aby instance třídy *TModel* vždy obsahovala jednotlivé kroky analýzy a informace o nich jen pro jeden interferogram a k němu příslušný povrch. Proto například při načtení interferogramu ze souboru do libovolné fáze analýzy dojde k odstranění všech ostatních kroků analýzy.

6.2 Seznámení s uživatelským prostředím softwaru

Software se skládá ze tří záložek – Interferogram, Histogram a Statistické výsledky. Nejdříve si probereme první záložku s názvem *Interferogram*. Ta je zobrazena na obrázku 6.1. Je rozdělena na dvě části, každá z nich nezávisle na sobě zobrazuje povrch nebo vybraný interferogram podle zvolené položky v combo boxu vlevo nahoře příslušné části. Na obrázku 6.1 je zvoleno v levé části zobrazení povrchu a v pravé zobrazení interferogramu. Můžeme v combo boxech zvolit jednu z těchto možností: Povrch, Interferogram, Filtrovaný interferogram, Interferogram po arg grad, Bipolarizovaný

interferogram, Separovaný interferogram a Interferogram s minimi.

Při zvolení položky *povrch* v combo boxu se v horní části zobrazí povrch, pokud je k dispozici. Jinak se zobrazí nápis „Povrch není k dispozici.“ Povrch se zobrazuje ve 256 odstínech šedi. Odstín šedi v daném bodě je roven výšce povrchu v tomto bodě, pokud všechny hodnoty výšek leží v intervalu $<0;255>$. V opačném případě se hodnoty odstínu šedi jednotlivých pixelů přepočítají tak, aby černá barva zobrazovala nejnižší bod a bílá barva nejvyšší bod povrchu. Ve spodní části se nalézají tlačítka: Generuj povrch, Načti povrch, Přidej šum, Ulož povrch a Proved' kompletní analýzu.



Obr. 6.1. Záložka Interferogram se zobrazeným povrchem a interferogramem

Jednotlivé funkce tlačítka jsou popsány níže:

- *Generuj povrch* – Vygeneruje povrch pomocí metody plátů o parametrech zadaných v editačních polích vlevo od tohoto tlačítka. Tento povrch se zobrazí v horní části.
- *Načti povrch* – Slouží k načtení povrchu ze souboru typu BMP. Po stisku tohoto tlačítka se objeví dialogové okno, které slouží k vyhledání BMP souboru na disku.
- *Přidej šum* – Po jeho stisku se vygeneruje nový povrch metodou plátů o stejných rozměrech jako má povrch k němuž chceme přidat šum. Ke generování se použijí parametry v editačních polích pro střední kvadratickou odchylku a koeficient deformace k. Tento nový povrch se přičte k tomu starému. Vznikne tak povrch zatížený šumem.
- *Ulož povrch* – Po stisku tlačítka se objeví dialogové okno, kde zadáme cestu a název pro uložení povrchu do souboru typu BMP.
- *Proved' kompletní analýzu* – Po stisku tohoto tlačítka se provede celková analýza povrchu, včetně statistického zpracování. Parametry pro generování interferogramu, výběr filtru a výběr metody hledání hran se stanoví z editačních polí nebo zaškrtnutých políček zobrazovaných při výběru položky „Interferogramm“, „Filtrovaný interferogram“ nebo „Interferogram s minimi“ v příslušném combo boxu.

Poslední tři tlačítka jsou k dispozici pouze za předpokladu, že nějaký povrch máme k dispozici.

Při zvolení položky *Interferogram* v combo boxu se v horní části zobrazí interferogram, který jsme buď načetli ze souboru nebo jsme získali pomocí simulace z povrchu, nebo hlášení „Interferogram není k dispozici.“. Ve spodní části se nalézají tlačítka: Vypočti interferogram, Načti interferogram, Ulož interferogram, Otoč interferogram a Dokonči analýzu.

Jednotlivé funkce tlačítka jsou popsány níže:

- *Vypočti interferogram* – Toto tlačítko je aktivní pouze, je-li v modelu k dispozici informace o povrchu. Po jeho stisku se vygeneruje interferogram z povrchu, který jsme získali ze záložky „Povrch“ načtením nebo generováním. Pro získání interferogramu jsou použity parametry zadané v editačních polích vlevo od tlačítka.
- *Načti interferogram* – Slouží k načtení interferogramu ze souboru typu BMP. Po stisku tohoto tlačítka se objeví dialogové okno, které slouží k vyhledání BMP souboru na disku.
- *Ulož interferogram* – Po stisku tlačítka se objeví dialogové okno, kde zadáme cestu a název pro uložení interferogramu do souboru typu BMP.
- *Otoč interferogram* – Otočí načtený nebo vygenerovaný interferogram o 90°.
- *Dokonči analýzu* – Po stisku tohoto tlačítka se dokončí celková analýza interferogramu, včetně statistického zpracování. Výběr filtru a metody hledání hranice se stanoví ze zaškrťovacích políček zobrazovaných při výběru položky „Filtrovaný interferogram“ nebo „Interferogram s minimi“ v příslušném combo boxu.

Poslední tři tlačítka jsou k dispozici pouze za předpokladu, že interferogram máme k dispozici.

Při zvolení položky *filtrovaný interferogram* v combo boxu se v horní části zobrazí interferogram, který jsme buď načetli ze souboru nebo získali filtrací původního interferogramu, nebo hlášení „Filtrovaný interferogram není k dispozici.“. Ve spodní části se nalézají tlačítka: Filtruj, Načti interferogram, Ulož interferogram a dokonči analýzu. Dále jsou zde zaškrťovací políčka, pro zvolení druhu filtrace. Můžeme volit mezi lineární filtrací, filtrací typu median 5 nebo median 3x3.

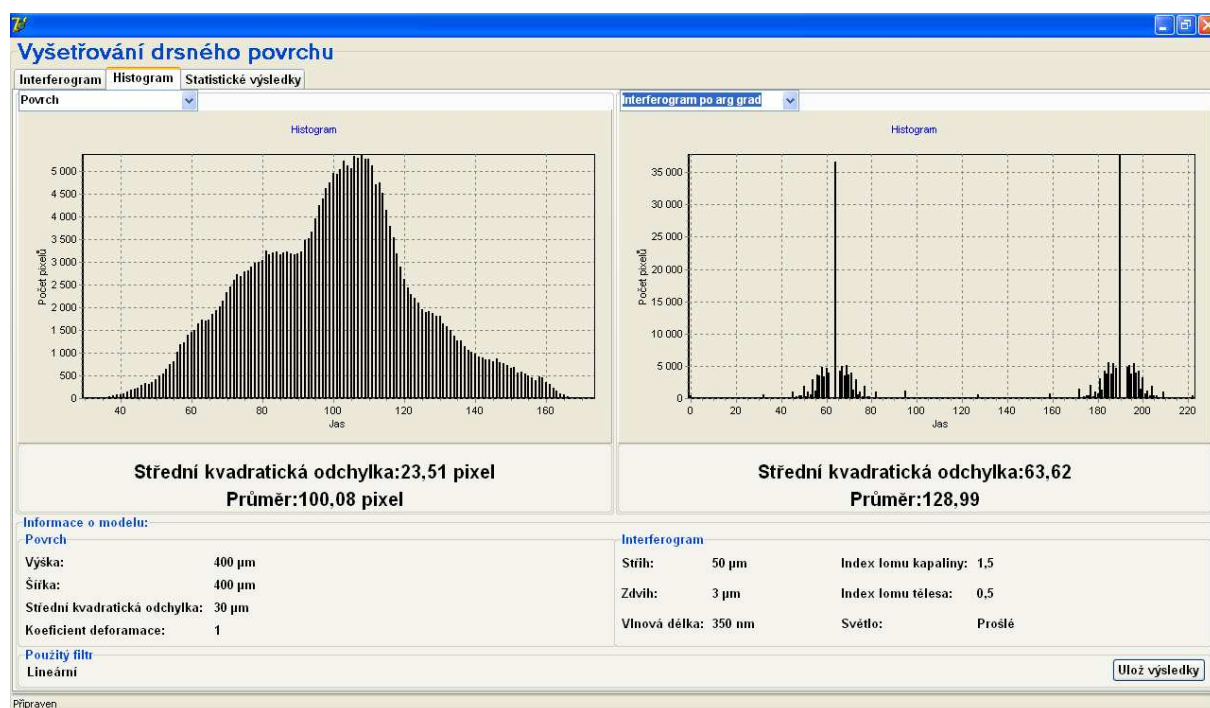
Tlačítka *Načti interferogram*, *Ulož interferogram* a *Dokonči analýzu* mají obdobnou funkci jak bylo popsáno výše při aplikaci na interferogram. Tlačítko *Filtruj* je aktivní pouze za předpokladu, že máme interferogram v modelu k dispozici. Po jeho stisku se provede filtrace interferogramu podle zaškrtnutého políčka nad tlačítkem.

Při zvolení položky *Interferogram po arg grad*, *Bipolarizovaný interferogram* nebo *Separovaný interferogram* v combo boxu se v horní části zobrazí interferogram v příslušném kroku zpracování, který jsme buď načetli ze souboru nebo získali z předchozího stavu interferogramu, nebo hlášení, že daný interferogram není k dispozici. Ve spodní části se nalézají tlačítka: *Načti interferogram* a *Ulož interferogram*, které mají obdobný význam, jaký jsme popsali výše. Dále se zde pro jednotlivé položky combo boxu nalézají tlačítka: *Aplikuj funkci arg grad*, *Bipolarizuj* a *Separuj*, které jsou aktivní jen, když existuje v modelu interferogram v předchozím kroku zpracování. Tyto tlačítka na ně aplikují příslušnou funkci.

Poslední položka combo boxu je *Interferogram s minimi*. Opět v horní části je zobrazen interferogram s vyznačenými středy interferenčních proužků nebo nápis „Interferogram s minimi není k dispozici“. Dole nalezneme tlačítka *Načti interferogram* a *Ulož interferogram*, jejichž funkce je známá z předchozího výkladu. Pak zde můžeme najít vlevo dole jedno z tlačítek: *Hledej minima*, *Odstraň zdvojené linie* nebo *Odstraň chybné linie*. Zobrazují se podle toho jaký krok při zpracování interferogramu s minimi by měl následovat. První z nich vytvoří ze separovaného interferogramu

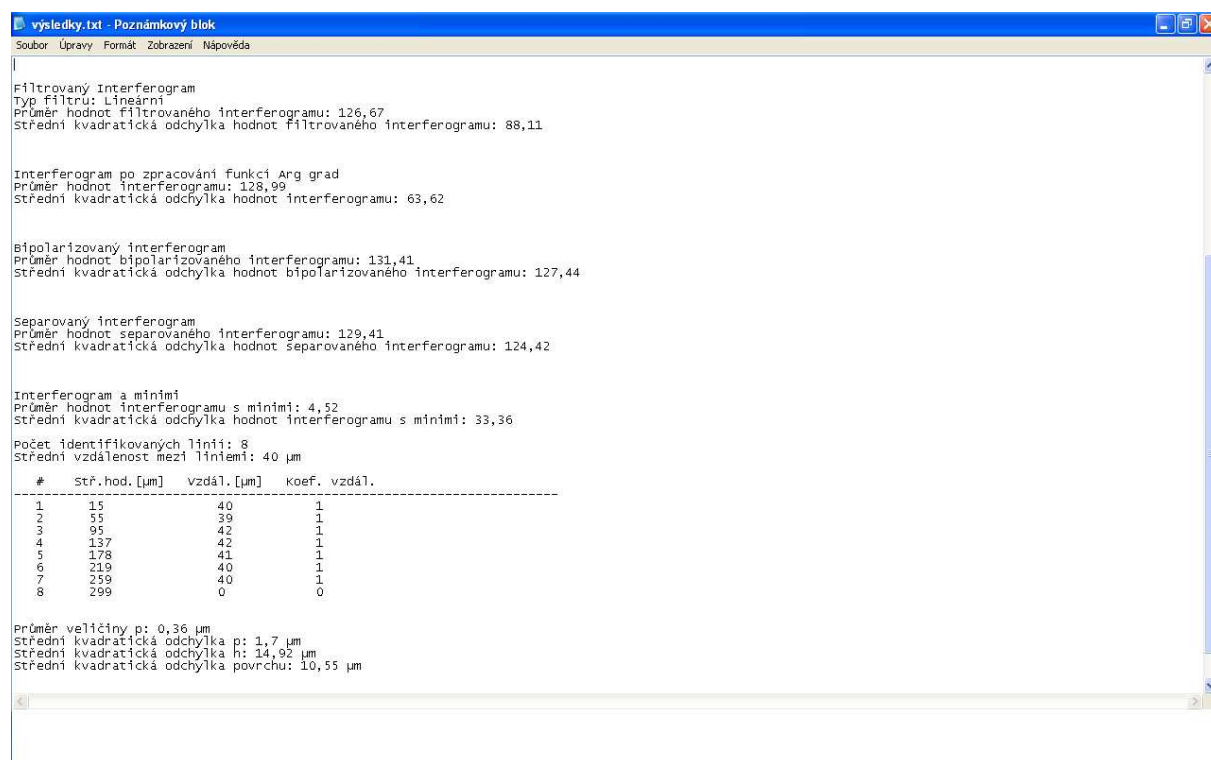
interferogram s vyznačenými jednopixelovými liniemi středů tmavých interferenčních proužků. Druhé tlačítko z tohoto interferogramu odstraní rozdvojené části linií. Poslední odstraní minoritní popřípadě majoritní linie a provede statistiku proužků. Dále zde najdeme zaškrtačací tlačítka pro volbu metody vyhledávání jednopixelových linií.

Přepneme-li horní záložku na *Histogram*, zobrazí se opět dvě na sobě nezávislé okna (Obr.6.2). Ty zobrazují podle zvolené položky v combo boxu příslušné histogramy ve své horní části nebo hlášení, že interferogram (povrch) není k dispozici. Je-li zvolena položka *Povrch*, tak se zobrazí histogram popisující rozdělení výšek v pixelech vygenerovaného nebo načteného povrchu. Pro ostatní volby v combo boxu se zobrazuje histogram popisující rozdělení jasu příslušných interferogramů. Pod histogramem je uvedena střední kvadratická odchylka a průměr příslušného histogramu. Spodní část obsahuje důležité informace o zkoumaném modelu, pokud jsou k dispozici. Vpravo dole se nalézá tlačítko s nápisem *Ulož výsledky*. Po jeho stisku se objeví dialogové okno, kde zvolíme název souboru a místo na disku, kam se výsledky mají uložit. Ty se uloží do souboru typu txt (Obr. 6.3).

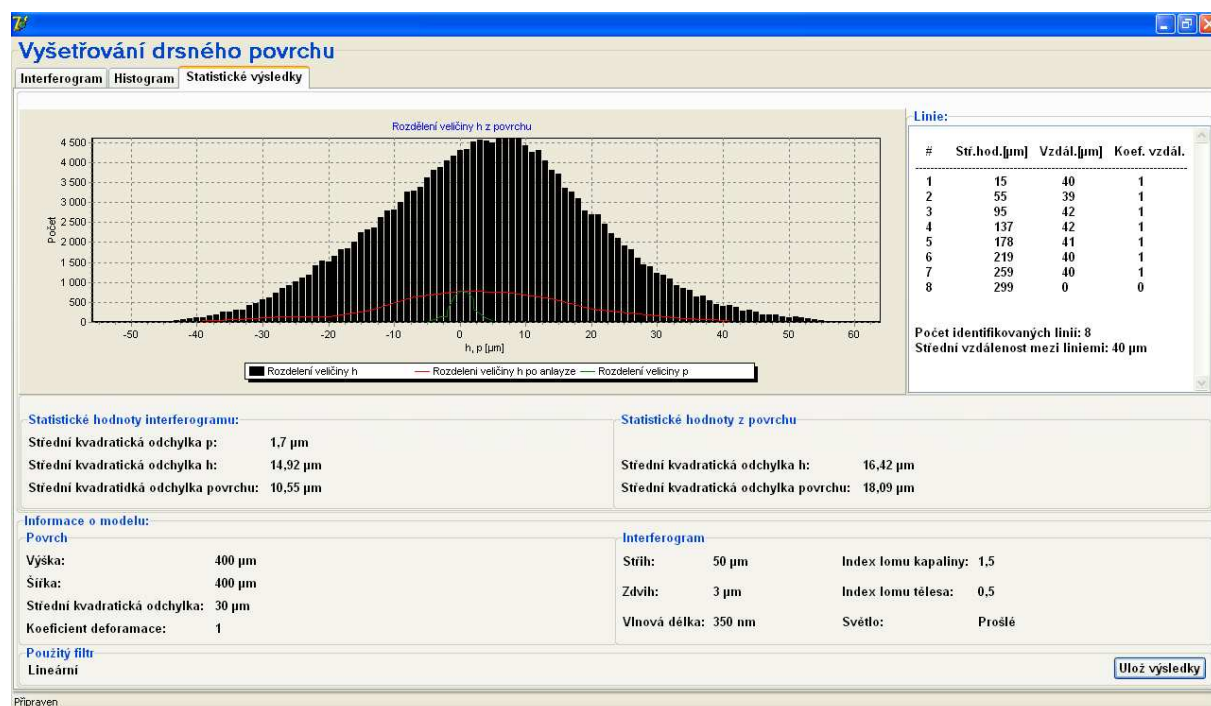


Obr 6.2 Záložka Histogram se zobrazeným povrchem a interferogramem po aplikování funkce $\arg \text{grad}$.

Poslední horní záložka nese název *Statistické výsledky* (Obr. 6.4). Spodní část je identická se spodní částí záložky *Histogram*. Vlevo nahoře najdeme graf, v něm jsou zakresleny tři histogramy: rozdělení veličiny h (vypočítané přímo z povrchu), rozdělení veličiny h po analýze (spočtené z linií v interferogramu s vyznačenými minimi) a rozdělení veličiny p (spočtené z linií v interferogramu s vyznačenými minimi). Vpravo od tohoto grafu se nalézají informace o jednopixelových liniích nacházejících se v interferogramu poslední fáze analýzy. Pod grafem vlevo nalezneme nejdůležitější informace zjištěné analýzou. Je to střední kvadratická odchylka veličiny p a h (σ_p a σ_h). σ_h se použije pro zjištění závislosti $\sigma_h(\tau)$. Dále zde nalezneme střední kvadratickou odchylku povrchu σ za předpokladu, že analýza interferogramu byla provedena pro střih $\tau > \tau^*$, kde τ^* značí začátek oblasti saturace. Vpravo o těchto tří údajů, pokud jsme povrch generovali programem, jsou pro srovnání uvedeny střední kvadratická odchylka h a střední kvadratická odchylka povrchu, které jsou vypočítány přímo z výšek studovaného povrchu.



Obr. 6.3 Ukázka uspořádání informací o modelu uložených do souboru typu txt.



Obr 6.4 Záložka Statistické výsledky

7 ZÁVĚR

Cílem projektu bylo navrhnout metody a algoritmy zpracování obrazu pro určení topografických parametrů povrchu a vytvořit konkrétní řešení v objektově orientovaném vývojovém prostředí Delphi.

Při realizaci těchto cílů se vycházelo z metody stříhové interferometrie, která byla vyvinuta na Ústavu fyzikálního inženýrství Fakulty strojního inženýrství VUT v Brně. Jedná se o optickou metodu, její bližší popis byl uveden v kapitole 2. V úvodu této kapitoly byly uvedeny vlastnosti, které musí splňovat studovaný povrch pro použití této metody. V kapitole dvě byla také naznačena základní fyzikální podstata dalších dvou optických metod.

Třetí kapitola byla zaměřena na popis algoritmů používaných při zpracovávání digitalizovaného obrazu. Jsou zde vybrány a popsány algoritmy, které by se daly využít ke zpracování interferogramu (výstupní obraz interferenčního mikroskopu používaného při metodě stříhové interferometrie) pro následné zjištění statistických hodnot povrchu pomocí stříhové interferometrie. Nalezneme zde možnosti jak reprezentovat digitalizovaný obraz v počítači, postupy předzpracování obrazu a algoritmy potřebné k analýze interferenčních proužků (získání hran a oddělení jednotlivých objektů obrazu).

Čtvrtá kapitola popisuje možnosti generování náhodně drsného povrchu a simulace interferenčního mikroskopu. Díky těmto algoritmům můžeme získat srovnání parametrů vygenerovaného povrchu s parametry zjištěnými pomocí algoritmů z kapitoly pět.

Pátá kapitola na základě vybraných algoritmů z kapitoly tři popisuje konkrétním návrh metod pro získání parametrů povrchu metodou stříhové interferometrie.

Součástí práce je také program, který automaticky zpracovává obraz interferogramu. Pro účel možnosti ověření výsledků je do programu přidána simulační část v unitě uSimul, která je schopna vytvořit náhodně drsný povrch a simulovat funkci interferenčního mikroskopu k získání interferogramu. Popis tohoto programu byl proveden v kapitole 6.

Bohužel v současné době nejsou k dispozici vzorky získané pomocí interferenčního mikroskopu, proto ověření správnosti použitých algoritmů je prováděno jen pomocí nasimulovaných povrchů metodou plátů z vytvořeného programu. Bylo vygenerováno několik povrchů s různými hodnotami vstupů generátoru. Po té se z těchto povrchů získalo pomocí simulace interferenčního mikroskopu interferogramy pro různé hodnoty vlnové délky, zdvihu a stříhu. Tyto interferogramy se nechaly zpracovat vytvořeným programem. Nakonec se provedlo srovnání veličiny σ_h získané pro daný stříh přímo z povrchu a z interferenčních proužků. Dosažené výsledky získané oběma způsoby se lišily okolo deseti procent. To odpovídá hodnotám, které byly získány v laboratoři FSI VUT v Brně pomocí ZPI interferenčního mikroskopu a programu SOFO DIPS 4.0.

SEZNAM POUŽITÉ LITERATURY

- [1] OHLÍDAL, M. *Optické metody hodnocení textury povrchu v mikrometrové oblasti* [online]. Brno. 2004. [cit. 20. 4. 2009]. Dostupný z: http://gps.fme.vutbr.cz/STAH_INFO/46_Ohlidal_VUTBR.pdf.
- [2] OHLÍDAL, M.; Ohlidal I.; Druckmüller, M.; Franta, D. *A method of shearing interferometry for determining the statistical quantities of randomly rough surfaces of solids*. Pure and Applied Optics, Volume 4, Issue 5, pp. 599-616(1995) . Printed in the UK.
- [3] OHLÍDAL, I.; Franta, D.; Klapetek, P. *Měření nanodrsnosti pomocí optických metod a mikroskopie atomové síly*. [online]. Brno. [cit. 20. 4. 2009]. Dostupný z: http://gps.fme.vutbr.cz/STAH_INFO/45_Ohlidal_MUNI.pdf.
- [4] LIPTÁK, O.; Békés, J. *Technológia výroby: obrábanie*. Bratislava : ALFA, 1979. 444 s. ISBN 63-565-79.
- [5] BENNETT, J. M. *Appl. Opt.* **15** 2705-21. 1976 .
- [6] OHLÍDAL, I.; Navrátil, K. *Appl. Opt.* **26** 2690-5. 1985.
- [7] OHTSUBO, J. *Measurement of Diamond-Turned Metal Surfaces by Light Scattering*. SPIE, **558** Inverse Optics II, 1985, 80-84.
- [8] LÉGER, D.; Perrin, J.C. *Real-time measurement of surface roughness by correlation of speckle patterns*. *J.Opt.Soc.Am.* **66**, 1976, 1210-1217.
- [9] LÉGER, D.; Mathieu, E.; Perrin, J.C. *Optical Surface Roughness Determination Using Speckle Correlation Technique*. *Appl. Optics.* **14**, 1975, 812-877.
- [10] OHLÍDAL, I.; Navrátil, K. *Analysis of the basic statistical properties of random rough surfaces by shearing interferometry*. *Appl. Opt.* **24**, 2690-2695 (1985).
- [11] OHLÍDAL, I.; Navrátil, K.; Ohlidal, M.; Druckmüller, M. *Characterization of the basic statistical properties of very rough surfaces of transparent solids by immersion shearing interferometry*. *Appl. Opt.* **33**, 7838-7845 (1994) .
- [12] OHLÍDAL, I.; Ohlidal, M.; Franta, D.; Michálek, A. *Method of shearing interferometry for characterizing non-gaussian randomly rough surfaces*. SPIE, **2775** , 442-451.
- [13] PTÁČEK, P. *Střihová interferometrie náhodně drsných povrchů*. Brno, 1998. Diplomová práce na Masarykově univerzitě.
- [14] SVOBODA, P; Brandejs, J; Prokeš, F. *Základy konstruování*. 2. vyd. Brno: CERM, 2003. ISBN 80-7204-306-4.
- [15] HLAVÁČ, V; Šonka, M. *Počítačové vidění*. Praha: Grada, 1992. ISBN 80-85424-67- 3
- [16] MAŘÍK, V; Štěpánková, O; Lažanský, J. *Umělá inteligence 2*. 1. vyd. Praha: Academia, 1997. ISBN 80-200-0504-8.

- [17] ŠŤASTNÝ, J. *Netradiční metody a algoritmy pro rozpoznávání objektů technologické scény*. Habilitační a inaugurační spisy. Brno: VUTUM, 2006. ISSN 1213- 418X, 30 pp.
- [18] SOCHOR, J; Žára, J. *Algoritmy počítačové grafiky*. Praha: Vydavatelství ČVUT, 1993. 258 s. ISBN 80-01-00949-1.
- [19] *Wikipedie: Otevřená encyklopedie: RGB* [online]. c2009 [citováno 28. 05. 2009]. Dostupný z : <http://cs.wikipedia.org/w/index.php?title=RGB&oldid=3962073> .
- [20] ŽÁRA, J., Beneš, B., Felkel, P. *Moderní počítačová grafika*. Praha: Computer Press, Praha, 1998.
- [21] KOTEK, Z., Mařík, V. *Metody rozpoznávání a jejich aplikace*. Praha: Academia, 1993.
- [22] *Wikipedie: Otevřená encyklopedie: Cannyho hranový detektor* [online]. c2008 [citováno 28. 05. 2009]. Dostupný z: http://cs.wikipedia.org/w/index.php?title=Cannyho_hranov%C3%BD_detektor&oldid=2166951 .
- [23] JELÍNEK, I., Sochor, J. *Algoritmy počítačové grafiky*. Praha: ČSVTS, 1989.
- [24] HARIHARAN, P. *Optical interferometry*. 2nd edition. Sidney: Academic Press - Elsevier, 2003.
- [25] PELIKÁN, J. *Počítačová grafika I* [online]. [citováno 28. 05. 2009]. Dostupný z: <http://cgg.mff.cuni.cz/~pepca/lectures/pgr003.html> .
- [26] BENEŠ, B. *Fraktály*. Bajt č.3, 15-17 (1991).
- [27] HARUYAMA, S. *Using stochastic modelling for texture generation*. IEEE CG&A March (1984).