



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Aplikace pro vyhodnocování průměrů vláken ze snímků vláknenných struktur

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Vladimír Hrdina**

Vedoucí práce: Ing. Petr Kretschmer





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Application for measuring of fibers diameters from pictures Scanning Electron Microscope

Diploma thesis

Study programme: N2612 – Electrical Engineering and Informatics

Study branch: 1802T007 – Information Technology

Author: **Bc. Vladimír Hrdina**

Supervisor: Ing. Petr Kretschmer



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Vladimír Hrdina**
Osobní číslo: **M13000189**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Informační technologie**
Název tématu: **Aplikace pro vyhodnocování průměrů vláken ze snímků vlá-
kenných struktur**
Zadávající katedra: **Ústav nových technologií a aplikované informatiky**

Z á s a d y p r o v y p r a c o v á n í :

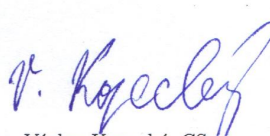
1. Seznamte se s problematikou zpracování obrazových informací (zaměřte se na problematiku hranových detektorů a problematiku úpravy rastrových obrázků)
2. Analyzujte vhodné postupy, které umožní řešení problému
3. Vytvořte aplikaci, která splňuje následující:
 - a) Načítá obrazová data vlákenných struktur
 - b) Aplikuje vhodný hranový detektor
 - c) Umožňuje detekci vláken v místě zadaném uživatelem
 - d) Umožňuje detekci vláken v automatickém režimu
 - e) Umožňuje export dat k dalšímu statistickému zpracování
 - f) Proveďte vlastní statistické zpracování četností výskytu průměrů vláken
 - g) Umožňuje export obrázku se zobrazenými (případně popsány) detekovanými průměry vláken
4. Ověřte správnou funkci aplikace a zhodnoťte dosažené výsledky na reálných datech

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **30 - 40 stran**
Forma zpracování diplomové práce: **tištěná/elektronická**
Seznam odborné literatury:

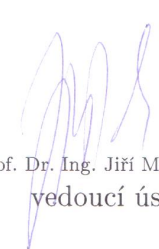
- [1] PLÍVA, Zdeněk; DRÁBKOVÁ, Jindra. Metodika zpracování diplomových, bakalářských a vědeckých prací na FM TUL. Vyd. 1. Liberec: Technická univerzita, 2007. 40 s. Dostupné z WWW:
<http://www.fm.tul.cz/files/jak_psat_DP.pdf>. ISBN 978-80-7372-189-3.
[2] SONKA, M.; HLAVAC, V.; BOYLE, R.: Image Processing, Analysis, and Machine Vision, PWS Publishing, 1998, ISBN 0-534-95393-X

Vedoucí diplomové práce: **Ing. Petr Kretschmer**
Ústav nových technologií a aplikované informatiky

Datum zadání diplomové práce: **20. října 2015**
Termín odevzdání diplomové práce: **16. května 2016**


prof. Ing. Václav Kopecký, CSc.
děkan




prof. Dr. Ing. Jiří Maryška, CSc.
vedoucí ústavu

V Liberci dne 20. října 2015

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Abstrakt

Tato diplomová práce je zaměřena na problematiku zpracování obrazu pro detekci a měření průměrů vláken ze snímků pořízených elektronovým mikroskopem. Popisuje problémy při detekci těchto vláken a uvádí postupy pro jejich eliminaci a volbu vhodného algoritmu pro hledání hran. Dále popisuje realizaci aplikace v jazyce C#, která snímek zpracuje a umožní měřit jeho průměr na základě zadaného měřítka.

Klíčová slova: Zpracování obrazu, detekce hran

Abstract

The thesis describes procedures of the automatic image analysis of the pictures from electron microscopy of the fiber structure. The work focuses on automatic detection and measurement of fiber diameters and algorithms used for optimization of the results. Thesis also describes development of Windows forms application written in C# for automatic measuring the diameter of fibers selected from these pictures.

Keywords: Image processing, edge detection

Poděkování

Na tomto místě bych rád poděkoval Ing. Petru Kretschmerovi za vedení práce, Ing. Michalu Komárkovi, Ph.D. za konzultace a Ing. Zdeňku Věchetovi za cenné rady a věcné připomínky při tvorbě práce. Dále také rodině za podporu při studiu.

Obsah

Abstrakt.....	5
Abstract.....	5
Poděkování.....	6
Seznam ilustrací.....	9
1 Úvod.....	11
2 Analýza problému a existující řešení.....	12
2.1 Tescan Atlas.....	12
2.2 ImageJ.....	13
2.3 NI Vision Assistant.....	14
3 Digitální obraz.....	15
3.1 2D konvoluce v digitálním obraze.....	16
3.2 Prahování.....	16
3.3 Adaptivní prahování.....	17
3.4 Detekce hran.....	18
3.5 Odstranění šumu.....	21
3.6 Barvení oblastí.....	23
4 Realizace v jazyce C#.....	25
4.1 Vývojové prostředí.....	25
4.2 Práce s bitmapou.....	25
4.3 Seznam posledních otevřených snímků.....	26
4.4 Předzpracování snímku.....	27
4.5 Detekce hran hranovými detektory.....	29
4.6 Detekce hran morfologickou erozí.....	31
4.7 Adaptivní prahování.....	32
4.8 Odfiltrování malých objektů.....	33
4.9 Označování a hledání vláken.....	33
4.10 Zobrazování výsledků měření a jejich export.....	37
4.11 Zobrazení rozložení průměrů vláken.....	37
4.12 Ukládání a načítání projektu.....	38
4.13 Snížení počtu naměřených hodnot.....	39

5 Vzhled aplikace a její ovládání.....	41
5.1 Načtení snímku.....	41
5.2 Úprava snímku před měřením.....	42
5.3 Měření vláken.....	44
5.4 Výsledky měření.....	46
5.5 Zadání měřítka.....	47
5.6 Zobrazení četností formou grafu.....	48
6 Závěr.....	49
Seznam použité literatury.....	50
Obsah přiloženého CD.....	51

Seznam ilustrací

Obrázek 2.1: Ukázka průběhu pořizování snímku aplikací Tescan Atlas.....	13
Obrázek 3.1: Ukázka konvoluce. Na žlutě označený pixel je použito jádro h.....	16
Obrázek 3.2: Původní snímek a oprahovaný snímek s prahem 100.....	17
Obrázek 3.3: Původní snímek, globální prahování s prahem 90 a adaptivní prahování s velikostí okolí $141 * 141$ a konstantou 15.....	18
Obrázek 3.4: Použití adaptivního prahování k detekci hran.....	18
Obrázek 3.5: Jednotlivé typy hran.....	19
Obrázek 3.6: Morfologická operace eroze.....	21
Obrázek 3.7: Ukázka postupného rotování masky.....	23
Obrázek 3.8: Kolize barev.....	23
Obrázek 3.9: Barvení oblastí.....	24
Obrázek 4.1: Tabulka časových náročností algoritmu při použití převodu bitmapy na pole bytů a přímé práce s bitmapou.....	26
Obrázek 4.2: Část diagramu tříd znázorňující dědičnost.....	27
Obrázek 4.3: Snímek po filtraci šumu průměrováním.....	28
Obrázek 4.4: Ukázka histogramů.....	29
Obrázek 4.5: Vliv filtrace šumu na prahování snímku.....	29
Obrázek 4.6: Detekované hrany s různou velikostí filtru.....	30
Obrázek 4.7: Detekce hran operátory Kirch a Prewitt.....	30
Obrázek 4.8: Hrany nalezené konvolucí s jádrem Prewittové před a po globálním prahování.....	31
Obrázek 4.9: Detekce hran erozí u oddělených a překrývajících se objektů.....	32
Obrázek 4.10: Hrany nalezené adaptivním prahováním před a po odstranění malých objektů.....	33
Obrázek 4.11: Algoritmus posunu a zvětšování kružnice reprezentující průměr vlákna.....	36
Obrázek 4.12: Naměřené hodnoty na křížení vláken.....	39
Obrázek 5.1: Rozbalovací menu.....	41
Obrázek 5.2: Ovládací panel detekce vláken.....	42
Obrázek 5.3: Jednotlivé vrstvy snímku.....	43
Obrázek 5.4: Hrany nalezené adaptivním prahováním a hranovým detektorem.....	44

Obrázek 5.5: Panel pro výběr způsobu měření a redukce hodnot.....	44
Obrázek 5.6: Ukázka měření tahem myši.....	45
Obrázek 5.7: Přibližovací okénko.....	46
Obrázek 5.8: Tabulka naměřených hodnot.....	47
Obrázek 5.9: Formulář pro ruční editaci měřítka.....	47
Obrázek 5.10: Ukázka grafu rozložení vláken.....	48

Seznam zkratek

- RGB – Červená-zelená-modrá, barevný model používaný v monitorech a projektorech
- CMY – azurová-purpurová-žlutá, barevný model používaný v tiskárnách
- YUV – barevný model používaný v televizním vysílání
- HSV – odstín, sytost, hodnota jasu, barevný model příbuzný RGB
- CSV – Comma-separated values, hodnoty oddělené čárkami
- XML – Extensible markup language, značkovací jazyk pro popis dat
- JSON – Javascript object notation, způsob zápisu dat pro přenos mezi platformami

1 Úvod

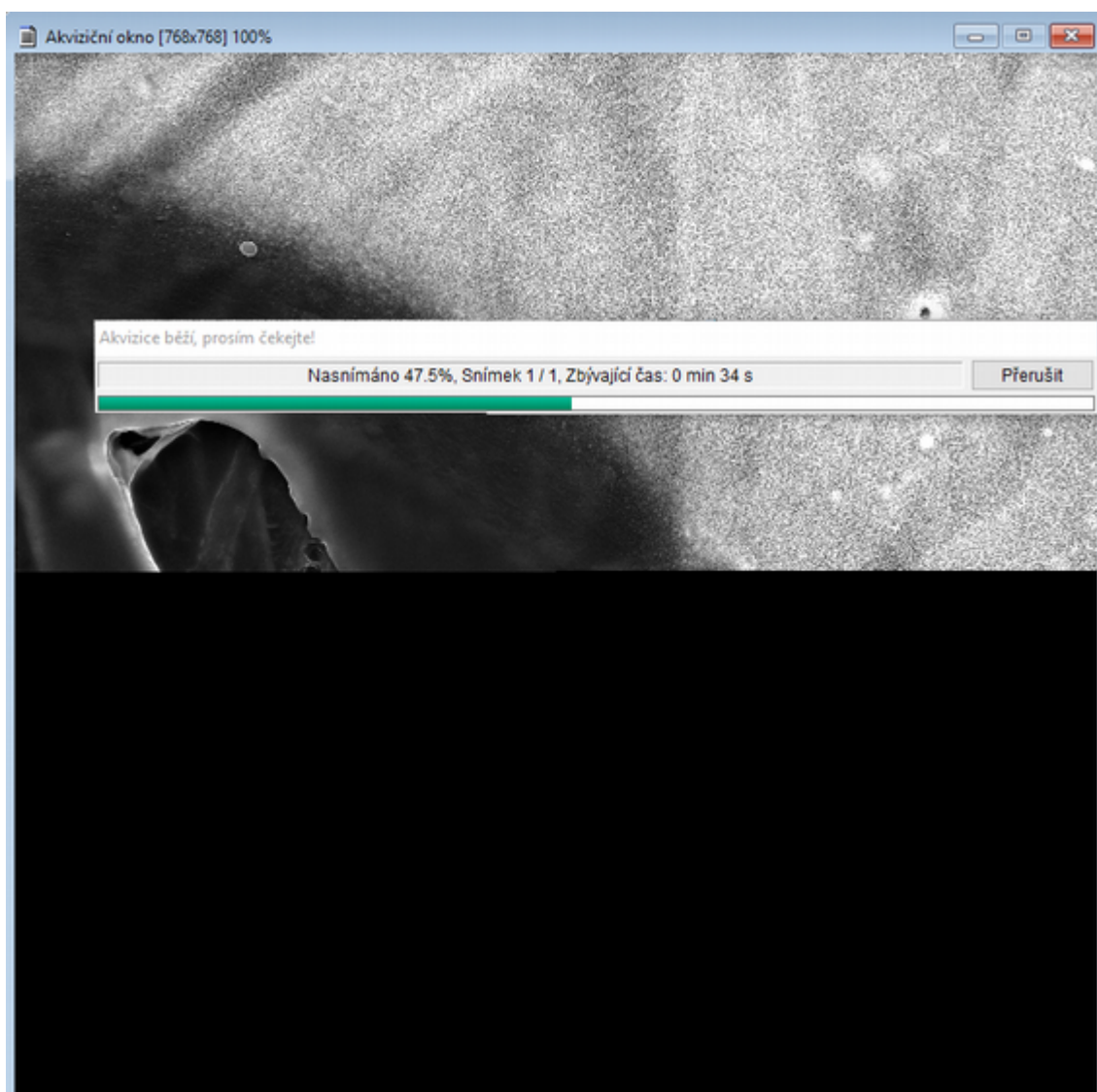
Výzkum nanomateriálů je aktuálně jedním z předních oborů v mnoha světových institutech i v laboratořích Technické univerzity v Liberci. Specifické vlastnosti těchto materiálů podmiňují rovněž inovace v analytických metodách umožňujících jejich popis. Příkladem je možné uvést nutnost použití elektronové mikroskopie pro kvalitní zobrazení struktur a morfologie nanomateriálů. Vzhledem k rozměrům struktur velmi blízkým vlnovým délkám spektra viditelného světla není klasická optická mikroskopie schopna dosáhnout potřebného rozlišení. Výstupy z elektronové mikroskopie jsou obvykle zpracovávány formou obrazové analýzy. V případě nanovláknenných materiálů, na zpracování jejichž obrazové dokumentace je tato diplomová práce zaměřena, obvykle bývají měřeny průměry vláken a jejich distribuční křivka, velikost mezivláknenných prostorů (pórů) a stereologie (míra izotropie) uspořádání vláken. Výsledky těchto analýz je vždy nutné statisticky zpracovávat, neboť nanovláknenné materiály typicky vykazují nestejnoměrnosti a relativně širokou distribuci popsanych vlastností. Aktuálně je převážná většina obrazové analýzy a statistického vyhodnocení prováděna manuálně pomocí dostupných komerčních softwarů, jež budou popsány v detailu v dalším textu práce. Manuální zpracování obrazové analýzy a statistiky je přirozeně značně časově náročné a v praxi výrazně omezuje velikost statistických souborů jednotlivých měření, tj. počet uskutečněných měření. . Možnost automatizace měření průměrů vláken ze snímků vláknenných struktur z elektronové mikroskopie a následného statistického zpracování, již se zabývá předkládaná diplomová práce by významnou měrou přispěla ke komfortu i zpřesnění analýzy morfologie nanovláknenných materiálů. Motivací práce je vytvoření volně dostupného software které by výzkumníkům v laboratořích TUL umožnilo využití automatizovaného měření průměrů vláken .

2 Analýza problému a existující řešení

V současném stavu je pomocí dostupného softwaru časově náročné vytvořit dostatečný počet měření pro statistické zpracování. Z ústavu pro nanomateriály, pokročilé technologie a inovace vzešel požadavek na volně dostupný software umožňující automatizaci měření. Software dodávaný s mikroskopem nevyužívá metod obrazové analýzy a při měření je nutné spolehnout se na přesnost výzkumného pracovníka. Situace je obdobná i u volně dostupných softwarů. Cílem této práce je nastudovat možnosti obrazové analýzy a zpracování obrazu a za pomoci získaných znalostí vytvořit aplikaci, která umožní komfortnější a rychlejší měření zvolených oblastí. Aplikace zpracuje snímek a s minimální zásahem uživatele detekuje vlákna kombinací detekce hran a prahování. Pro změření vlákna nebude nutné ručně označovat hranice vlákna, jako v případě aplikace dodávané k mikroskopu. Uživatel pouze označí oblast ve které bude vlákno hledáno a měřeno. Automatická detekce vláken v celém snímku umožní získat statistickou analýzu materiálu.

2.1 Tescan Atlas

Aplikace Tescan Atlas je dodávaná s elektronovým mikroskopem a používá se ke samotnému snímání vzorků a ovládání mikroskopu. Dále také umožňuje manuální měření v pořízených snímcích. K nabízeným možnostem patří například měření vzdálenosti mezi dvěma rovnoběžkami, měření úhlů, měření průměrů a obsahů zvolených oblastí. Zásadním nedostatkem znemožňujícím komfortní práci je to, že aplikace nevyužívá technologií zpracování obrazových dat k určování hran, oblastí apod. Uživatel musí ručně zadávat jednotlivé body měření a není tak možno práci automatizovat. Výhodou aplikace je zpracovávání metadat pořízeného snímku, což umožňuje například přesný výpočet vzdálenosti na základě měřítka. Tyto metadata jsou bohužel v uzavřeném formátu, který nelze využít v aplikacích třetích stran.



Obrázek 2.1: Ukázka průběhu pořizování snímku aplikací Tescan Atlas

Dalším nedostatkem je, že aplikace je vázána na počítač připojený k mikroskopu a neumožňuje tak práci více uživatelům současně.

2.2 ImageJ

ImageJ je robustní otevřený software pro zpracování obrazu. ImageJ je otevřený software napsaný v Javě, tudíž je i multiplatformní. V základní verzi implementuje nepřehledné množství možností zpracování a analyzování obrazu. Další funkce lze přidat pomocí zásuvných modulů [8].

Pomocí této aplikace můžeme realizovat postupy předzpracování obrazu popsané dále v této diplomové práci. Možnosti měření jsou obdobné jako u aplikace Tescan

Atlas. Opět se zde ale nevyužívá analýzy snímku pro měření v obraze. Tato vlastnost znemožňuje automatizaci měření, byť aplikace podporuje vytváření vlastních maker. Aplikace ImageJ je navíc pro uživatele, na které je zaměřena aplikace realizována v této diplomové práci, složitá a nepřehledná.

2.3 NI Vision Assistant

Vision Assistant je software od firmy National Instruments a je součástí vývojového balíku LabVIEW určeného pro počítačové vidění [9]. Vision Assistant umožňuje snímání obrazu a jeho další zpracování. Implementuje velké množství funkcí pro úpravu obrazu a měření. Aplikace umožňuje hledání hran po úsečce a hledání rovných, či kruhových hran ve zvolené oblasti. Vzdálenosti mezi nalezenými body hran lze měřit pomocí několika nástrojů. Algoritmus měření v obraze se sestavuje posloupností jednotlivých funkcí. Při vhodném nastavení se algoritmus přizpůsobí změnám mezi různými snímky a je tak vhodný například pro kontrolu sériově vyráběných výrobků jako jsou například pístní kroužky a podobně.

V případě snímků vlákných struktur, které obsahují velké množství objektů, není možné v celém snímku automaticky určit všechny hranové body, tudíž odpadá možnost automatizovaného měření celého snímku. V případě ručního měření jednotlivého vlákna je možnost změřit průměr vepsáním kružnice. Kružnice ale musí být definována minimálně třemi body, což je omezení, které neumožňuje námi požadované měření.

Prostředí aplikace není příliš intuitivní a vyžaduje znalost technik zpracování obrazu. Vhodně sestavený algoritmus lze vyexportovat pro další použití v aplikacích LabVIEW, případně do jazyka C, nebo C#.

3 Digitální obraz

Zpracování obrazových dat má široké uplatnění, ať už v oblasti rozpoznávání obličejů, detekci pohybu na kamerách, či jako automatická kontrola kvality výrobků v průmyslu, ale i např. potravinářství. Tato kapitola se bude zabývat definicí digitálního obrazu, metodami pro segmentaci obrazu, redukci šumu a na závěr porovná některé existující nástroje pro zpracování obrazu.

Digitální obraz je reprezentace dvourozměrného obrazu. Obrázek je v počítači reprezentován buď jako rastrový, tj. popisem pozice bodu v mřížce a jeho barvy popsané vhodným barevným modelem (RGB, CMYK, HSV, YUV, ...), nebo jako vektorový pomocí základních tvarů, jako jsou body, přímky, křivky a mnohoúhelníky. Vektorové obrázky je možné, na rozdíl od rastrových, beze ztráty kvality libovolně zvětšovat a zmenšovat. U obrázků s malým počtem objektů je vektorová grafika méně výpočetně náročná než rastrová, ale při překročení určitého počtu objektů se vektorová grafika stává výpočetně náročnější. Náročné je také pořízení vektorového obrázku a na fotografii se tato prezentace nehodí. Proto jsem v diplomové práci pracoval pouze s rastrovou reprezentací.

Nejčastěji používaným barevným modelem je aditivní způsob míchání barev RGB. V RGB je pro každý pixel popsána mohutnost červené, zelené a modré barvy. Intenzita jednotlivých složek může být popsána v rozsahu od 0 (minimum) do 1 (maximum), procentuálně, a nebo pro počítačové výpočty vhodně v rozsahu 0 až 255. V případě poslední možnosti je jedna složka popsána jedním bytem.

Velké množství metod zpracování obrazu pracuje s šedotónovým snímkem. V RGB mají u šedotónového snímku všechny kanály stejnou hodnotu. Tato výsledná jasová hodnota se pro šedotónový snímek obecně spočítá jako

$$Y(x,y)=\frac{1}{3}\cdot R(x,y)+\frac{1}{3}\cdot G(x,y)+\frac{1}{3}\cdot B(x,y).$$

Sofistikovanější způsob je rozložení vah jednotlivých barevných složek, podle toho, jak je lidské oko na dané barevné složky citlivé. Příkladem takového rozložení může být $Y(x,y)=0,3\cdot R(x,y)+0,59\cdot G(x,y)+0,11\cdot B(x,y)$. Lidské oko nejcitlivěji vnímá zelenou barvu, a proto má zelená složka největší váhu. [1]

3.1 2D konvoluce v digitálním obraze

Konvoluce je matematický operátor využívaný při úpravách všech možných druhů signálů. V případě digitálního obrazu se používá diskrétní 2D konvoluce definovaná vztahem:

$$g(x, y) = \sum_{k=-M/2}^{M/2} \sum_{l=-N/2}^{N/2} (h(x-k, y-l) \cdot f(k, l)),$$

kde $f(x, y)$ je obraz a $h(x, y)$ je konvoluční jádro [2].

Jádro můžeme chápat jako masku, která se položí na příslušné místo v obraze. Každý pixel překrytý tabulkou vynásobíme příslušným koeficientem z tabulky a všechny takto získané hodnoty sečteme a získáme tak novou hodnotu pixelu.

$f =$	202	170	219	210	179
	245	173	198	190	169
	161	214	205	219	247
	228	222	164	154	189
	248	226	179	249	192

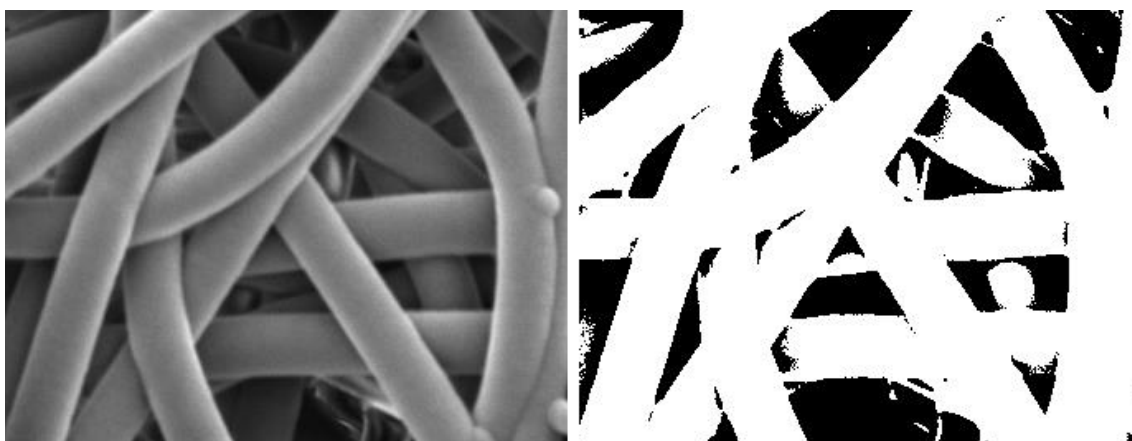
$h = \frac{1}{10} \cdot$				
1	1	1		
1	2	1		
1	1	1		

$g =$	202	170	219	210	179
	245	173	198	190	169
	161	214	194	219	247
	228	222	164	154	189
	248	226	179	249	192

Obrázek 3.1: Ukázka konvoluce. Na žlutě označený pixel je použito jádro h

3.2 Prahování

Prahování je nejjednodušší funkce segmentace obrazu, založená na vyhodnocování jasu pixelu. V nejjednodušší variantě je určen globální práh a výsledný obraz je binární. $f(c) = \begin{cases} A, & \text{pokud } c < \text{práh} \\ B, & \text{pokud } c \geq \text{práh} \end{cases}$ kde A a B zpravidla nabývají dvou hodnot 0 a 255 [3].



Obrázek 3.2: Původní snímek a oprahovaný snímek s prahem 100

Prahování lze rozšířit o více prahových hodnot, čímž vznikne obraz s omezeným počtem jasových úrovní.

Hodnotu prahu lze určit i automaticky, například pomocí analýzy obrazového histogramu¹. Pro použití této metody je nutné, aby se výrazně lišila minima a maxima histogramu.

3.3 Adaptivní prahování

V případě snímku bez výrazného vrcholu v histogramu (Obrázek 3.3) globální prahování selhává. Pro tento případ lze použít metodu adaptivního (proměnného) prahování.

Hodnota prahu je vypočítávána jako vážený průměr jasových hodnot okolí bodu dané velikosti. Pokud v obraze není výrazný rozdíl mezi pozadím a popředím, není hodnota váženého průměru vhodná pro použití jako prahové hodnoty. To lze vylepšit zavedením konstanty. Konstanta určuje, jak moc se musí zpracováváný pixel lišit od průměru okolí. Použitím konstanty jsou okolní pixely s podobným jasem nastaveny jako pozadí. [4]

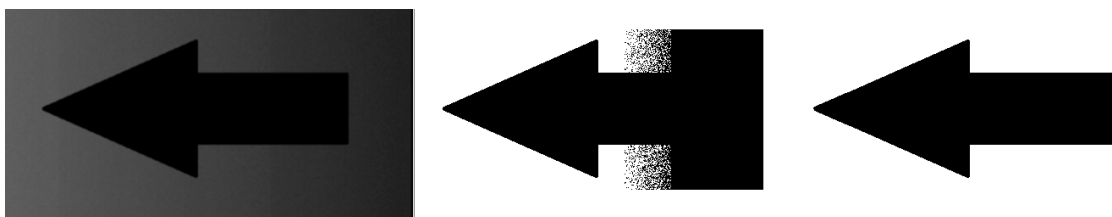
¹ Obrazový histogram je grafické znázornění četnosti jasových hodnot. Každý barevný kanál má svůj histogram. Celkový histogram je složený z histogramů jednotlivých kanálů. Výsledný jas se spočítá způsobem uvedeným v úvodu kapitoly.

V případě velikosti okolí 3 * 3 okolí bodu

$$K = \begin{pmatrix} 200 & 190 & 190 \\ 180 & 160 & 155 \\ 170 & 170 & 170 \end{pmatrix}$$

a konstantě $C=3$ bude hodnota prahu:

$$P = \sum_{x=0}^2 \sum_{y=0}^2 M_{x,y} * K_{x,y} / \sum_{x=0}^2 \sum_{y=0}^2 M_{x,y} - C \doteq 173.$$



Obrázek 3.3: Původní snímek, globální prahování s prahem 90 a adaptivní prahování s velikostí okolí 141 * 141 a konstantou 15

Při vhodném snímku a zvolení malého okolí (například 7 * 7) lze použít tuto metodu i pro detekci hran.



Obrázek 3.4: Použití adaptivního prahování k detekci hran

3.4 Detekce hran

Detekce hran je další metodou segmentace obrazu. Hrany jsou místa v obraze s velkou změnou hodnoty jasu. V obraze nemusí však být hrany pouze na hranici objektů. Jako hrany jsou také detekovány hranice světla a stínu. Ne vždy je tento přechod mezi světlem a stínem prudký (skoková hrana). V digitálním obraze se spíše nacházejí šikmé hrany, kde je změna jasu postupná. Obrázek 3.5 znázorňuje jednotlivé

typy hran, tj. skokovou, dvě skokové za sebou (čára), šikmou a dvě šikmé za sebou (střecha) [7].



Obrázek 3.5: Jednotlivé typy hran

Z hlediska digitálního obrazu jsou hrany detekovány i v místech různých odlesků a podobně, které lidské oko jako hranice objektů nevnímá. Tato vlastnost způsobuje problémy při zpracování detekovaných hran.

3.4.1. Detekce dle změny jasu

V místě hrany je velká hodnota derivace (gradient) jasové funkce. Velká skupina metod na detekci hran aproximuje tuto derivaci pomocí konvoluce s vhodným jádrem. Často se používá konvoluce ve dvou směrech, tedy konvoluce s dvěma jádry (jedno pro vertikální a jedno pro horizontální směr) [7]. Nejjednoduššími jádry mohou být:

$$\begin{pmatrix} -1 & 1 \end{pmatrix} a \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \text{ případně } \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} a \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}.$$

Výsledná hodnota jasu u dvoujádrových detektorů je pak určena podle vzorce:

$$Jas = \sqrt{Jas_{Hor}^2 + Jas_{Vert}^2}.$$

Aproximace první derivace nachází hrany v místech lokálních extrémů jasové funkce. Existuje řada navržených jader, nejčastěji o velikosti 3×3 . Některá jádra existují i ve variantách větší velikosti.

Příkladem mohou být jádra navržená Judith Prewittovou :

$$\text{Jádro pro vertikální směr} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \text{ jádro pro horizontální směr} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}.$$

Detekci můžeme zlepšit počítáním gradientu ve více směrech. Při použití osmi konvolučních jader jádra získáme otáčení o 45° .

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \dots$$

Výsledná hodnota gradientu je pak rovna největší odezvě na jednu z masek.

Velmi podobné jsou masky navržené Irwinem Sobelem a Gary Feldmanem:

$$\text{Jádro pro vertikální směr} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \text{jádro pro horizontální směr} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}.$$

Druhým typem jsou operátory aproximující druhou derivaci. Využívají toho, že nalezení průchodu nulou je snazší, než nalezení extrémů funkce. Druhá derivace je ale citlivější na šum [7]. Často se používá Laplaceův operátor, který má stejné vlastnosti ve všech směrech. Konvoluční jádro Laplaceova operátoru pro okolí 3×3 je :

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}.$$

Z důvodu větší citlivosti na šum se Laplaceův operátor často kombinuje s Gaussovým filtrem. Výsledkem je pak filtr nazývaný mexický klobouk:

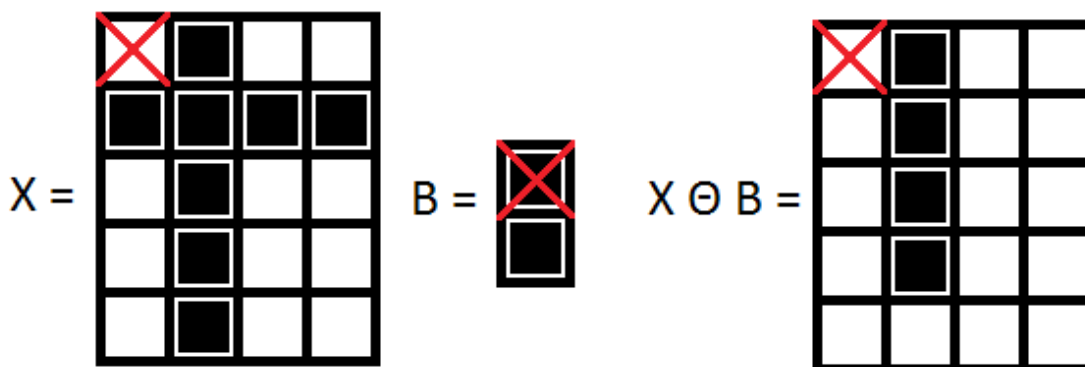
$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}.$$

3.4.2. Detekce erozí

Další metodou detekce hran je využití morfologické operace eroze na oprahovaný snímek. Morfologie je nauka o tvarech. Nástroje matematické morfologie jsou založeny na nelineárních operacích v obraze a slouží k získání informace o tvaru objektů a jejich popisu. Dělí se na operace pracující s binárním obrazem a pracující s šedotónovým obrazem. Morfologická transformace je relace mezi obrazem a strukturním elementem.

Eroze je definována jako $X \ominus B = \{p \in \varepsilon^2 : p+b \in X \text{ pro každé } b \in B\}$, tj. pro každý bod obrazu p se ověřuje, zda výsledek $p + b$ leží v X [7].

Pro detekci hran se od původního obrazu odečte erodovaný obraz



Obrázek 3.6: Morfologická operace eroze

3.5 Odstranění šumu

Šum jsou obecně náhodné a nepředvídatelné signály překrývající požadované informace. V oblasti digitálního obrazu vzniká šum často už při pořizování snímku vlivem špatných světelných podmínek, případně poškozením čoček nebo tepelnými změnami u snímačů, či dalších součástek. Šum může také vzniknout při přenosu obrazu digitálním kanálem. [6] Šum má nepříznivý vliv na zpracování obrazu, ať už na detekci hran, prahování apod. Proto je výhodné použít, v závislosti na typu šumu, některou z metod filtrace šumu.

3.5.1. Průměrování

Průměrování patří mezi nejjednodušší metody filtrace šumu. Používá se konvoluční jádro (pro okolí 3×3):

$$h = \frac{1}{9} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

případně s váhováním bodů:

$$h = \frac{1}{10} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, h = \frac{1}{16} \cdot \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

Další modifikací průměrování je průměrování podle Gaussova rozložení. Podle vzorce:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right),$$

kde σ je směrodatná odchylka a x, y jsou souřadnice v obraze, se spočítá váha daných bodů v lokálním okolí. Při aplikaci masky s Gaussovským rozložením se předpokládá, že na snímek působil šum s normálním rozložením.

Použitím průměrovacího filtru dochází k rozostření obrazu. Čím je větší okolí bodu pro průměrování, tím je rozostření větší. Proto dochází i porušování a rozmazávání hran.

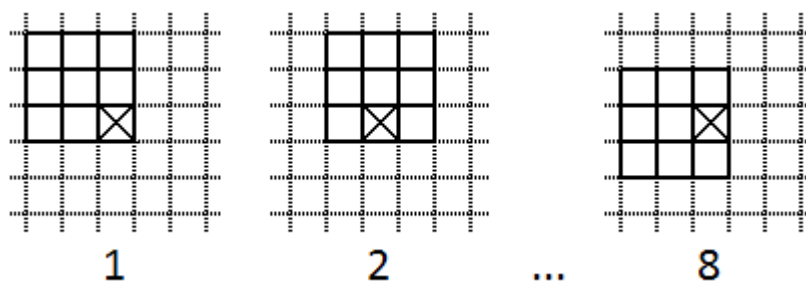
3.5.2. Medián

Mediánový filtr sestaví posloupnost uspořádaných hodnot z daného okolí pixelu. Prostřední hodnota této posloupnosti se stane novou hodnotou zpracovávaného pixelu. Filtr nerozmazává hrany, ale zároveň porušuje tenké čáry a ostré rohy. Mediánový filtr je vhodný pro filtraci šumu způsobeného ztrátou obrazové informace na náhodných místech obrazu.

3.5.3. Metoda rotující masky

Metoda rotující masky vyhledává v okolí bodu 5×5 homogenní části masky velikosti 3×3 . Maska je postupně posouvána po okolí bodu (celkem 8 poloh). V každé poloze je spočítána střední hodnota a rozptyl jasů, definovaný jako průměr součtu čtverců odchylek jasu od střední hodnoty. Jako nová hodnota jasu pixelu se použije střední hodnota masky s nejmenším rozptylem. Tato metoda mírně ostří a nerozmazává hrany. Střední hodnota se spočítá jako aritmetický průměr hodnot masky. Rozptyl pak

jako $\frac{1}{n} \sum_{i=1}^n (x_i - E(X))^2$, kde $E(X)$ je střední hodnota a x_i jsou hodnoty z masky.



Obrázek 3.7: Ukázka postupného rotování masky

3.6 Barvení oblastí

Pro odstranění malých objektů v binarizovaném obraze po předzpracování snímku je použita metoda identifikace oblastí. Při prvním průchodu snímku se prochází okolí bodu (i, j) pomocí masky:

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	

Pokud má některý okolní bod přiřazené číslo oblasti, přiřadí se toto číslo i aktuálnímu bodu. Pokud ne, je bodu přiřazeno následující přirozené číslo. V některých případech může nastat kolize. Zpracovávanému pixelu na obrázku 3.8 bude přiřazeno číslo 2. Zároveň ale podle bodu $(i, j-1)$ sousedí i s oblastí označenou číslem 1. Do tabulky ekvivalencí se tato kolize uloží a bude zpracována při druhém průchodu snímku [3].

1		2
1		2
1	1	?

Obrázek 3.8:

Kolize barev

Při druhém průchodu se podle tabulky ekvivalencí sloučí sousední oblasti. Z výstupního snímku lze určit počet objektů ve snímku a jejich velikost.

0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	1	0	0	0	0
0	0	1	1	0	0	1	1	0
0	0	1	1	0	1	0	0	0
0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

Binární obraz

0	0	0	0	0	0	0	0	0
0	2	2	0	3	0	0	0	0
0	2	2	2	3	0	0	0	0
0	0	2	2	0	0	4	4	0
0	0	2	2	0	4	0	0	0
0	0	0	2	0	4	4	0	0
0	0	0	0	0	0	0	0	0

První průchod

0	0	0	0	0	0	0	0	0
0	2	2	0	2	0	0	0	0
0	2	2	2	2	0	0	0	0
0	0	2	2	0	0	3	3	0
0	0	2	2	0	3	0	0	0
0	0	0	2	0	3	3	0	0
0	0	0	0	0	0	0	0	0

Druhý průchod

Obrázek 3.9: Barvení oblastí

4 Realizace v jazyce C#

V této kapitole se budeme zabývat výběrem metod zpracování obrazu vhodných pro realizaci aplikace pro měření průměrů vláken. Dále pak vnitřní realizací samotného hledání a měření vláken. Volba běhového prostředí a vzhledu aplikace byla přizpůsobena cílové skupině uživatelů.

Po načtení snímku provede aplikace několik kroků předzpracování před měřením. Dojde k redukci šumu, detekci hran a prahování. Poté uživatel aplikace zkontroluje výsledek předzpracování a doladí některá nastavení. Poté uživatel může spustit automatické vyhledání vláken a jejich měření, nebo ručně určit oblasti měření. Následně může odstranit nevhodná data a naměřené hodnoty uložit ve formě obrázku, grafu, či datového listu ve formátu CSV.

4.1 Vývojové prostředí

Aplikace je založena na .NET frameworku verze 4. .NET framework umožňuje vyvíjet krom klasických aplikací pro Windows také webové aplikace a služby, aplikace pro mobilní zařízení a mnohé další. Framework obsahuje velké množství funkcí a knihoven a i samotné běhové prostředí [10]. K vývoji je použit programovací jazyk C# a knihoven tříd pro grafické rozhraní Windows Forms, kde ovládací prvky vypadají stejně jako ve standardních Windows aplikacích.

4.2 Práce s bitmapou

Aplikace je založena na práci s obrázky a jejich zpracováním. Rastrový obrázek a jeho vlastnosti zapouzdřuje třída `Bitmap`. U všech metod přístupujících ke snímku bychom si vystačili s metodami `GetPixel` a `SetPixel` této třídy. Zpracování snímku je v tomto případě pomalé. Proto aplikace využívá metody třídy `BitmapData` pro převedení bitmapy na jednorozměrné pole bytů. Testováním bylo zjištěno, že využití pole je při stejné algoritmické složitosti přibližně 40–60krát rychlejší než přímá práce s bitmapou. Velikost snímku rychlost zpracování výrazně ovlivňuje. Testování bylo prováděno na algoritmu adaptivního prahování.

1024 × 768 px				256 × 192 px		
Roz. jádra	Pole (s)	Bitmapa (s)	Poměr časů	Pole (s)	Bitmapa (s)	Poměr časů
7 × 7	0,71	31,32	44,43	0,04	2,19	54,75
11 × 11	1,49	66,91	44,91	0,08	4,95	61,88
15 × 15	2,65	131,34	49,56	0,14	8,34	59,57
21 × 21	5,13	236,33	46,07	0,31	17,37	56,03

Obrázek 4.1: Tabulka časových náročností algoritmu při použití převodu bitmapy na pole bytů a přímé práce s bitmapou

Při vytváření instance třídy `BitmapData` je nutné určit formát uložení pixelů. Barevný snímek používá formát `Format24bppRgb`, kde je pixel popsán 24 bity (8 bitů pro každý z kanálů R, G a B). Často se používá i formát `Format32bppArgb`, kde dalších 8 bitů zabírá alfa kanál určující průhlednost snímku. Aplikace pracuje pouze se šedotónovými snímky a tak stačí použít formát pixelů `Format8bppIndexed`, kde je pixel reprezentován 8 bity, čili jeho jasovou hodnotou. Velikost pole odpovídá počtu pixelů a pozice jasové hodnoty pixelu $p(x, y)$ je dána rovnicí:

$$p(x, y) = x + \text{šířka} \cdot y.$$

U některých operací jako je orientační zobrazení detekovaných hran výraznou barvou na původním snímku se kvůli průhlednosti využívá formát `Format32bppArgb`. Pozice hodnoty kanálu R pixelu $p(x, y)$ je dána rovnicí:

$$p(x, y) = x \cdot 4 + \text{šířka} \cdot y$$

Pozice zbylých kanálů je vždy o jedničku vyšší.

4.3 Seznam posledních otevřených snímků

Aplikace uchovává seznam posledních deseti otevřených snímků. Seznam je řazen sestupně podle času posledního otevření a je datového typu `List<KeyValuePair<String, DateTime>>`. Po otevření souboru je zjištěno, zda se soubor v seznamu nachází. Pokud ne, je přidán a při překročení počtu deseti souborů je nejstarší soubor odstraněn ze seznamu. Pokud soubor v seznamu je, aktualizuje se jeho datum posledního otevření.

4.4 Předzpracování snímku

Většina operací používaných pro předzpracování snímku před samotným měřením má několik společných vlastností. Pracují se snímkem ve formě pole bytů a využívají dvourozměrnou diskrétní konvoluci. Podle operace ale pracují s konvolucí vypočítanými hodnotami jinak. Tato situace je vhodná pro využití dědičnosti a abstraktních metod.

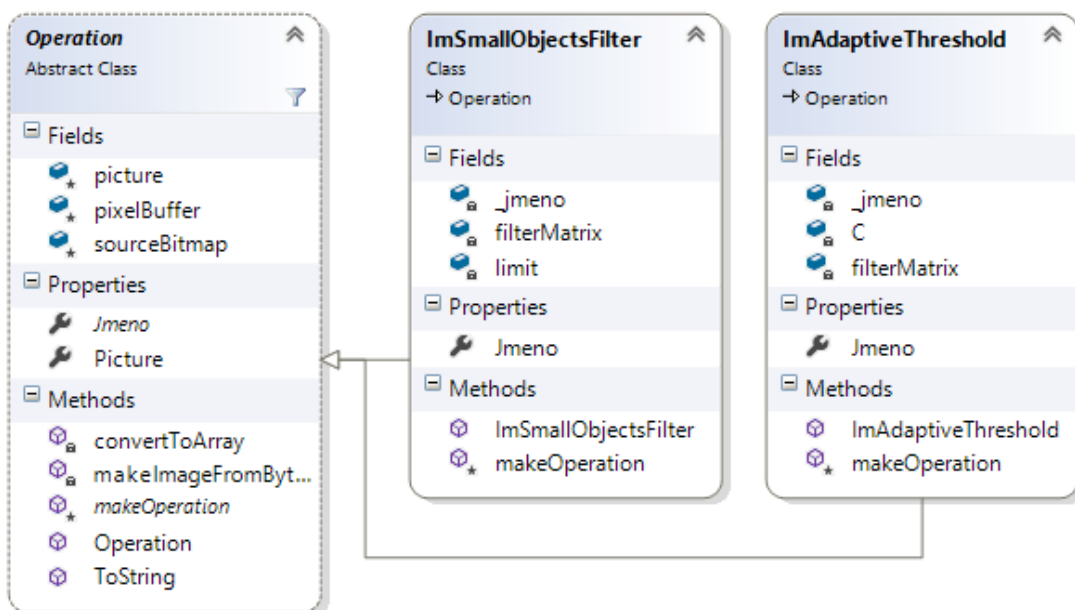
4.4.1. Abstraktní třída Operation

Rodičovská třída zastřešující jednotlivé metody zpracování obrazu je v aplikaci nazvána `Operation`. Jedná se o abstraktní třídu, která mj. obsahuje abstraktní metodu `makeOperation`, kterou musí všichni potomci této třídy definovat. V této metodě potomek zpracuje snímek dle své funkcionality.

Konstruktor třídy přijímá jako parametr zdrojovou bitmapu a konvoluční jádro ve formě dvourozměrného pole. V konstruktoru je převedena bitmapa na pole bytů pomocí nástrojů popsaných v předchozí podkapitole. Třída také obstarává sestavení výsledné bitmapy z pole bytů po provedení všech operací.

Třída dále obsahuje abstraktní vlastnost `name`, která definuje název operace.

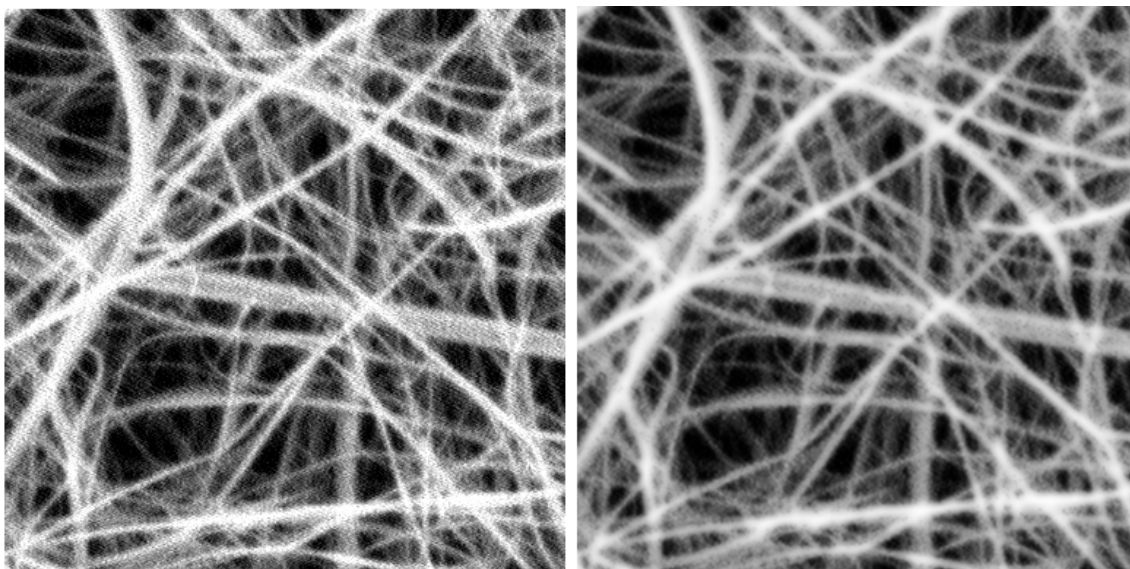
Konvencí zavedenou v této aplikaci je, že všechny třídy, kterou tuto třídu dědí mají předponu `Im`.



Obrázek 4.2: Část diagramu tříd znázorňující dědičnost

4.4.2. Průměrování snímku

Některé snímky jsou velmi zašuměné a vlákno tak není celistvé. Aplikací průměrovacího filtru dojde k mírnému rozmazání, ale redukuje se šum, což umožní přesnější detekci vláken.

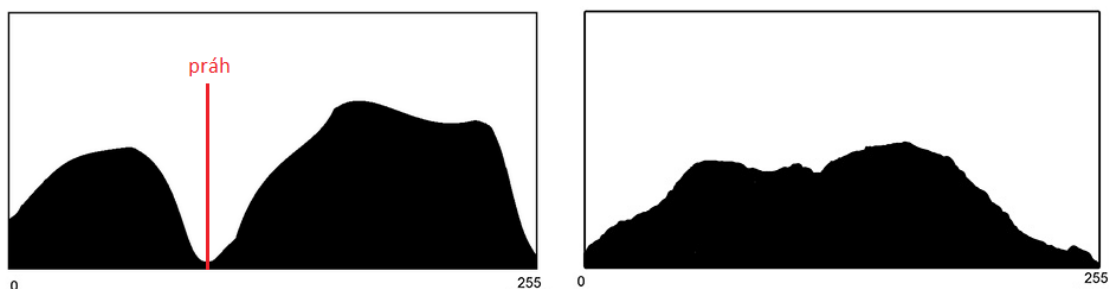


Obrázek 4.3: Snímek po filtraci šumu průměrováním

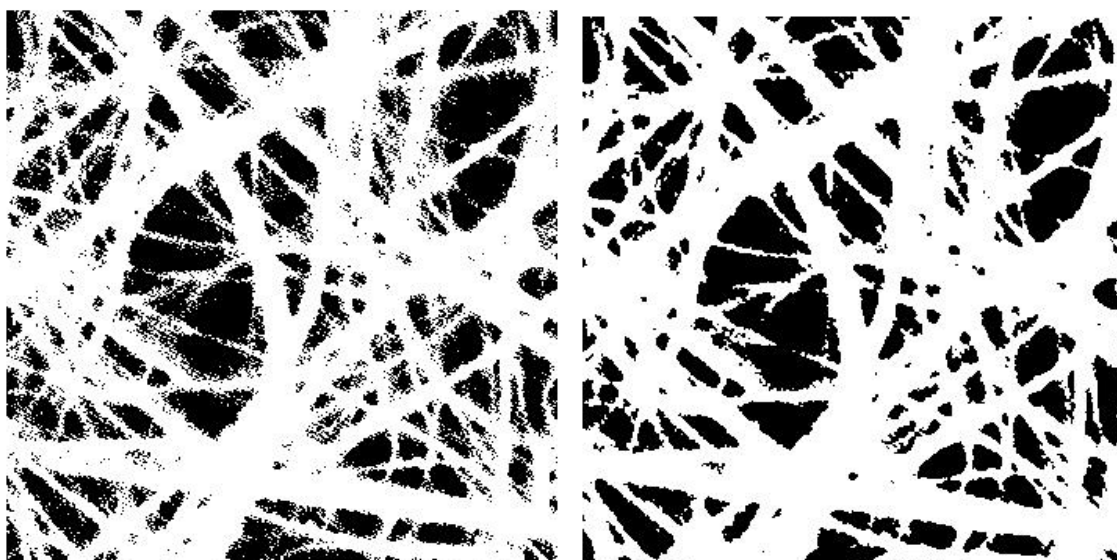
Průměrování se realizuje pomocí metod třídy `ImAverageFilter`, která redukuje šum metodou váženého průměrování popsané v kapitole 3.5 Odstranění šumu. Na vstupní snímek aplikuje konvoluční jádro předané konstruktoru třídy.

4.4.3. Prahování

Pro později popsané měření průměrů je třeba získat binární snímek vláken pomocí prahování. K prahování se používá třída `ImThreshold`. Automatické určení prahu z histogramu zdrojových snímků nerealizovatelné. Histogram vhodný pro automatické určení můžeme vidět na obrázku 4.4 vlevo. Histogramy zdrojových snímků (Obrázek 4.4 vpravo) nemají výrazná minima a maxima a nejsou tak vhodné pro automatické určení.



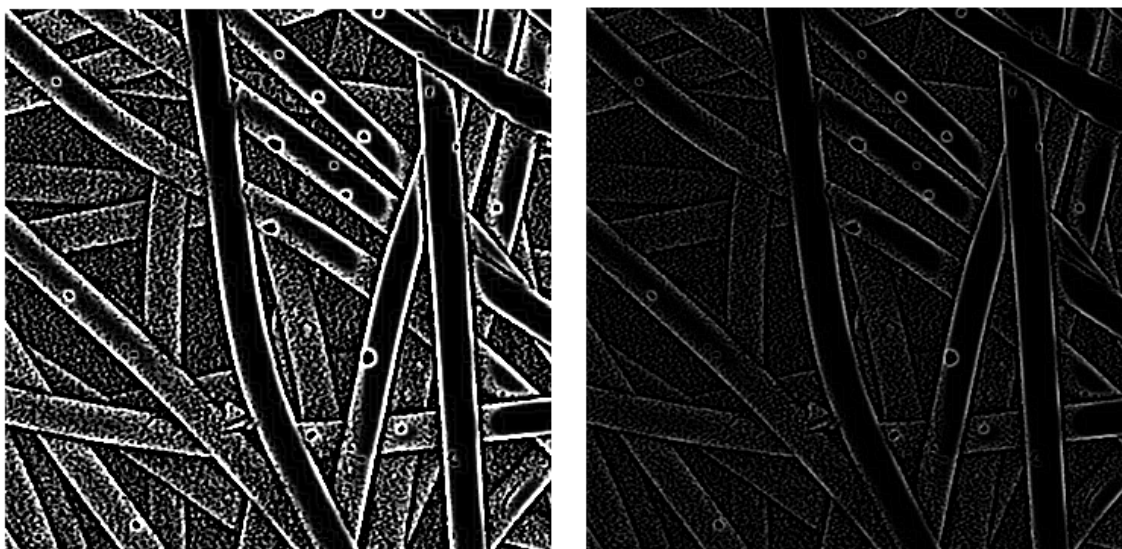
Obrázek 4.4: Ukázka histogramů



Obrázek 4.5: Vliv filtrace šumu na prahování snímku

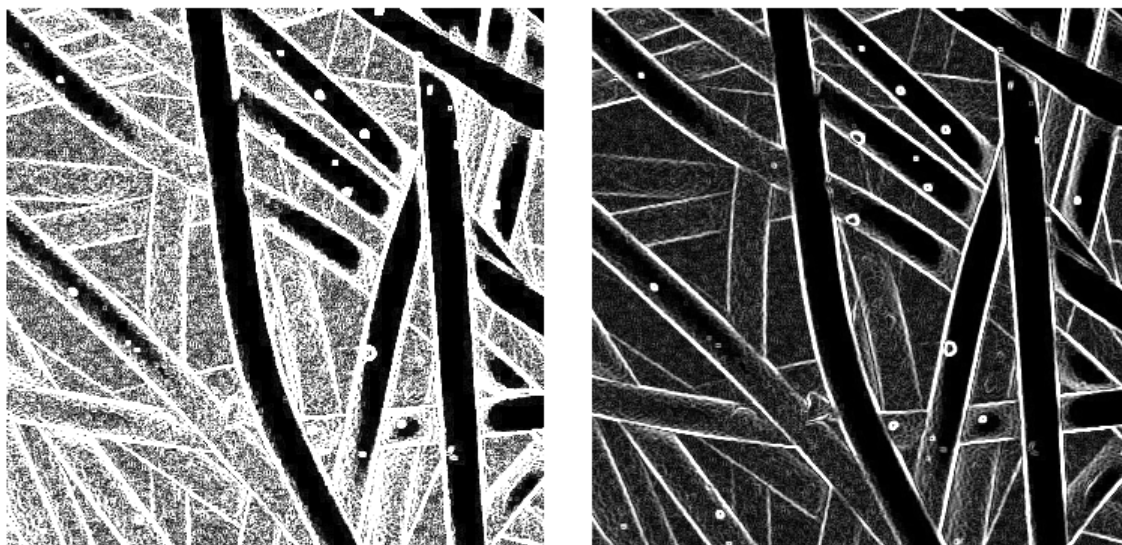
4.5 Detekce hran hranovými detektory

Třída `ImEdgeDetector` detekuje hrany vláken v zpracovaném snímku metodou konvoluce s vhodným jádrem. Třída může hledat hrany pomocí jednoho konvolučního jádra (například mexický klobouk), nebo dvou jader. V případě dvou jader je jedno pro vertikální a druhé pro horizontální směr. Nalezené hrany nejsou zcela ostré. Pro použití detekce vláken je snímek s hranami nutné oprahovat.



Obrázek 4.6: Detekované hrany s různou velikostí filtru

Obrázek 4.6 vlevo: Použito jádro velikosti 5×5 typu Laplacian of Gaussian, kde je velikost Gaussovského filtru 3×3 . Vpravo je jádro 3×3 se stejnou velikostí filtru. V prvním případě jsou díky použití většího okolí hrany výraznější, ale snímek obsahuje velké množství šumu s podobnou intenzitou jasu jako hrany a je tak velmi těžko filtrovatelný. V druhém případě není šum tak výrazný, ale opět má velmi podobnou intenzitu jasu. Pokud použijeme Laplacian 3×3 filtrovaný filtrem 5×5 je šum výrazně potlačen, ale na úkor výraznosti hran.



Obrázek 4.7: Detekce hran operátory Kirch a Prewitt

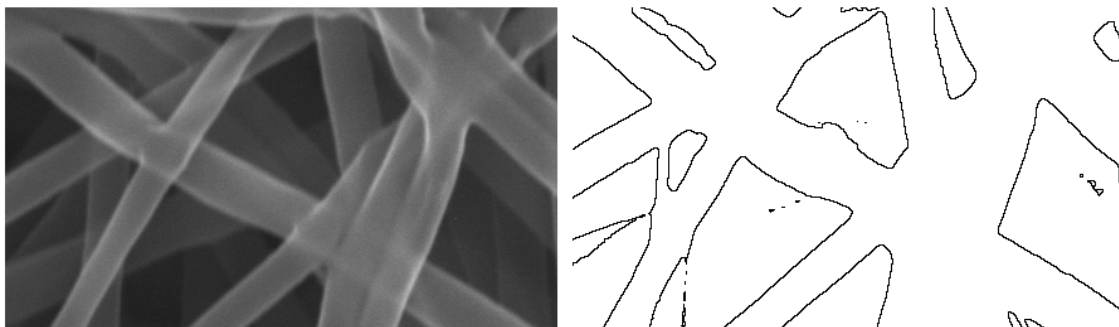
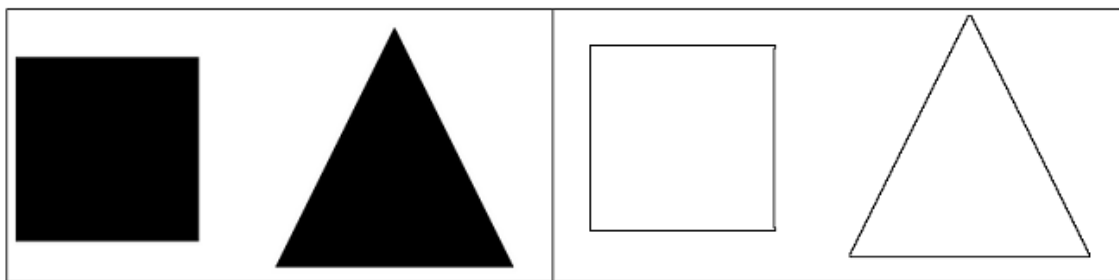
Na obrázku 4.7 vlevo je použito Kirschovo jádro. Toto jádro se ukázalo jako nevhodné pro zpracováváné snímky. Vpravo je použito jádro Prewittové, které sice obsahuje šum, ale s výrazně rozdílným jasem. Globálním prahováním dosáhneme výsledku použitelného pro měření průměrů vláken.



Obrázek 4.8: Hrany nalezené konvolucí s jádry Prewittové před a po globálním prahování

4.6 Detekce hran morfologickou erozí

Další možností hledání hran je morfologická operace eroze popsáná v kapitole 3.4 Detekce hran. Odečtením erodovaného snímku od zdrojového získáme hrany. Pro tyto operace jsou připraveny třídy `ImErode` a `ImSubtract`. Při velikosti strukturního elementu 3×3 mají všechny hrany šířku 1 pixel a jelikož eroze pracuje s binárními obrazy jsou všechny hrany stejně výrazné, což je výhodné pro detekci vláken, neboť jsou hranice objektů velmi přesně definovány. Prahováním binarizovaný objekt ztrácí informaci o hranicích mezi překrývajícími se objekty a je tak vhodná pouze pro snímky s oddělenými objekty.



Obrázek 4.9: Detekce hran erozí u oddělených a překrývajících se objektů

Metoda detekce hran erozí není kvůli nevýhodě spojené se ztrátou informací, obecně příliš použitelná pro typ snímků zpracovávané v realizované aplikaci. Pro některé snímky, zejména pro ruční doměřování některých oblastí, však mít uplatnění může.

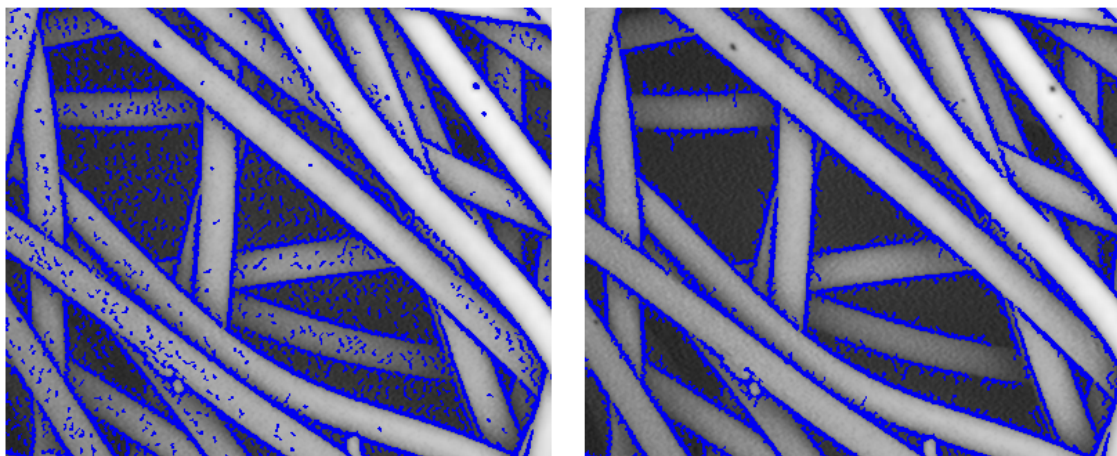
4.7 Adaptivní prahování

Použití algoritmu adaptivního prahování popsaného v kapitole 3.3 lze při vhodné zvoleném malém okolí použít pro detekci hran. Třídou pro adaptivní prahování je třída `ImAdaptiveThreshold`. Parametry třídy jsou zdrojová bitmapa, matice definující velikost okolí bodu a hodnota konstanty.

Adaptivní prahování se v aplikaci využívá jako jedna ze dvou metod detekce hran. Testováním byla nalezena vhodná velikost okolí pro detekci hran o velikosti okolí 7×7 . Po použití adaptivního prahování jsou na hranách patrné „otřepy“. Tyto chyby se projevují pouze na vnější straně vláken a na měření průměrů vláken tak nemají téměř žádný vliv. Projeví se pouze u křížujících se vláken, ale hodnoty naměřené v místech těchto křížení nebývají platné a jsou z výsledného snímku uživatelem odstraněny.

Z důvodu většího rozsahu jasových hodnot na vláknech může docházet k znečištění

snímku hran malými objekty zrnovitého charakteru. Tyto objekty jsou ale odstraněny filtrem malých objektů, popsáným v následující podkapitole.



Obrázek 4.10: Hraný nalezené adaptivním prahováním před a po odstranění malých objektů

4.8 Odfiltrování malých objektů

U obou metod detekce hran se setkáváme s nechtěnými malými objekty ve snímcích hran. Pro jejich odstranění byla navržena třída `ImSmallObjectsFilter` využívající algoritmu barvení oblastí popsaném v kapitole 3.6 Barvení oblastí. Tato třída jako jediná přepisuje metodu sestavení výstupní bitmapy, jelikož přijímá jako parametr minimální velikost oblasti určenou posuvníkem v aplikaci. Tento parametr udává, kolik pixelů musí na snímku spojitá oblast zabírat. Oblasti menší než daný limit jsou z výstupního snímku hran odstraněny. Protože jsou výsledné hrany zobrazeny na původním snímku, musí být výstupní snímek hran v dvaatřicetibitovém formátu. Hraný jsou modré a zbytek snímku má nastavenou plnou průhlednost pomocí alfa kanálu. Zobrazení filtrovaných a nefiltrovaných hran můžete vidět na obrázku 4.10.

4.9 Označování a hledání vláken

Cílem práce je měření průměrů vláken. Co vlákno z pohledu počítačového vidění je a co není si musíme definovat. Pracujeme s rastrovou grafikou a nemáme tak k dispozici vlákno ve formě objektu. Proto budeme zjišťovat, zda se daná množina

obrazových bodů nachází na vlákne. K tomu využijeme binarizovaný snímek vláken, kde jsou oddělena vlákna a pozadí. Tento předpoklad ale nestačí, neboť se vlákna protínají. Musíme tedy zohlednit hranice vláken získané metodami detekce hran v obraze.

4.9.1. Detekce vláken úsečkou

Jednou z metoda ručního měření průměrů je označení vlákna, případně více vláken, tažením kurzoru myši přes vlákna ve snímku. Mezi koncovými body dráhy kurzoru je vytvořena pomyslná úsečka. Mezi body, kde se úsečka protne s hranou se potenciálně nachází vlákno. Jelikož je možné protnou více hran, nemusí dvojice průsečíků patřit jednomu vláknu, ale může se jednat o vnější hrany dvou různých vláken. Přítomnost vlákna je nutné ověřit z binarizovaného snímku vláken. Pro hledání průsečíků je nutné znát všechny body úsečky. K tomu musíme znát obecnou rovnici přímky $p: ax+by+c=0$. Koeficienty a , b a c zatím neznáme. Z krajních bodů úsečky zjistíme směrový vektor, který otočením souřadnic a změnou znaménka jedné ze souřadnic získáme normálový vektor. Ten odpovídá koeficientům a a b . Koeficient c vypočítáme dosazením jednoho z bodů do rovnice $c=(-1)*(a*x_1+b*y_1)$.

$$\text{Z rovnic } y_i = \frac{a*x_i+c}{-b}, i \in \langle x_A, x_B \rangle, \text{ resp. } x_i = \frac{b*y_i+c}{-a}, i \in \langle y_A, y_B \rangle \text{ je možné}$$

vypočítat všechny body přímky mezi body A a B. Pokud je sklon přímky v intervalu $315^\circ - 45^\circ$ nebo $135^\circ - 225^\circ$ dopočítává se souřadnice y , jinak souřadnice x . Pokud by se nerozlišoval sklon, nedocházelo by pro určité úhly k dopočítání všech bodů. Obrázek 4.11 názorně ukazuje rasterizaci úsečky s vhodně a nevhodně zvoleným výpočtem bodů.

Z snímku detekovaných hran se zjistí průsečíky s úsečkou. Hrana bývá tlustší než jeden pixel. Jako mezní bod je brán vnitřní pixel. Vzdálenost bodů odpovídá průměru vlákna pouze tehdy, pokud je vlákno označeno tahem myši kolmo, čehož je v praxi těžké dosáhnout. Z tohoto důvodu aplikace zkouší mezi body vepsat kružnici.

Body kružnice jsou počítány podle vzorce $x=x_0+r*\cos(\varphi)$, $y=y_0+r*\sin(\varphi)$. Střed kružnice je určen bodem v polovině vzdálenosti bodů a poloměr nastaven

na 2 pixely. Dokud všechny body kružnice leží na vláknu a zároveň se kružnice nedotýká hran je poloměr zvyšován.

4.9.2. Automatická detekce v celém snímku

V případě automatického detekování vláken je postup obdobný. Střed je určen jako bod v obraze určený detektorem a nemá přímou souvislost s vláknem, narozdíl od detekce po úsečce, kde je střed určen v polovině vzdálenosti mezi hranami. Určený střed je pouze výchozí bodem pro algoritmus popsany digramem Obrázek 4.11. Z binarizovaného snímku vláken se určuje, zda se střed a body kružnice nacházejí na vlákně. Pro práci s hranami je kružnice rozdělena na půlkvadranty. Na základě toho, v jakých půlkvadrantech kružnice protíná hranu je spočítán úhel posunutí středu kružnice. Posun je možný v případě dotyku hran v méně než pěti půlkvadrantech a pokud nejsou půlkvadranty protilehlé. Po posunutí se zvětší poloměr postup se opakuje, dokud je možný posun a kružnice je na vlákně.

4.9.3. Realizace přímky a kružnice v C#

Pro úsečku byla navržena třída `MyLine`. Při vytváření objektu se spočítají koeficienty obecné rovnice přímky. Třída dále obsahuje metodu pro výpočet všech bodů úsečky a metodu pro procházení úsečky a hledání průsečíků, která zároveň zajišťuje vytváření kružnic mezi průsečíky.

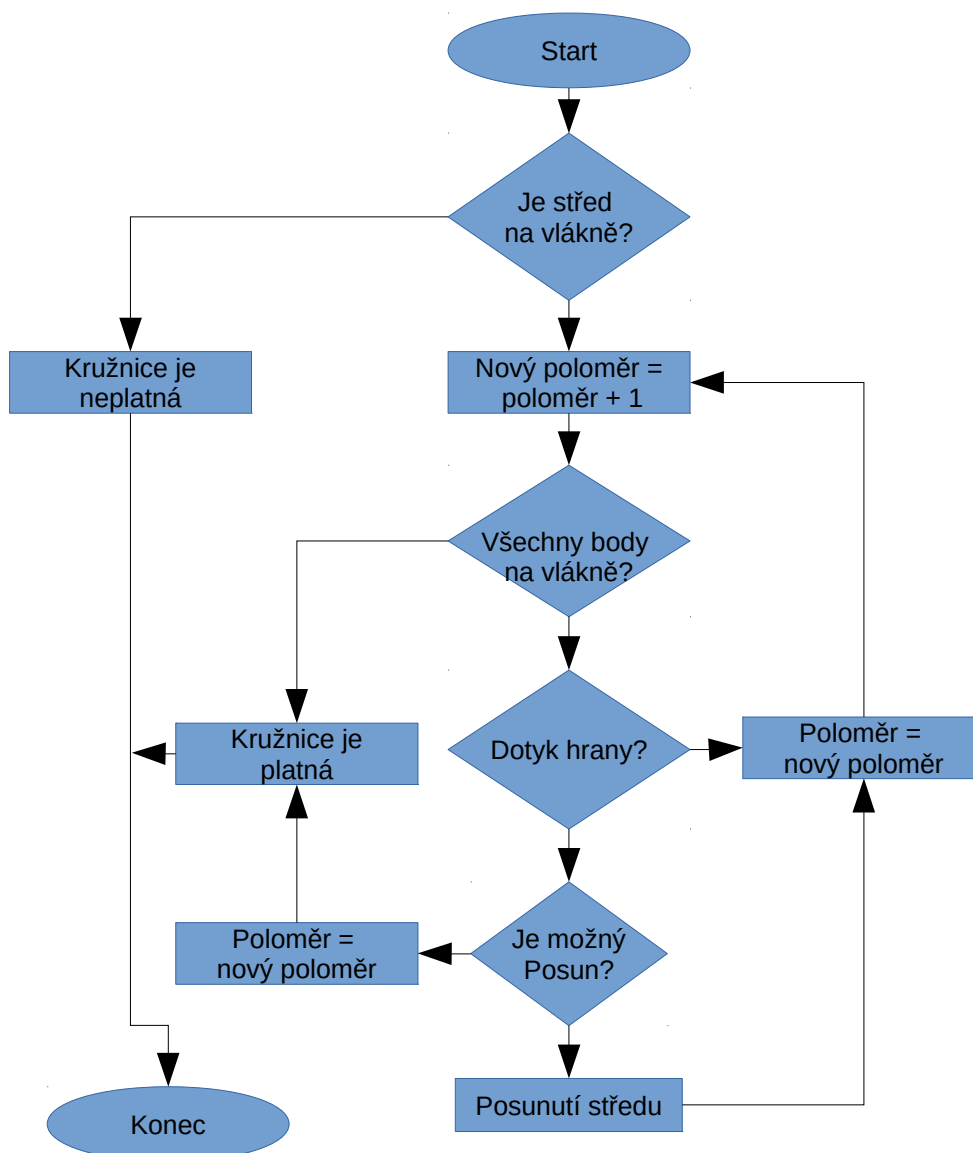
Třída `Graphics` zapouzdřuje v C# vykreslování grafických objektů na uživatelské rozhraní [11]. Metodou `drawLine` této třídy je v průběhu tahu myši úsečka pro lepší orientaci vykreslována na snímek.

Kružnice jsou v aplikaci reprezentovány třídami `Circle` a `CircleAuto`. Tyto třídy implementují rozhraní `ICircle`, které definuje hlavičky metod, které musí třídy implementovat. Použití rozhraní umožňuje přiřazení kružnic různých tříd do jedné kolekce. Kružnice detekovaná úsečkou je v reprezentována třídou `Circle`. Vlastnostmi třídy je bod určující střed kružnice, poloměr, barva a id a snímky hran a binarizovaných vláken ve formě pole bytů. Metoda `drawEllipse` třídy `Graphics` neumí vykreslit kružnici podle středu a poloměru, ale podle levého horního rohu a rozměrů obdélníku obklopujícího elipsu. Je tedy nutné od obou souřadnic středu odečíst poloměr a jako rozměry obdélníku použít průměr kružnice.

Třída má také vlastnost `RadiusToNm` vracející průměr kružnice přepočítaný pomocí měřítka na celé nanometry, který je použit v tabulce naměřených hodnot a při výpisu průměrů do obrázku.

Třída `CircleAuto` reprezentuje kružnici detekovanou automatickým detektorem. Je rozšířením třídy `Circle`. Poloměr a případně novou pozici středu určuje konstruktor třídy. Konstruktor využívá výše popsaného algoritmu (Obrázek 4.11).

Všechny nalezené kružnice jsou přidávány do kolekce typu `BindingList`, která slouží jako datový zdroj pro seznam průměrů a jejich případný export.



Obrázek 4.11: Algoritmus posunu a zvětšování kružnice reprezentující průměr vlákna

Měřítka je v aplikaci reprezentováno třídou `Meritko`, které uchovává délku měřítka v pixelech a hodnotu délky v mikrometrech. Koeficient přepočtu pixelů na nanometry je pak počítán jako tisícinásobek poměru mikrometrů ku pixelům.

4.10 Zobrazování výsledků měření a jejich export

Všechny kružnice reprezentující vlákna jsou ukládány do seznamu typu `List<ICircle>`. Tento seznam je nastaven jako zdroj dat pro komponentu `DataGridView`, která zobrazuje tabulku naměřených hodnot. Díky tomu není nutné přidávat data do tabulky programově, ale změny v seznamu se okamžitě projeví i v tabulce. Výběrem kružnice v tabulce se kružnice ve snímku zvýrazní žlutou barvou. Komponenta rovněž zachytává stisk klávesy `Delete`, nebo `Backspace` a zvolenou kružnici odstraňuje.

Aplikace umožňuje export naměřených hodnot do souboru v jednoduchém formátu CSV (Comma-separated values, hodnoty oddělené čárkami) pro jejich použití v jiných aplikacích. Formát je popsán v dokumentu RFC 4180 organizace IETF [12]. Export je realizován jako uložení textového řetězce do souboru. Do ukládaného textového řetězce jsou postupně přidávány řádky ve formátu `id; průměr v nanometrech; nový řádek`. Druhou formou exportu jsou data ve formátu `průměr v nanometrech; počet; nový řádek`, řazená vzestupně podle průměru. Tyto hodnoty lze použít pro tvorbu grafu například v tabulkových procesorech.

4.11 Zobrazení rozložení průměrů vláken

Důležitou funkcí aplikace je zobrazení četnosti průměrů vláken pomocí grafu. Na osu x jsou vynášeny hodnoty průměrů v nanometrech (není-li definováno měřítko, hodnoty jsou v pixelech). Na ose y pak četnosti jednotlivých průměrů. Graf je vykreslován komponentou `Chart`. Každému bodu je nadefinován text vlastnosti `tooltip`, který se zobrazí po najetí kurzorem na místo v grafu. Je tak možné přesně odečíst konkrétní hodnoty.

4.12 Ukládání a načítání projektu

Pokud má určitá skupina snímků podobné vlastnosti je vhodné, aby se nemuselo znovu nastavovat měřítko a hodnoty pro prahování či detekci hran. Může také nastat situace, kdy bude uživatel chtít změřit vlákna v jiné části snímku, nebo bude chtít znovu vidět naměřená data ve zdrojovém snímku. Pro tyto situace je v aplikaci implementována možnost uložit, popř. nahrát rozpracovaný projekt.

Pro zjednodušení ukládání dat rozličných formátů, jako je měřítko, naměřené hodnoty a jednotlivé vrstvy obrázku je vytvořena třída `Projekt`. Instance této třídy sdruží potřebná data do jednoho objektu, který je jednoduše serializovatelný². Metody pro serializaci jsou součástí `.NET`. Serializace lze realizovat více způsoby. Serializace do XML, JSON, případně jiných formátů tohoto typu není vhodná. Uložený soubor lze snadno modifikovat a tím strukturu dat poškodit. Problematická by byla také serializace obrázků datového typu `Bitmap`. Proto byla zvolena binární serializace pomocí třídy `BinaryFormatter` a jejích metod `Serialize` a `Deserialize`. Aby bylo možné serializaci projektu provést, musejí všechny serializované třídy obsahovat atribut `[Serializable]`. Většina objektů v jazyce C# je serializovatelná, ale například třída `Pen` využívaná při vykreslování kružnic a úseček serializovatelná není a musí být rozdělena na hodnoty udávající šířku a barvu. Proto třídy objektů, které jsou vykreslovány na snímek (kružnice, úsečky, ...) vlastnost datového typu `Pen` vůbec nemají. Při vykreslování je pak objekt typu `Pen` sestaven dynamicky z hodnot šířky a barvy.

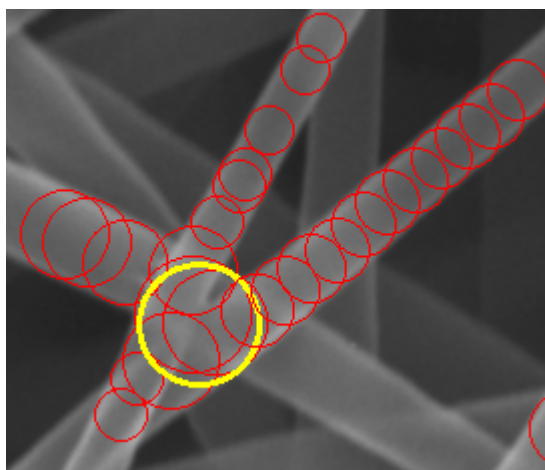
Uživatel má možnost projekt uložit, nebo uložit jako, stejně jako je zvyklý například z textového procesoru. Volba `Uložit` nabídne při prvním použití klasický dialog pro výběr adresáře a názvu souboru pro uložení. Pokud je projekt již uložen, tak další použití této volby automaticky přepíše soubor vytvořený při prvním použití. Volba `Uložit jako` vyvolá dialog pro volbu umístění vždy. Ukládané soubory mají příponu `.mvp`.

² Serializace slouží pro uchování stavu instance objektu. Objekt se převádí na posloupnost bitů, kterou můžeme uložit do souboru, případně přenést po síti a podobně.

Při načítání projektu se uživateli zobrazí opět klasické dialogové okno pro výběr souboru s omezením na soubory s příponou .mvp. Vybraný soubor je deserializován a jsou nastaveny příslušné proměnné.

4.13 Snížení počtu naměřených hodnot

Výsledky naměřené pomocí automatického měření obsahují mnohdy zbytečně hustě naměřené oblasti vláken, popř. je z důvodu nevýrazných hran zobrazena oblast zpravidla v místech křížení vláken.



Obrázek 4.12: Naměřené hodnoty na křížení vláken

V aplikaci je realizováno několik způsobů manuálního odstranění těchto nepřesností. Funkce se volí v pravé části aplikace vedle seznamu naměřených hodnot.

První funkcí je určení oblasti kliknutím myši, kdy jsou odstraněny všechny kružnice obsahující zvolený bod. Odstranění probíhá pomocí jednoduchého algoritmu. Bod se nachází v kružnici, pokud je vzdálenost jejího středu a zvoleného bodu menší, než její poloměr. Postupným procházením kolekce naměřených dat je vytvořena nová kolekce bez nechtěných dat.

Na podobném principu funguje funkce redukce oblasti. V tomto případě nejsou odstraněny všechny kružnice, ale náhodně vybrané kružnice v počtu daném vzorcem

$$P_n = \lfloor P_c / 2 \rfloor,$$

kde P_n je počet kružnic k odstranění a P_c je počet kružnic obsahujících zvolený bod

Náhodné kružnice jsou vybírány funkcí `next` třídy `Random`.

Poslední funkcí je mazání oblasti určené čtvercovým výběrem. Při výběru oblasti tažením kurzoru myši je čtverec vykreslován funkcí `drawRectangle`.

5 Vzhled aplikace a její ovládání

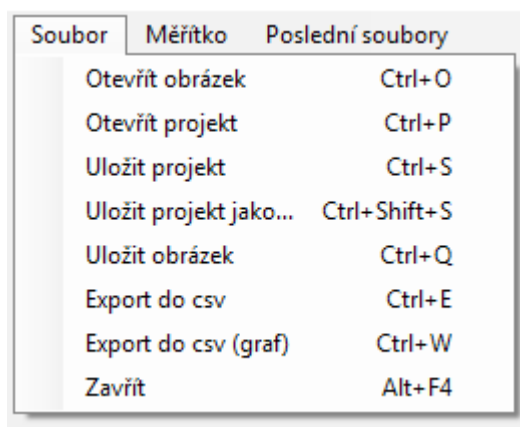
Aplikaci není nutné instalovat. Podmínkou spuštění aplikace je přítomnost .NET Frameworku verze 4 a vyšší. Framework lze bezplatně stáhnout ze stránek Microsoftu. Aplikace se spustí souborem `MereniVlaken.exe`. Po spuštění je aplikace roztažena na celou obrazovku, ale rozměry okna lze libovolně měnit. Pro správné zobrazení všech ovládacích prvků je nutné rozlišení obrazovky minimálně 1024×768 obrazových bodů.

5.1 Načtení snímku

Aplikace sestává z jednoho okna. V horní části se nachází rozbalovací menu, které slouží pro otevírání snímků a projektů, k jejich ukládání a exportu výsledků do CSV souboru. Z menu je rovněž možno vyvolat formulář pro úpravu hodnoty měřítka.

Chceme-li měřit průměry vláken, musíme nejprve načíst snímek. Vyvolání dialogu pro výběr souboru docílíme výběrem volby `Otevřít` z rozbalovacího menu, případně klávesovou zkratkou `Control + O`. Pokud otevíráme soubor, který jsme v poslední době zpracovávali, můžeme si urychlit výběr souboru použitím menu `Poslední soubory` z horního menu.

Po výběru snímku začne aplikace snímek předzpracovávat pro měření. Předzpracování může trvat řádově jednotky až desítky sekund, v závislosti na velikosti snímku a výkonu uživatelského počítače. Během předzpracování je zobrazeno okno upozorňující na probíhající zpracování a nelze s aplikací pracovat.



Obrázek 5.1: Rozbalovací menu

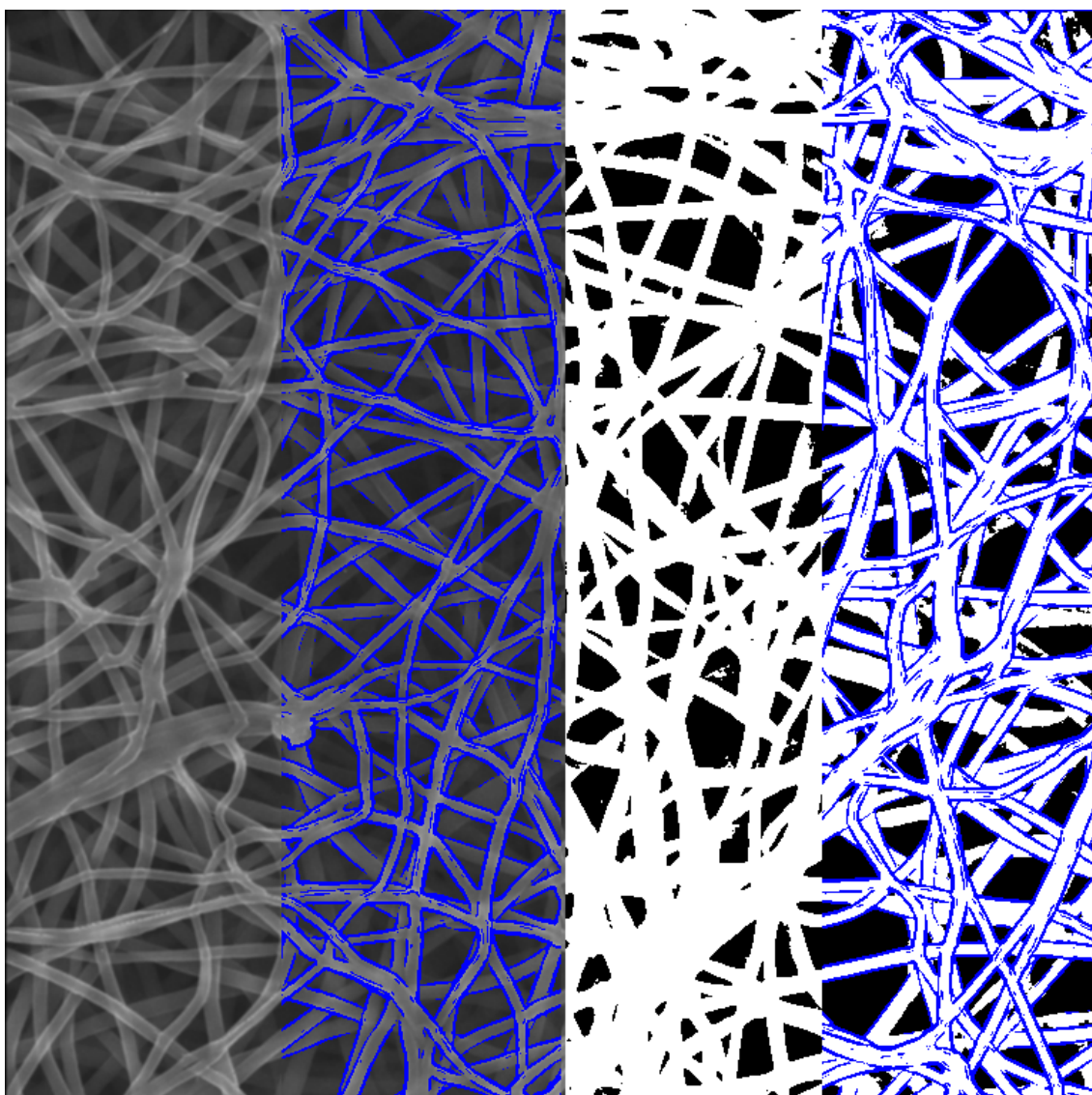
5.2 Úprava snímku před měřením

Po dokončení předzpracování je doporučeno ověřit, zda jsou hodnoty pro prahování snímku a hledání hran vhodně nastaveny. K zobrazení těchto dvou vrstev slouží přepínače pod obrázkem.



Obrázek 5.2: Ovládací panel detekce vláken

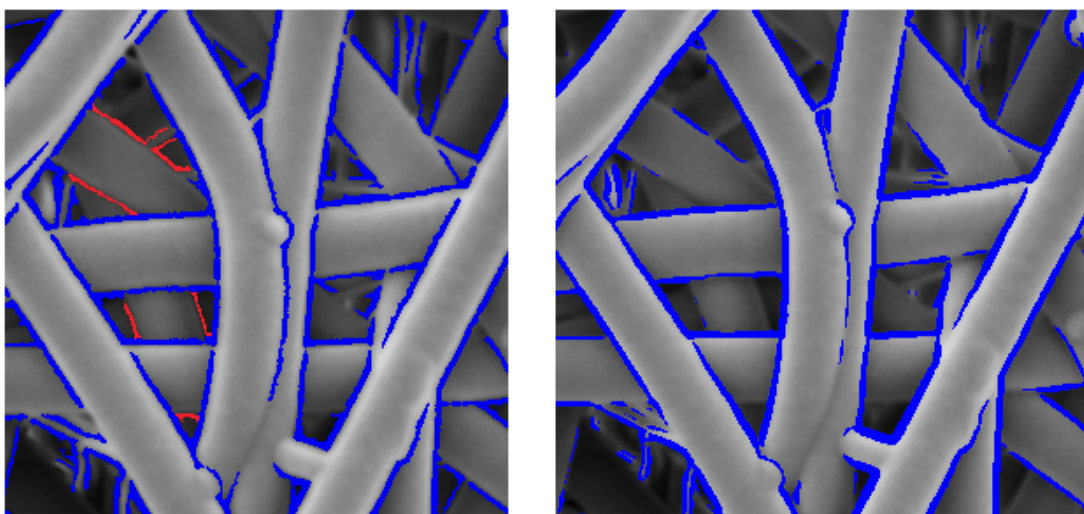
Pro dosažení co nejlepších výsledků měření, je nutné hodnoty ručně doladit. Jednotlivé vrstvy snímku můžeme přepínat ve spodní části aplikace. Obrázek 5.3 ukazuje, jak se od sebe liší původní snímek, původní snímek s hranami a prahovaný snímek s hranami a bez nich.



Obrázek 5.3: Jednotlivé vrstvy snímku

Pro různé sady snímku se ideální hodnoty liší. Hodnoty pro prahování snímku a hran můžeme měnit posuvníky pod snímkem (Obrázek 5.2). Změny se na snímku projevují okamžitě.

Pro detekci hran lze využít dvou přístupů. Použití hranového detektoru, nebo adaptivního prahování. Každý přístup je vhodný pro jiný typ snímku. Obecně lze říci, že detekce hranovým detektorem je více úspěšná než adaptivní prahování na snímcích, kde se jas vláken a pozadí příliš neliší. Naopak adaptivní prahování nalezne více hran, a tím umožní přesnější měření, ve snímcích s výraznými vlákny. Obrázek 5.4 červeně zobrazuje hrany, které hranový detektor nenalezl, zatímco adaptivní prahování ano.

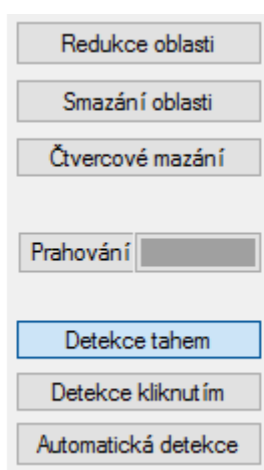


Obrázek 5.4: Hrany nalezené adaptivním prahováním a hranovým detektorem

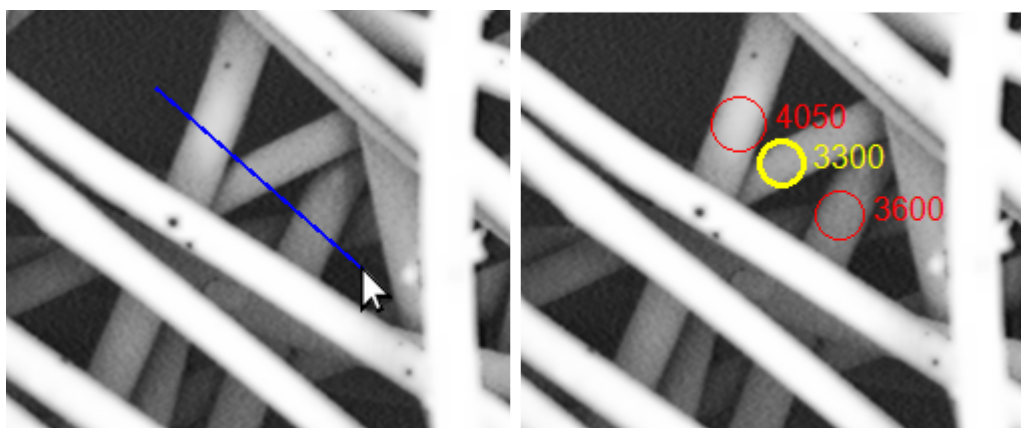
Síla a detekce hran u hranového detektoru se nastavuje posuvníkem Hrany. U adaptivního prahování lze ovlivnit kvalitu hran odfiltrováním malých objektů, které tvoří hrany. K tomu slouží posuvník Filtrování malých objektů (Obrázek 5.2).

5.3 Měření vláken

Aplikace umožňuje 3 způsoby měření. Způsoby se volí v pravé části aplikace. První variantou je měření detekce tahem myši přes vlákno. Stisknutím levého tlačítka myši se zahájí vykreslování přímky, kterou je vlákno, případně více vláken, protnuto. Uvolněním tlačítka myši jsou na zadané přímce detekována vlákna a změřeny jejich průměry. Průběh měření a jeho výsledek zobrazuje obrázek 5.6



Obrázek 5.5: Panel pro výběr způsobu měření a redukce hodnot



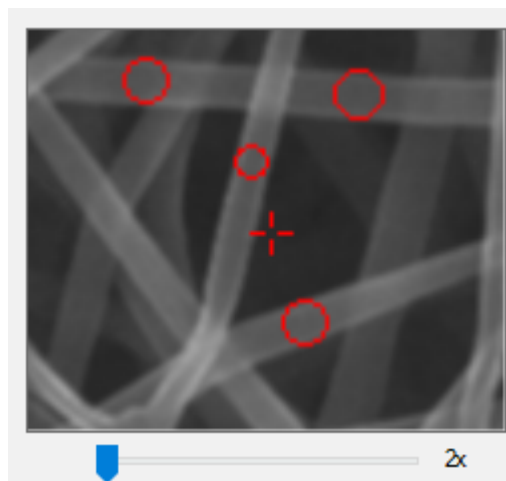
Obrázek 5.6: Ukázka měření tahem myši

Druhou variantou je detekce v místě zadaném poklepáním myši. Tento způsob lze s úspěchem použít v případě, že detektor ve zvoleném místě nerozpoznal jednu, případně obě hrany. Poklepáním na vlákno zadáme aplikaci střed kružnice a postupným zvětšováním průměru a posunem středu dojde k vykreslení kružnice a tím změření průměru vlákna.

Poslední možností je použití automatické detekce vláken v celém snímku. Před započítím měření je vhodné upravit krok detektoru posuvníkem v dolní části aplikace (Obrázek 5.2). Pro snímek obsahující mnoho tenkých vláken je vhodné zvolit malý krok. U snímků s větším přiblížením je vhodné krok zvětšit, jinak dojde k naměření zbytečně velkého počtu hodnot.

Přesnost měření, či výběru oblasti může být ověřena zvětšovacím okénkem v pravé dolní části aplikace. Okénko podle nastavení zvětšuje snímek na pozici myši dvakrát až pětkrát, v závislosti na nastavení posuvníku pod okénkem.

Průměry jednotlivých změřených vláken mohou být vepsány do obrázku volbou Průměry do obrázku. Průměr vlákna je pak zobrazen napravo od kružnice.



Obrázek 5.7: Přibližovací okénko

5.4 Výsledky měření

Naměřené hodnoty se zobrazují v tabulce v pravé části aplikace. Tabulka obsahuje dva sloupce. ID a průměr v nanometrech. Pokud není zadáno měřítko, místo průměru je zapsána zkratka „NM“. Po najetí myši na zkratku je zobrazen komentář „Není zadáno měřítko“. Pod tabulkou jsou dvě tlačítka. Jedno pro smazání všech naměřených hodnot a další pro mazání zvolené hodnoty. Aktivní hodnotu je též možné smazat stiskem klávesy `delete` nebo `backspace`. Vedle tlačítek je vypsán počet naměřených položek.

Hodnoty lze redukovat také přímo ve snímku, a to třemi způsoby. U prvních dvou způsobů, tj. „Redukce oblasti“ a „Smazání oblasti“ se určí kliknutím myši bod a všechny kružnice obsahující ten bod budou odstraněny, respektive snížen jejich počet. Třetí možností je „Obdélníkové mazání“, kde se tahem myši určí oblast, která bude odstraněna. Tato možnost je vhodná pro odstranění oblastí, kde se vlákna překrývají, a podobně.

Z hlavního menu je možné exportovat naměřené hodnoty do CSV souboru a to ve dvou provedeních. Jako netříděný seznam naměřených hodnot a nebo jako seznam četností jednotlivých průměrů. Tyto volby je také možné obsluhovat klávesovými zkratkami `Control + E` a `Control + W`.

Id	Průměr
0	800
1	1100
2	900
3	700
4	1200
5	1400
6	1800
7	1700
8	1600
9	2000
10	2200
11	2200
12	2300
13	2400

Smazat vše Smazat

Obrázek 5.8: Tabulka naměřených hodnot

5.5 Zadání měřítka

Pro přepočet pixelů na nanometry je nutné nastavit měřítko. To se provádí stiskem tlačítka *Nové nastavení* v dolní části aplikace a označením měřítka v obrázku tahem myši. V potaz je brán pouze rozdíl x-ových souřadnic. Poté je nutné zadat délku úseku v mikrometrech. Obě hodnoty se dají upravit vyvoláním formuláře *Měřítko* v horním menu.

Zadání měřítka

Hodnota v pixelech

428

Hodnota v μm

20

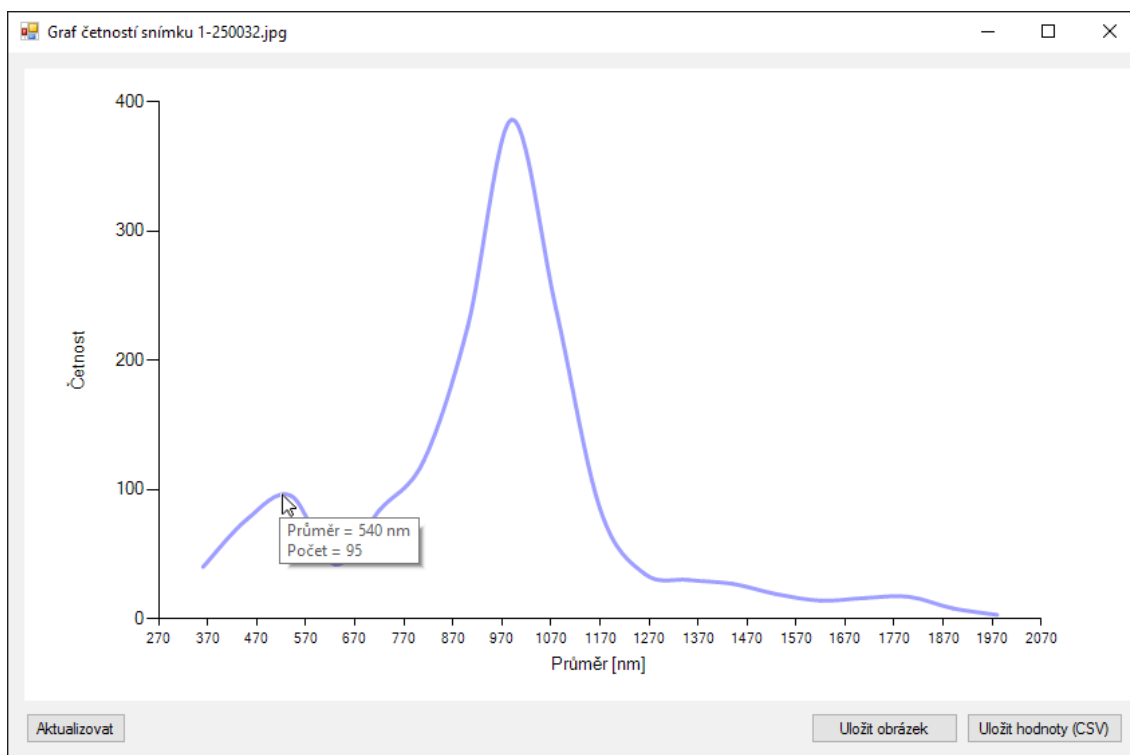
Potvrdit

Zrušit

Obrázek 5.9: Formulář pro ruční editaci měřítka

5.6 Zobrazení četností formou grafu

Důležitou funkcí aplikace je zobrazení rozložení četností ve formě bodového grafu. Pro jeho zobrazení musí být naměřené nejméně dvě hodnoty. Zobrazení grafu může být vyvoláno z kontextové nabídky **Nástroje** nebo klávesovou zkratkou **Control + G**. Graf se zobrazí v samostatném okně. Pokud došlo ke změnám v měření, je možné graf aktualizovat. Při najetí kurzorem myši na čáru grafu dojde vedle kurzoru k zobrazení hodnoty a četnosti kružnic daného průměru. Tlačítka pod grafem můžeme graf uložit jako obrázek ve formátu **PNG**, případně hodnoty exportovat v **CSV** formátu pro použití v tabulkovém procesoru.



Obrázek 5.10: Ukázka grafu rozložení vláken

6 Závěr

Cílem diplomové práce bylo seznámit se s problematikou zpracování rastrových obrázků, zejména s detekcí hran. Na základě těchto znalostí vytvořit aplikaci, která umožní měřit průměry vláken ze snímků pořízených mikroskopem a následně pracovat s výsledky měření. Teoretická část práce je zaměřena zejména na potlačení šumu a detekci hran.

Realizovaná aplikace zjednodušuje měření průměrů vláken pro získání představy o složení materiálu. Aplikace je zaměřena na řešení konkrétního úkolu a přináší výhody oproti existujícím obecně zaměřeným aplikacím. Popisovaná aplikace řeší nedostatek v současnosti používané aplikace Tescan Atlas, kde je měření prováděno ručním označováním vlákna. Zároveň nevyžaduje po uživateli znalost metod obrazové analýzy. Předkládaná aplikace vlákna detekuje sama a uživatel pouze upřesňuje detekované hodnoty. Měření průměrů pak probíhá označováním oblasti požadovaného měření. Součástí diplomové práce je i popis ovládání a práce s aplikací.

Aplikace splňuje všechny požadavky dané zadáním. Umožňuje ruční i automatické měření průměrů vláken. Možnost manuální úpravy některých hodnot a umožňuje optimalizaci detekování vláken i při změně vlastností zdrojových snímků. Práci s různými sadami snímků usnadňuje možnost uložení nastavení pro charakteristické sady.

Nabízí se rozšíření aplikace o možnost dávkového zpracování snímků z adresáře a přepínání mezi nimi v rámci aplikace, což by umožnilo i hromadné statistické zpracování celé sady. Toto rozšíření je však nad rámec rozsahu této práce.

Seznam použité literatury

- [1] COOK, John D. *Three algorithms for converting color to grayscale* [online]. [cit. 3. 2. 2016]. Dostupné z: <http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>
- [2] 2D Signall Processing. *EE502, Dublin City University* [online]. [cit. 8. 2. 2016]. Dostupné z: <http://www.eeng.dcu.ie/~ee502/EE502s4.pdf>
- [3] PARKER J.R. *Algorithms for Image Processing and Computer Vision*. Wiley Publishnig, 2011. ISBN: 978-0-470-64385-3
- [4] Adaptive threshold. *RoboRealm*. [online]. [cit. 15. 2. 2016]. Dostupné z: http://www.roborealm.com/help/Adaptive_Threshold.php
- [5] Image Edge Detection. *SoftwareByDefault*. [online]. [cit. 15. 2. 2016]. Dostupné z: <http://code.msdn.microsoft.com/Image-Edge-Detection-5c5a0dc2>
- [6] FIKKER, Jaroslav. *Šum v digitální fotografii* [online]. [cit. 1. 3. 2016]. Dostupné z: <https://www.fotoaparát.cz/clanek/323/sum-v-digitalni-fotografii-7193/>
- [7] HLAVÁČ, Václav, Miloš Sedláček. *Zpracování signálů a obrazů*. Praha: Vydavatelství ČVUT, 2002. ISBN 80-01-02114-9
- [8] *ImageJ Features* [online]. [cit. 25. 7. 2016]. Dostupné z: <https://imagej.nih.gov/ij/features.html>
- [9] Vision Development Module. *National Instruments*. [online]. [cit. 15. 4. 2016]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/209860>
- [10] HERCEG, Tomáš. *Úvod do .NET frameworku* [online]. [cit. 19. 5. 2016]. Dostupné z: <http://www.dotnetportal.cz/clanek/125/Uvod-do-NET-Frameworku>
- [11] Graphic class. *Microsoft .NET Framework Class Library*. [online]. [cit. 5. 4. 2016]. Dostupné z: [https://msdn.microsoft.com/en-us/library/system.drawing.graphics\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.drawing.graphics(v=vs.110).aspx)
- [12] SHAFRANOVICH, Y. *Common Format and MIME Type for CSV Files*. [online]. [cit. 19. 5. 2016]. Dostupné z: <http://tools.ietf.org/html/rfc4180>
- [13] *Adaptive Thresholding* [online]. [cit. 19. 5. 2016]. Dostupné z: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm>

Obsah přiloženého CD

- Elektronická verze této práce v PDF
- Aplikace MereniVlaken.exe
- Zdrojové kódy aplikace
- Ukázkové projekty pro načtení v aplikaci
- Ukázkové snímky