



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV FYZIKÁLNÍHO INŽENÝRSTVÍ

INSTITUTE OF PHYSICAL ENGINEERING

# APLIKACE PROGRAMOVACÍHO JAZYKA PYTHON V ANALÝZE OBRAZU A MODELOVÁNÍ FYZIKÁLNÍCH PROCESŮ GRAFENU

APPLICATION OF PYTHON PROGRAMMING LANGUAGE IN IMAGE ANALYSIS AND MODELING OF  
PHYSICAL PROCESSES OF GRAPHENE

## DIPLOMOVÁ PRÁCE

MASTER'S THESIS

## AUTOR PRÁCE

AUTHOR

**Bc. Kamil Stehlíček**

## VEDOUCÍ PRÁCE

SUPERVISOR

**doc. Ing. Miroslav Bartošík, Ph.D.**

**BRNO 2024**



# Zadání diplomové práce

Ústav:	Ústav fyzikálního inženýrství
Student:	<b>Bc. Kamil Stehlíček</b>
Studijní program:	Fyzikální inženýrství a nanotechnologie
Studijní obor:	bez specializace
Vedoucí práce:	<b>doc. Ing. Miroslav Bartošík, Ph.D.</b>
Akademický rok:	2023/24

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:

## **Aplikace programovacího jazyka Python v analýze obrazu a modelování fyzikálních procesů grafenu**

### **Stručná charakteristika problematiky úkolu:**

Python je vysokoúrovňový skriptovací programovací jazyk s otevřeným zdrojovým kódem (open-source), který v současnosti podporuje obrovské množství knihoven vhodných pro řešení problémů fyziky pevných látek. Jeho knihovny usnadňují například analýzu velkého množství dat, analýzu obrazu, numerické řešení diferenciálních rovnic i využití metod strojového učení. Diplomové práce je zaměřena na použití Pythonu při analýze obrazu galiových a galium nitridových shluků na grafenu, které se používají pro vývoj senzoru ultra-fialového záření. Python bude také použit k řešení difuzních rovnic šíření náboje na grafenové Hallově struktuře na oxidu křemičitém. Centrální část přitom bude tvarována do podoby vhodných nanostruktur grafen oxidem. Význam druhého úkolu spočívá ve využitelnosti takových struktur v jednoduchých elektronických nanozařízeních – senzorech, kvantových bodových kontaktech apod.

### **Cíle diplomové práce:**

1. Proveďte představení a literární rešerši uvedených fyzikálních problémů.
2. Analizujte Ga, GaN shluky na grafenu v závislosti na experimentálních parametrech.
3. Proveďte numerické řešení difuzních rovnic pro šíření náboje u 2D Hallové struktury grafenu tvarované pomocí GOx a umístěné na SiO<sub>2</sub>.
4. Z fyzikálního hlediska interpretujte dosažené výsledky.

### **Seznam doporučené literatury:**

STEWART, John M. Python for Scientists. Cambridge: Cambridge University Press, 2014. ISBN 978-1107061392.

VANDERPLAS, Jacob T. Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly, 2016. ISBN 1491912057.

MACH, Jindřich; PROCHÁZKA, Pavel; BARTOŠÍK, Miroslav, et al. Electronic transport properties of graphene doped by gallium. *Nanotechnology* [online]. 2017, 28(41), 415203 [cit. 2018-11-05]. DOI: 10.1088/1361-6528/aa86a4. ISSN 0957-4484. Dostupné z: 10.1088/1361-6528/aa86a4

ŠVARC, Vojtěch; BARTOŠÍK, Miroslav; KONEČNÝ, Martin; PIASTEK, Jakub; NEZVAL, David et al. Side charge propagation in simultaneous KPFM and transport measurement of humidity exposed graphene FET sensor. *Carbon*. 2023, roč. 215, č. 118471, s. 12.

KONEČNÝ, Martin, Miroslav BARTOŠÍK, Jindřich MACH, et al. Kelvin Probe Force Microscopy and Calculation of Charge Transport in a Graphene/Silicon Dioxide System at Different Relative Humidity. *ACS applied materials & interfaces* [online]. 2018, 10(14), 11987 [cit. 2018-11-05]. DOI: 10.1021/acsami.7b18041. Dostupné z: 10.1021/acsami.7b18041

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

---

prof. RNDr. Tomáš Šikola, CSc.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty



## **Abstrakt**

V této práci se zaměřujeme na vyhodnocování experimentálních dat pomocí programovacího jazyka Python napříč třemi různými fyzikálními úlohami zabývajícími se grafenem. Cíle práce vycházejí z praktických experimentů, při kterých využíváme gallium či nitrid gallia pro změnu elektrooptických vlastností grafenu nebo experimentů, které vyžadují simulaci šíření náboje v grafenové nanoelektronice. Tyto úlohy postupně využívají analýzu obrazu a numerické simulace. Teoretická část práce slouží jako rešerše a jako uvedení základních algoritmů zpracování obrazu a postupů v oblasti numerických simulací. Praktická část práce se pak zaměřuje na vyhodnocování úspěšnosti jednotlivých programů, jejich implementaci při praktickém vyhodnocení a vysvětlení experimentálních výsledků.

## **Summary**

In this thesis, we focus on evaluating experimental data using the Python programming language across three different physics problems dealing with graphene. The goals of the thesis are based on practical experiments that use gallium or gallium nitride to alter the electro-optical properties of graphene or experiments that require the simulation of charge propagation in graphene nanoelectronics. These tasks successively use image analysis and numerical simulations. The theoretical part of the thesis serves as a research and as an introduction to basic image processing algorithms and numerical simulation techniques. The practical part of the thesis then focuses on the evaluation of the success of each program, its implementation in practical evaluation and explanation of experimental results.

## **Klíčová slova**

Python, grafen, zpracování obrazu, simulace, difuze náboje, hallbar, gallium, nitrid gallia.

## **Keywords**

Python, graphene, image processing, simulation, charge diffusion, hallbar, gallium, gallium nitride.

STEHLÍČEK, K. *Aplikace programovacího jazyka Python v analýze obrazu a modelování fyzikálních procesů grafenu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2024. 51 s. Vedoucí diplomové práce doc. Ing. Miroslav Bartošík, Ph.D..



Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, pouze pod odborným vedením doc. Ing. Miroslava Bartošíka, Ph.D. a s použitím literatury, kterou cituji. Grafické podklady a snímky jsem zpracoval a vytvořil samostatně, pokud u konkrétního případu není uvedeno jinak.

Bc. Kamil Stehlíček



Děkuji doc. Ing. Miroslavu Bartošíkovi, Ph.D. za odborné vedení, konzultace a za veškeré cenné podněty při psaní této práce. Dále děkuji Ing. Ondřeji Špačkovi za cenné rady, Ing. Jakubovi Piastkovi, Ing. Vojtěchu Švarcovi a Bc. Jakubovi Šťastnému za poskytnutí experimentálních dat. Rád bych také z celého srdce poděkoval celé svojí rodině, všem nejbližším a přátelům, kteří mě po celou dobu studia podporovali a pomáhali mi. Na závěr děkuji také komunitě *Stack Overflow* za pomoc s vyřešením mnoha problémů.

Bc. Kamil Stehlíček



# Obsah

Úvod	3
<b>1 Teoretická část</b>	<b>4</b>
1.1 Grafen	4
1.1.1 Fyzika uhlíku	4
1.1.2 Geometrie grafenu	4
1.1.3 Fyzikální vlastnosti grafenu	5
1.1.4 Dopování grafenu	7
1.1.5 Výrobní procesy grafenu	7
1.1.6 Grafen s příměsmi	8
1.1.7 Hallbar	9
1.2 Metody zpracování obrazu	9
1.2.1 Prahování	9
1.2.2 Otsuova metoda prahování	10
1.2.3 Dilatace a eroze obrazu	11
1.2.4 Metoda lokálních maxim	12
1.2.5 Houghova kruhová transformace	13
1.2.6 Hledání kontur	14
1.2.7 Laplacián gaussianu ( <i>LoG</i> )	15
1.2.8 Analýza hlavních komponent	17
1.3 Difuze	18
1.3.1 Fickovy zákony	18
1.3.2 Difuzní rovnice ve 2D	19
1.3.3 Numerická řešení difuzní rovnice	19
<b>2 Praktická část</b>	<b>21</b>
2.1 Charakterizace galliových kuliček na grafenu	21
2.1.1 Uvedení problému	21
2.1.2 Požadavky na program	21
2.1.3 Příprava obrazu	22
2.1.4 Konkrétní kroky analýzy	23
2.1.5 Filtrování dat	25
2.1.6 Vyhodnocení výsledků	25
2.2 Simulace šíření náboje na <i>hallbaru</i>	28
2.2.1 Uvedení problému	28
2.2.2 Realizace výpočtu	28
2.2.3 Požadavky na program	29
2.2.4 Konkrétní kroky výpočtu	30
2.2.5 Simulace	31
2.2.6 Porovnání s experimentem	33
2.2.7 Vyhodnocení výsledků	34
2.3 Charakterizace gallium nitridových objektů na grafenu	34
2.3.1 Uvedení problému	34
2.3.2 Požadavky na program	35

## OBSAH

2.3.3	Konkrétní kroky analýzy obrazu . . . . .	35
2.3.4	Zjišťování orientace objektů . . . . .	36
2.3.5	Vyhodnocení výsledků . . . . .	39
<b>Závěr</b>		<b>42</b>
<b>3 Literatura</b>		<b>43</b>
<b>Seznam příloh</b>		<b>47</b>
	Příloha A: Souhrn simulací šíření náboje . . . . .	47
	Příloha B: Souhrn analýz obrázků gallium nitridových struktur . . . . .	50



# Úvod

Programovací jazyk Python zažívá velký rozkvět. Stává se stále více populárnějším díky své univerzálnosti, jednoduchosti a široké škále využití. Stává se z něj standard v oblasti datové analýzy a strojového učení, disponuje širokou škálou webových frameworků a nachází uplatnění i ve fyzice a vědě obecně. Python je open-source, intuitivní a snadno se učí. Disponuje rychle rostoucí komunitou vývojářů a podporou velkých technologických společností. Jeho popularita a především oblíbenost i mezi vědci stále roste a s ní i možnosti jeho využití v nejrůznějších oblastech, včetně fyziky.[1] Z těchto důvodů je Python využit i pro řešení problémů v této diplomové práci.

Společným jmenovatelem kromě Pythonu je pro všechny úlohy této práce grafen. Jedná se o přelomový materiál poprvé vyrobený v roce 2004.[2] Grafen je přelomový hlavně díky svým jedinečným fyzikálním vlastnostem.[3, 4] Má i mnoho aplikací, které sahají od senzorů plynů přes biosenzoriku až například po zpevňování materiálů.[5] Všechny tyto aplikace mají ale jedno společné úskalí. Každá z nich vyžaduje mnoho experimentů k tomu, aby se vyladila tak, aby byla komerčně či vědecky použitelná v praxi. Na tento problém se snažíme v této práci odpovědět a přijít s postupy, jak ulehčit práci experimentátorům alespoň v určitých oblastech aplikací grafenu.

Snažíme se analyzovat data a simulovat fyzikální procesy tak, aby měli experimentátoři méně práce s manuálním vyhodnocováním dat nebo aby měli numerickou představu pro srovnání s experimentem. Proto máme v této práci snahu vyvinout nástroje pro generování ucelených dat z experimentů, která může nadále experimentátor použít ve svůj prospěch a komplexnější analýzy.

V první úloze chceme analyzovat snímky galliových kuliček na grafenu. Zajímá nás především jejich poloha, velikost a vývoj těchto veličin v čase, z čehož můžeme vyvodit různé závěry pro experimenty jako například to, jestli se vzorky chovají lépe při menším, nebo větším pokrytí kuličkami.

Druhá úloha řeší numerickou simulaci šíření náboje na grafenové struktuře. Jedná se o geometrii velice podobnou klasickému plem řízenému tranzistoru. Stejně tak má i tato struktura využití založené na podobné bázi jako plem řízený tranzistor. Uplatnění nachází od jednoduchých senzorů magnetického pole až po biosenzoriku a mnoho dalších aplikací.

Poslední úloha je značně podobná první, jelikož pracujeme se snímky gallium nitridových struktur na grafenu. Opět se jedná o zpracování obrazu. Zde ovšem s tím rozdílem, že nepracujeme s časovou sérií snímků, ale s jedním snímkem, který reprezentuje výstup celého experimentu. Mimo to jsou gallium nitridové struktury odlišné od galliových kuliček. Mají osu symetrie (namísto kuličky, která má bod symetrie) a je proto možné jim přisoudit orientaci. Právě tuto orientaci chceme detekovat stejně tak jako pozici a velikost daných struktur.

# 1. Teoretická část

## 1.1. Grafen

### 1.1.1. Fyzika uhlíku

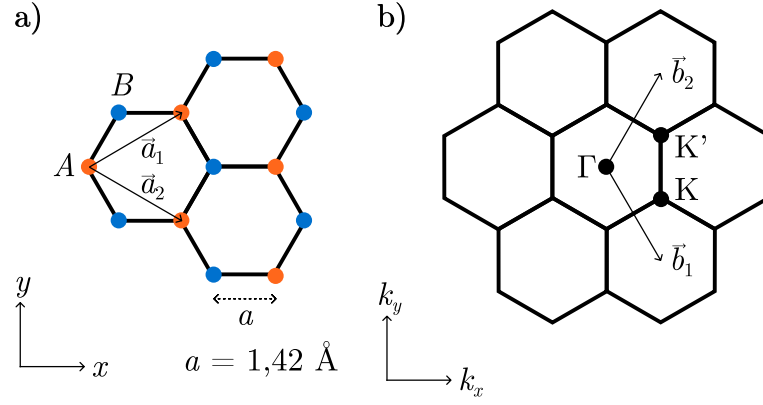
Uhlík je základním stavebním prvkem mnoha struktur, mezi které patří i grafen. Je tomu tak mimo jiné i díky jeho schopnosti hybridizace orbitalů. Hybridizace orbitalů je jev, při kterém dojde téměř ke sloučení orbitalů z jejich energetického hlediska. To vede k tomu, že se orbitály pak chovají energeticky velmi podobně i přes to, že byly původně na odlišných energiích.[6] Tím se vytvoří sada nových orbitalů, které dokáží tvořit mnoho různých vazeb.

Hybridizovat může molekula uhlíku do tří různých stavů. Jsou to stavy  $sp^3$ ,  $sp^2$  a  $sp^1$ . Stavy odpovídají postupně jednoduché, dvojně a trojitě chemické vazbě. Příkladem látek pro různé stavy hybridizací může být diamant [7] či methan ( $CH_4$ ) pro  $sp^3$ , grafen, ethylen ( $C_2H_4$ ) či benzen pro  $sp^2$  a na konec například acetylen ( $C_2H_2$ ) pro  $sp^1$  hybridizace.[6]

Pro nás je zajímavá podoba benzenového jádra, jelikož má velice podobnou strukturu vazeb jako grafen. Při pohledu na vazby můžeme pozorovat, že vazby mezi sousedními dvěma uhlíky a jedním vodíkem jsou realizovány třemi hybridizovanými orbitály  $sp^2$ . Každý uhlík navíc k tomu disponuje jedním volným nehybridizovaným orbitalem  $p$ . Zásadní je zde ta vlastnost, že všechny hybridizované orbitály (tedy i vazby) leží v jedné rovině. To umožňuje grafenu být 2D materiálem. Další vlastností, které je vhodné si všimnout, je natočení nehybridizovaného orbitalu  $p$ , který je kolmý na rovinu tří  $sp^2$  vazeb. Každý tento orbital přispívá jedním volným elektronem. To zapříčiňuje celkem výjimečné vodivostní vlastnosti grafenu.[8] Stejně tak je tento volný  $p$  orbital důvodem, proč se na kolmý směr k rovině grafenu mohou navazovat další látky. Tak poté vzniká dobře známý grafit, což je několik vrstev grafenu spojených volnými  $p$  orbitály.[8]

### 1.1.2. Geometrie grafenu

Grafen je monovrstva atomů uhlíku, uspořádaná do pravidelné šestiúhelníkové mřížky (neboli je v uspořádání včelí plástve - anglicky *honeycomb*).[9] Primitivní buňku grafenu tvoří dva bazové atomy uhlíku a dva bazové vektory  $\vec{a}_1$  a  $\vec{a}_2$ . [7] Vektory svírají vzájemně úhel  $60^\circ$  a vzdálenost dvou nejbližších uhlíků činí  $1,42 \text{ \AA}$ , jak je ukázáno na obrázku 1.1. Stejně tak poté definujeme bazové vektory reciprokého prostoru grafenu  $\vec{b}_1$  a  $\vec{b}_2$  vycházející ze středu Brillounovy zóny  $\Gamma$ . [7]



Obrázek 1.1: a) Atomová mřížka grafenu s bázovými atomy  $A$  a  $B$  a bázovými vektory  $\vec{a}_1$ ,  $\vec{a}_2$  a mřížkovou konstantou  $a = 1,42 \text{ \AA}$ . b) Reciproká mřížka grafenu s vektory  $\vec{b}_1$  a  $\vec{b}_2$  vycházejících ze středu Brillouinovy zóny  $\Gamma$ .

Konkrétně můžeme pak zapsat všechny vektory jako

$$\vec{a}_1 = \frac{a}{2} (3, \sqrt{3}), \quad (1.1)$$

$$\vec{a}_2 = \frac{a}{2} (3, -\sqrt{3}), \quad (1.2)$$

$$\vec{b}_1 = \frac{2\pi}{3a} (1, -\sqrt{3}), \quad (1.3)$$

$$\vec{b}_2 = \frac{2\pi}{3a} (1, \sqrt{3}), \quad (1.4)$$

kde  $a$  je mřížková konstanta o hodnotě  $1,42 \text{ \AA}$ .

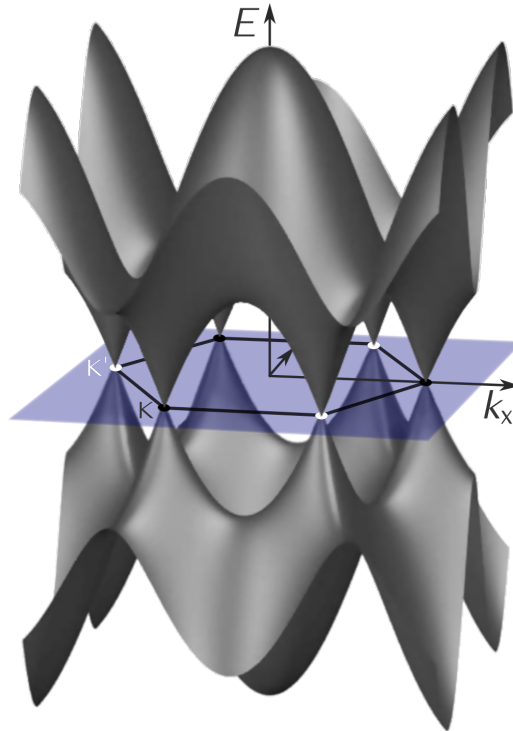
### 1.1.3. Fyzikální vlastnosti grafenu

Při řešení pásové struktury grafenu dostáváme následující disperzní relaci

$$E(k_x, k_y) \propto \pm \sqrt{1 + 4 \cos^2 \frac{\sqrt{3}ak_y}{2} + 4 \cos \frac{3ak_x}{2} \cos \frac{\sqrt{3}ak_y}{2}}, \quad (1.5)$$

kde  $a$  značí mřížkovou konstantu. Kladné znaménko ve výrazu odpovídá vodivostnímu pásu a záporné znaménko valenčnímu.[10] Vizualizaci takového pásu můžeme vidět na následujícím obrázku 1.2.

## 1.1. GRAFEN



Obrázek 1.2: Pásová struktura grafenu. Vidíme lineární chování v oblasti Diracových K, K' bodů. Převzato z [11].

Při pohledu na pásovou strukturu grafenu na obrázku 1.2 můžeme vidět zajímavý jev v bodech K a K' reciprokého prostoru. V těchto bodech, které se nazývají Diracovy body, dochází ke vzájemnému dotyku valenčního a vodivostního pásu. Můžeme grafen tedy považovat za polovodič s nulovým pásem zakázaných energií.[4]

V blízkosti Diracových bodů K a K' lze vyjádřit disperzní závislost elektronů grafenu jako

$$E(k) = \pm \hbar v_F k, \quad (1.6)$$

kde  $v_F$  je Fermiho rychlost.[4] Lze si povšimnout, že je to vzorec velmi podobný disperzní relaci fotonů (ve vakuu)  $E(k) = \hbar ck$ , kde je energie také přes konstantu lineárně závislá na velikosti vlnového vektoru  $\vec{k}$ . Z pásové struktury pak plyne mnoho vlastností grafenu.

### Optické vlastnosti

Jelikož má grafen pás zakázaných energií o nulové velikosti, absorbuje v oblasti všech vlnových délek počínaje infračerveným zářením až po viditelné světlo. Přesto má nízký absorpční koeficient přibližně 2,3 % pro světlo v těchto vlnových délkách, což je zapříčiněno jeho tenkostí. Teoretická hodnota absorpčního koeficientu je úměrná konstantě jemné struktury  $\alpha$ , se kterou se v kvantové fyzice běžně setkáváme a to konkrétně podle výrazu  $\pi\alpha \approx 2,3\%$ . I přes to, že je tedy grafen (monovrstvta uhlíku) v podstatě průhledná, absorpční vlastnosti má téměř identické jako klasický grafit.[4]

### Mechanické a tepelné vlastnosti

Tepelná vodivost grafenu činí v rovině chemických vazeb zhruba  $3000 \text{ W m}^{-1} \text{ K}^{-1}$ , což z grafenu činí skvělý tepelný vodič. Mimo to disponuje grafen i vynikajícími mechanickými

vlastnostmi, kde v tahových testech dosahuje Youngova modulu pružnosti až 1 TPa [3], což je v porovnání s hodnotami běžných ocelí (208 GPa [12]) velmi vysoká hodnota. Je však na pováženu, jak tato data interpretovat, jelikož grafen považujeme za 2D materiál, pro který neplatí běžný Hookův zákon a jeho odvození v podobě  $\sigma = E\epsilon$ . V práci [4] uvádí, že získaný Youngův modul o hodnotě 1 TPa pro grafen je přepočítán s použitím předpokladu, že tloušťka monovrstvy grafenu  $d$  činí 3,35 Å. Hodnota  $d$  však vychází pouze ze skutečnosti, že dvě monovrstvy grafenu v grafitu jsou vzdáleny právě o tuto hodnotu. Závěrem je ale třeba poznamenat, že vypočtená hodnota Youngova modulu pro grafen velice dobře odpovídá Youngovu modulu klasického 3D grafitu, který činí 1,02 TPa.[4]

## Elektrické vlastnosti

Výčet skvělých vlastností grafenu je završen jeho elektrickými vlastnostmi, kde disponuje možná největší škálou potenciálních aplikací. Pro charakterizaci elektrických vlastností nejlépe poslouží koncentrace nosičů náboje  $n$  a pohyblivost nosičů náboje  $\mu$ , které pro grafen činí  $n \approx 10^{11} \text{ cm}^{-2}$  a  $\mu \geq 10^5 \text{ cm}^2\text{V}^{-1}\text{s}^{-1}$ . [4]

Pro vysoce kvalitní grafen s  $n \approx 10^{12} \text{ cm}^{-2}$  pak nastává jev, kdy střední volná dráha  $l$  (podléhající vztahu  $l = (h/e)\mu(n/\pi)^{\frac{1}{2}}$ ) elektronů pohybujících se v grafenu je v řádech stovek nm.[13] To má pak za následek, že můžeme pozorovat projevy kvantové fyziky, jako například kvantový Hallův jev, vznik Landauových hladin a nebo například Schubnikovy-de-Haseho oscilace.[14]

### 1.1.4. Dopování grafenu

Grafen nám umožňuje manipulovat s Fermiho hladinou, která určuje, jakým způsobem grafen reaguje na elektrické pole. Posunutím této hladiny nahoru (n-dopování) přidáváme do materiálu elektrony, čímž se z něj stává vodič, ve kterém proud proudí díky elektronům. Naopak posunutím hladiny dolů (p-dopování) odebíráme elektrony a vytváříme díry, které se stávají nosiči náboje. [15]

Jedním ze způsobů, jak posunout Fermiho hladinu, je přiložení napětí na grafen.[16] Další možností je dopování grafenu různými prvky, které ovlivňují jeho pásovou strukturu. Například atomy gallia se shromažďují do kulatých útvarů a dopují grafen elektrony, čímž jej n-dopují a posunují Fermiho hladinu nahoru.[17]

Díky nastavení Fermiho hladiny můžeme z grafenu dělat vodič, nebo polovodič, což otevírá dveře k široké škále aplikací v oblasti elektroniky, optiky a dalších oblastech.

### 1.1.5. Výrobní procesy grafenu

I přes to, že byla existence stabilního grafenu dle dřívějších předpokladů fyziků vyvrácena [18], další teoretické výpočty tento výpočet upravily a uvedly, že grafen může existovat i ve stabilní formě díky předpokladům pokřivené mřížky a silných vazeb mezi uhlíky.[2] Určité výrobní techniky grafenu (primárně exfoliace) jsou na první pohled celkem jednoduché. Každopádně připravit chemicky čistou větší plochu kvalitního grafenu bez defektů je výzva, která je dnes často řešena.[19]

## 1.1. GRAFEN

### Mechanická exfoliace

Jeden z nejlhčích postupů přípravy grafenu je exfoliace probíhající pouze s pomocí grafitu a lepicí pásky. Jednoduchým nalepením a navazujícím odlepením pásky z kusu grafitu dokážeme extrahovat grafenové vrstvy. Na pásku dostaneme tímto způsobem mnoho různých celků grafenu. Převážná většina těchto celků má desítky až stovky vrstev. [7] Ovšem několik málo celků tvoří opravdovou monovrstvu žádaného grafenu. Celky jsme schopni přenést na substrát a na něm dále zkoumat, který z nich je žádanou monovrstvou. Tento postup je velice jednoduchý i časově nenáročný. Je to ovšem za cenu velké nespolehlivosti a faktu, že výrobní postup není nijak dobře definovaný. Tím pádem nelze ani zaručit příliš velkou opakovatelnost výroby jednotného vzorku grafenu.[7]

### Epitaxní růst

O něco sofistikovanější metoda využívá epitaxního růstu například vysokým ohřevem SiC substrátu. Zde už je postup výroby definovanější a předvídatelnější. Problém zde ovšem nastává tím, že nevíme, na jaké atomové rovině máme substrát ukončený (a tím pádem na jaké atomové rovině bude grafen růst). Vyrostlý grafen má pak jiné vlastnosti podle toho, jestli byl rosten na Si vrstvě, nebo na C vrstvě substrátu.[7]

### Chemická depozice z plynné fáze

Metoda chemické depozice z plynné fáze (*Chemical Vapor Deposition - CVD*) pro výrobu grafenu je považována zatím za jednu z nejefektivnějších z hlediska kvality a pokrytí větších ploch. Umožňuje vytvořit polykrystalický grafen o doménách velikostí až 100  $\mu\text{m}$ . Tento postup je rovněž dobře škálovatelný a definovaný.

Pro výrobu grafenu zjednodušeně probíhá chemická depozice z plynné fáze tak, že se do komory se substrátem vpouští methan v plynné fázi. Ten dosedá na povrch substrátu, kde se uhlík zachytává na substrát a vytváří postupně mřížku grafenu. Mimo to musí být ale odveden zbytkový vodík. Proto celý experiment probíhá za vysokých teplot okolo 1300° C.[20]

#### 1.1.6. Grafen s příměsmi

Jak bylo zmíněno v kapitole 1.1.3 grafen samotný disponuje rovnoměrnou absorpcí světla v rozsahu od 300 nm až do 2500 nm. Proto nelze grafen z tohoto pohledu použít jako příliš dobrý detektor infračerveného, viditelného a ultrafialového elektromagnetického záření. Při přidání nanostruktur na grafen se situace ovšem mění a můžeme dosáhnout například až 30% absorpce v oblasti viditelného světla díky přidání zlatých nanočástic.[21] Velkou výhodou je i laditelnost oblasti absorpce podle velikosti nanočástic.[21] Kromě zlata lze použít i jiné prvky, mezi kterými je například gallium. To dokáže upravovat plazmonickou odezvu grafenu na dopad elektromagnetického záření. Následnou mutací gallia jsme schopni vytvořit například gallium nitrid, který dokáže rozšířit (původně nulový) pás zakázaných energií grafenu tak, že můžeme takovou strukturu použít k detekci ultrafialového záření.[5] Právě aplikací galliových a gallium nitridových nanostruktur na grafenu se v této práci zabýváme.

### 1.1.7. Hallbar

Grafen má využití i v čisté podobě a to v případech, kdy z něj vytvoříme nějakou geometrickou strukturu. Buďto můžeme využít mechanických vlastností a vytvářet kompozitní materiály s příměsí grafenu, které dosahují skvělých mechanických vlastností.[22] Nebo můžeme využít elektrických vlastností grafenu a vytvořit takzvaný *hallbar* znázorněn na obrázku 2.12 (v této práci budeme mluvit i nadále o *hallbaru* i přes to, že se jedná o převzaté slovo z angličtiny). Jedná se o grafen s obdélníkovou geometrií na substrátu (nejčastěji SiO<sub>2</sub>). Geometrie grafenu je navíc obohacena o další 4 vývody umožňující čtyřbodové měření. Obecně je pak celý vzorek podobný typickému tranzistoru řízenému elektrickým polem (*Field effect transistor - FET*). K podobným účelům (jako polem řízený tranzistor) lze také *hallbar* využívat. Dále jsme schopni s *hallbarem* měřit Hallův jev, nebo i projevy kvantové fyziky (viz kapitola 1.1.3). Zajímavým aspektem je také to, jak se taková struktura chová například při různých vlhkostech nebo přiložení hradlového napětí. Jsme pak schopni vytvářet detektory plynu, nebo například senzory použitelné v biologických aplikacích reagující na určité molekuly.[23]

## 1.2. Metody zpracování obrazu

Úkol zpracování obrazu může zahrnovat mnoho požadavků odvíjejících se od toho, jaká data chceme vyčíst, nebo i od toho, z jakého obrazu chceme data vyčíst. V úlohách, které budeme dále řešit, chceme charakterizovat polohu, velikosti objektů a případně jejich orientaci. Popíšeme zde tedy metody, které se používají hlavně při řešení těchto úkolů.

V celé sekci 1.2 budeme uvažovat, že lze libovolný obrázek reprezentovat funkcí  $f(x, y)$  v prostoru souřadnic  $\{x, y\}$ , která vrací diskrétní hodnotu obrazu na dané diskrétní pozici  $(x, y)$  (typicky 0 až 255 pro 8 bitové obrázky).

### 1.2.1. Prahování

Prahování tvoří centrální operaci v metodách zpracování obrazu, kterou lze dělit dle dvou hlavních kritérií. Prvním z nich je, kde a jak je prahování aplikováno a druhým je, na jaké hodnoty mapujeme výsledný obraz.

Dle prvního kritéria dělíme prahování na globální a na lokální. Globální prahuje celý obrázek dle jedné jediné hodnoty, zatímco lokální se přizpůsobuje menším podoblastem obrázku.

Druhé kritérium zase rozděluje prahování například na prahování na nulu, prahování binární a další typy. Při zpracování obrazu se velice často uchylujeme k binárnímu prahování, jelikož pak výrazně zjednodušuje další práci s obrazem a detekcí objektů.

Můžeme tedy definovat globální binární prahování obrazu o hodnotě  $t$ , jako

$$f(x, y) = \begin{cases} 1 & \text{kde } f(x, y) \geq t \\ 0 & \text{jinak.} \end{cases} \quad (1.7)$$

Jak najít optimální hodnotu prahu  $t$  je ovšem výzvou, kterou řeší mnoho algoritmů. Často chceme v obrázcích rozlišit světlé popředí a tmavé pozadí.[24] Tímto problémem

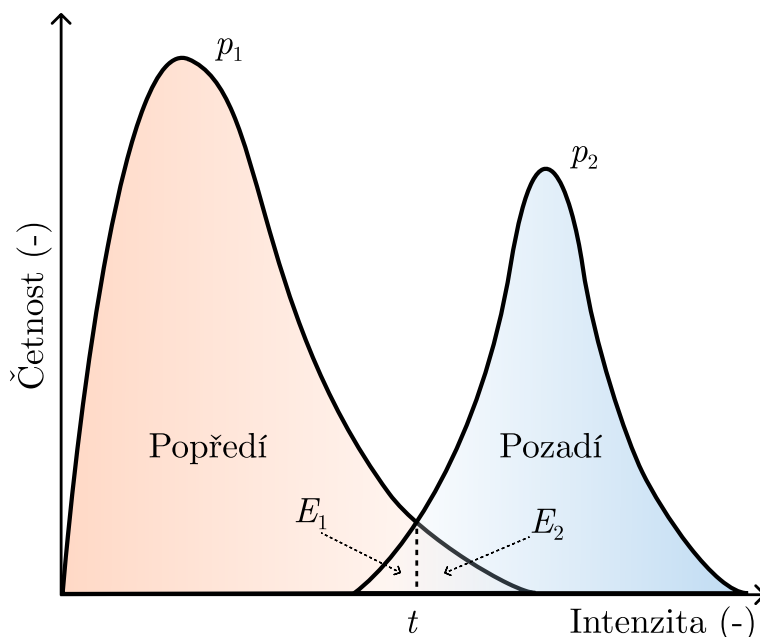


## 1.2. METODY ZPRACOVÁNÍ OBRAZU

se zabývá mnoho algoritmů, ovšem v této práci budeme využívat takzvanou Otsuovu metodu popsanou v následující kapitole 1.2.2.

### 1.2.2. Otsuova metoda prahování

Tato metoda vychází z myšlenky, že můžeme histogram obrázku aproximovat složením dvou oddělených funkcí (na které můžeme nahlížet jako na pravděpodobnostní rozdělení). První funkce slouží jako reprezentace světlého popředí obrázku, zatímco druhá jako tmavého pozadí viz obrázek 1.3.



Obrázek 1.3: Ukázkové rozdělení histogramu podle Otsuovy metody.

Z obrázku 1.3 tedy plyne, že je potřeba najít číslo  $t$  tak, aby byla obě rozdělení co nejdále od sebe. Vždy se dopustíme chyby kde kus rozdělení popředí utneme prahovou hodnotou  $t$  a naopak kus rozdělení pozadí vezmeme v potaz nad prahovou hodnotou  $t$ . Tuto chybu se snažíme v Otsuově metodě minimalizovat. Z obrázku 1.3 můžeme definovat chybnou klasifikaci popředí jako pozadí  $E_1(t)$  jako

$$E_1(t) = \int_{-\infty}^t p_2(x) dx. \quad (1.8)$$

Podobně zdefinujme chybnou klasifikaci pozadí jako popředí  $E_2(t)$  v podobě

$$E_2(t) = \int_t^{\infty} p_1(x) dx. \quad (1.9)$$

Dále předpokládáme, že obě rozdělení mají odlišnou pravděpodobnost výskytu. Konkrétně mějme pravděpodobnost výskytu  $P_1(t)$  pro rozdělení  $p_1(x)$  a pravděpodobnost výskytu  $P_2(t)$  pro rozdělení  $p_2(x)$ , kde  $P_1 + P_2 = 1$ . [25] Celkovou chybu pak můžeme vyjádřit jako

$$E(t) = P_1(t)E_1(t) + P_2(t)E_2(t). \quad (1.10)$$



Můžeme ukázat, že rovnice 1.10 lze transformovat do podoby, kde vystupuje součet vážených rozptylů  $\sigma_1^2(t)$  a  $\sigma_2^2(t)$  dvou rozdělení  $p_1(x)$  a  $p_2(x)$ . [26] Rovnici 1.10 lze pak uvést ve tvaru

$$\sigma^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t), \quad (1.11)$$

Pokud tento problém přeneseme do diskretních hodnot, definujeme vztahy pro diskretní rozdělení  $p(k)$  na základě hodnoty jasu  $k$

$$p(k) = \frac{n_k}{N}, \quad (1.12)$$

$$\omega_1(t) = \sum_{k=0}^{t-1} p(k), \quad (1.13)$$

$$\omega_2(t) = \sum_{k=t}^{L-1} p(k), \quad (1.14)$$

kde  $k$  označuje konkrétní úroveň jasu obrázku,  $n_k$  konkrétní počet pixelů s hodnotou jasu  $k$ ,  $N$  celkový počet pixelů a  $L$  celkový počet úrovní jasu. Tuto chybu  $\sigma$  se pak snažíme minimalizovat s ohledem na parametr  $t$ . [26] Problém minimalizace  $\sigma(t)$  lze ekvivalentně vyřešit maximalizací výrazu

$$\omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2, \quad (1.15)$$

kde platí následující vztahy

$$\mu_1(t) = \frac{\sum_{k=0}^{t-1} kp(k)}{\omega_1(t)}, \quad (1.16)$$

$$\mu_2(t) = \frac{\sum_{k=t}^{L-1} kp(k)}{\omega_2(t)}. \quad (1.17)$$

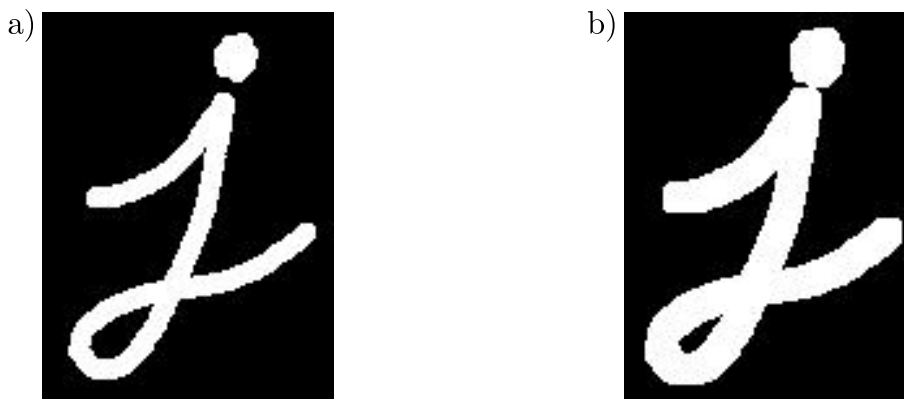
### 1.2.3. Dilatace a eroze obrazu

Při zpracování obrazu se setkáváme s takzvanými operacemi dilatace a eroze obrazu. Obě operace jsou charakterizovány svými jádry (maticemi) - specificky se jedná hlavně o velikost jejich jader, jelikož se typicky jádra definují jako matice o konstantní hodnotě. Pro obrazové aplikace lze dilataci, značenou jako  $\oplus$  zadefinovat následovně

$$(f \oplus B)(x, y) = \max_{(k,l) \in B} (f(x+k, y+l)), \quad (1.18)$$

kde  $B$  je jádro operace dilatace. [27] Zjednodušeně tedy vytváříme nový obraz tak, že procházíme jádrem přes původní obraz a za hodnotu nového pixelu považujeme maximální hodnotu původního obrazu, která leží v překryvu s jádrem operace. Operace dilatace je nejlépe vystihnuta na následujícím obrázku 1.4.

## 1.2. METODY ZPRACOVÁNÍ OBRAZU

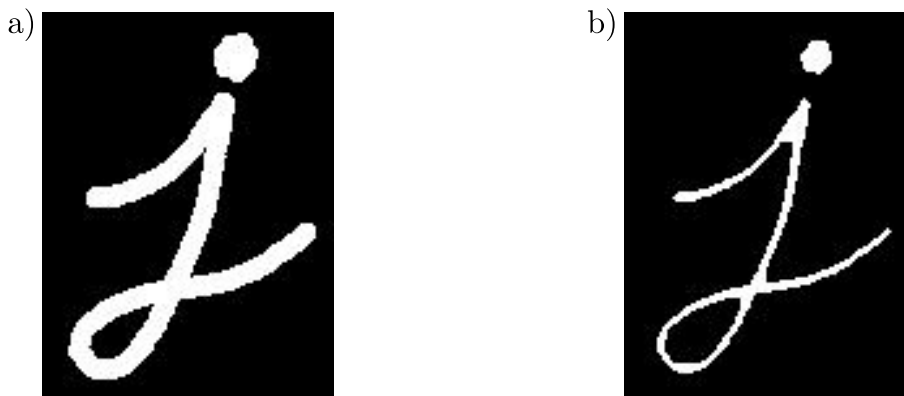


Obrázek 1.4: a) Původní obrázek. b) Obrázek po operaci dilatace. Převzato z [28].

Je tedy vidět, že dilatace roztahuje světlé objekty na obrázku. Velice podobně bychom definovali i operaci eroze značenou  $\ominus$ , jako

$$(f \ominus b)(x, y) = \min_{(k,l) \in B} (f(x+k, y+l)). \quad (1.19)$$

Vidíme, že operace eroze funguje zcela identicky jako dilatace s tím rozdílem, že bereme místo maxima minimum.[29] Můžeme říci, že operace eroze zužuje světlé objekty na obrázku. Ukázku prezentujeme na obrázku 1.5.



Obrázek 1.5: a) Původní obrázek. b) Obrázek po operaci eroze. Převzato z [28].

Obě tyto operace se hojně využívají v nejrůznějších metodách zpracování obrazu. Často se aplikují na obraz obě metody (dilatace a eroze) po sobě tak, aby v ideálním případě zůstal obraz nepozměněn, ale snížil se jeho šum.[30]

### 1.2.4. Metoda lokálních maxim

Všechny úkoly, které řešíme ve spojitosti s analýzou obrazu, se odvíjejí od lokalizace galliových kuliček. Lze proto předpokládat, že výsledný obraz kuličky bude mít vždy lokální maximum a to v jejím středu.[31]

Můžeme tak hledat lokální maxima v obraze a s jejich pomocí lokalizovat kuličky. V praxi se ovšem setkáváme se zašuměným obrazem, a proto se často stává, že maximum existuje na dané nanostruktuře vícero. Proto je potřeba tento problém řešit pokročilejším algoritmem, který dokáže například hledat maxima s určitou vzájemnou minimální vzdáleností od sebe.

Postupů, jak toho docílit, je více. My zde rozebíráme implementaci funkce *peak\_local\_maxima* z knihovny *scikit-image*. Funkce nejdříve aplikuje operaci dilatace na původní obrázek a poté porovnává nově vzniklý obrázek s původním. Místa, kde se hodnota pixelu na novém obrázku shoduje s hodnotou pixelu ze starého obrázku, pak funkce prohlásí za lokální maximum. [32]

Metody hledání lokálních maxim jsou často jednoduché na implementaci a bývají i výpočetně rychlé. Ovšem pro náš úkol nejsou až tak přínosné, jelikož s nimi dokážeme gallyové kuličky pouze lokalizovat. Už nám ale nic neřeknou o velikosti kuličky. Metody slouží v našem případě tedy spíše ke kontrole dat o počtu nalezených kuliček.

### 1.2.5. Houghova kruhová transformace

Houghova kruhová transformace je velice silná transformace umožňující hledat teoreticky libovolné útvary, které lze parametrizovat pomocí zápisu

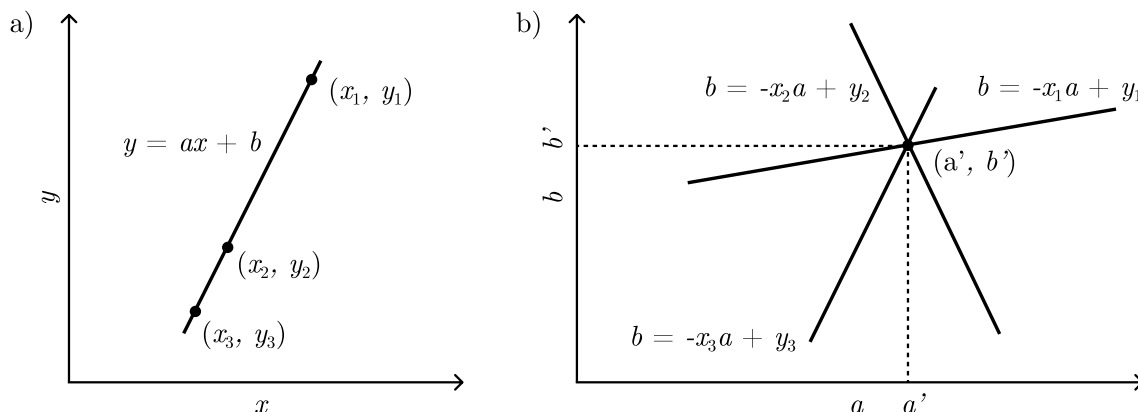
$$f(\vec{v}, \vec{c}) = 0, \quad (1.20)$$

kde  $\vec{v}$  značí vektor všech prostorových souřadnic a  $\vec{c}$  značí vektor konstantních parametrů. [25] V našem případě pak řešíme rovnici

$$(x - c_x)^2 + (y - c_y)^2 = c_r^2, \quad (1.21)$$

kde  $c_x$  a  $c_y$  a  $c_r$  značí pořadě souřadnici středu kružnice podél osy  $x$ , souřadnici středu kružnice podél osy  $y$  a poloměr kružnice.

Vysvětlení Houghovy kruhové transformace provedeme na jednodušším příkladu hledání přímků Houghovou klasickou transformací. Úkol hledání kružnic je pak už velice podobný. [33] Hledáme tedy v obraze všechny přímky s obecným předpisem  $y = ax + b$ , kde  $a$  a  $b$  jsou parametry (viz obrázek 1.6).



Obrázek 1.6: a) Tři body v prostoru  $\{x, y\}$  popisující přímku  $y = ax + b$ . b)  $\{a, b\}$  prostor s bodem  $(a', b')$ , charakterizujícím přímku  $y = a'x + b'$  v prostoru  $\{x, y\}$ .

Rovnici přímky  $y = ax + b$  můžeme přepsat na

$$b = -xa + y. \quad (1.22)$$

Z toho plyne, že přímce  $y = ax + b$  v prostoru  $\{x, y\}$  odpovídá dvojice parametrů  $(a, b)$ . Můžeme proto vytvořit nový prostor  $\{a, b\}$  popsán souřadnicemi  $a$  a  $b$ , ve kterém každý jeho

## 1.2. METODY ZPRACOVÁNÍ OBRAZU

bod bude ekvivalentní jedné přímce z prostoru  $\{x, y\}$ . Pokud se nyní zaměříme na rovnici 1.22, vidíme, že se jedná o klasickou rovnici přímky v prostoru  $\{a, b\}$ .

Při pohledu na konkrétní bod  $(a', b')$  můžeme vidět, že jím prochází právě tolik přímek, kolik bodů se nachází na přímce  $y = a'x + b'$  v  $\{x, y\}$  prostoru. Bod  $(a', b')$  je pro nás proto velmi důležitý, neboť vystihuje hledanou přímku  $y = a'x + b'$ .

Konkrétně algoritmus hledání přímek lze provádět tak, že pro každý světlý bod v obraze (prostor  $\{x, y\}$ ) uvážíme všechny možné kombinace přímek jím procházejících. To znamená, že v prostoru  $\{a, b\}$  nám vznikne přímka. Takto v původním obraze iterujeme přes všechny světlé body. To vyústí v mnoho zaznamenaných přímek v prostoru  $\{a, b\}$  a právě tam, kde budou mít přímky největší četnost průsečíků, se nachází kombinace parametrů  $(a', b')$ , která popisuje hledanou přímku v obraze (prostoru  $\{x, y\}$ ).

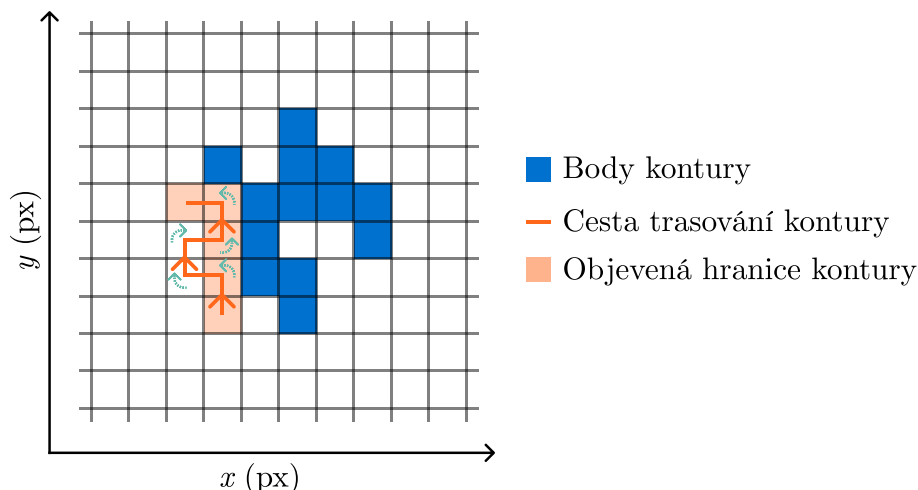
Problém s vyjádřením přímky ve tvaru  $y = ax + b$  je ten, že jím nelze popsat přímky typu  $x = \text{konst.}$  Proto se zavádějí jiná parametrická vyjádření přímek a to například  $r = x \cos \phi + y \sin \phi$ . Výpočet pak probíhá naprosto identicky.[25]

Popsali jsme zde metodu pro hledání přímek, jelikož je jednodušší na pochopení a hlavně obsahuje 2 parametry, Je proto názorná ve 2D obrázcích. Oproti tomu hledání kružnic je již problém o třech parametrech, a proto by se těžko ukazoval. Obecná metoda hledání kružnic zůstává i přesto identická.

### 1.2.6. Hledání kontur

V kontextu analýzy obrazu je konturami myšlena množina bodů, které mají podobnou barvu nebo intenzitu a tvoří hranici nějakého objektu. Dle kontury můžeme pro daný objekt získat jeho lokalizaci a také další informace o jeho poloze, rozloze a podobně.[34] Hledání kontur téměř vždy vyžaduje, aby byly hodnoty v obraze prahovány z toho důvodu, aby bylo možné odlišit pozadí a objekty v popředí. Ideálním případem je binární prahování zmíněné v 1.2.1, jelikož je poté hledání kontur velice jednoduché. Uvažujeme proto obecně při hledání kontur, že objekty, které nás zajímají, jsou v popředí a mají světlé pixely, zatímco pozadí reprezentované tmavými pixely je mimo náš obor zájmu.[35]

Postupů, jak kontury hledat, je mnoho. Mezi nejjednodušší patří například *square tracking* algoritmus, který je založen na jednoduché myšlence reprezentované na obrázku 1.7. Postup je takový, že začneme na libovolném pixelu kontury a označíme ho, jelikož patří do kontury. Poté se posouváme na další pixely dle následujícího pravidla: pokud jsme byli na pixelu popředí, otočíme se vlevo, pokud jsme byli na pixelu pozadí, otáčíme se vpravo. Přitom pokaždé, když se nacházíme na popředí, zaznameneáme pixel jako konturu.[36]

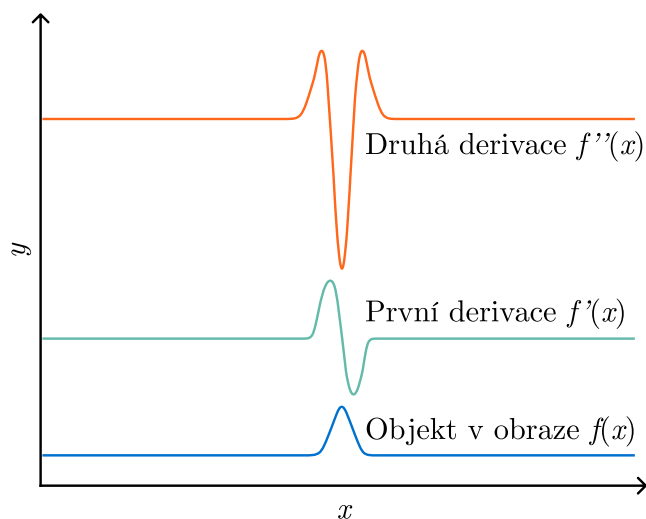


Obrázek 1.7: Ukázka kontury a její detekce pomocí *square tracing* algoritmu.

V praxi je ale možné mít kontury s řádově složitější stavbou, kde je kontura například děravá a nebo i obsahuje více podřadných kontur. V takovém případě je potřeba řešit problém složitějšími postupy, jako je například algoritmus prezentovaný v práci [37]. Ten dokáže kontury nejen detekovat, ale i je řadit do hierarchie. Dokáže tedy například klasifikovat konturu obsaženou v nadřazené kontuře a tento vztah obecně popsat. Tento algoritmus, který v práci využíváme na mnoha místech, implementuje knihovna *OpenCV* pomocí funkce *findContours*.

### 1.2.7. Laplacián gaussianu (*LoG*)

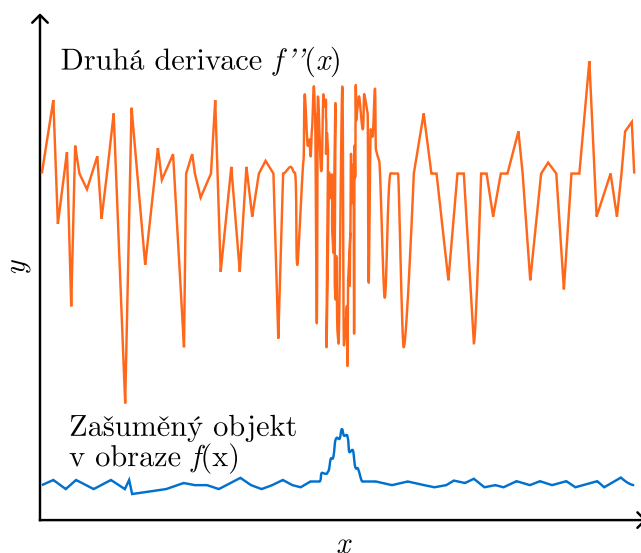
Pro detekci objektů v obraze, které jsou reprezentovány odlišným jasnem od zbytku obrazu, by teoreticky stačilo pracovat s jeho derivacemi. Při pohledu na obrázek 1.8 je patrné, že první derivace  $f'(x)$  nám dle očekávání dává informaci o rychlosti změny původní funkce  $f(x)$ . Druhá derivace  $f''(x)$  nám dá pak nejdůležitější informaci pro takzvaný *zero crossing point* (neboli bod, kde se mění znaménko funkce) první derivace  $f'(x)$ . Tento bod je pro nás velice důležitý, jelikož v něm druhá derivace  $f''(x)$  nabývá svého maxima. [38]



Obrázek 1.8: Působení derivací na funkci  $f(x)$  reprezentující objekt v obraze.

## 1.2. METODY ZPRACOVÁNÍ OBRAZU

Teoreticky by se dalo říci, že je tento postup zbytečný, jelikož jsme z jednoho maxima v obraze (reprezentovaného funkcí  $f(x)$ ) udělali nové maximum na stejné pozici reprezentované druhou derivací  $f''(x)$ . Jsou ale dva důvody, proč tomu tak není. Prvním z nich je, že druhá derivace maximum umocnila co do absolutní číselné hodnoty funkce. Druhým je, že praxe je často jiná a pracujeme s daty, které trpí šumem, nejsou přesně popsateľné jednoduchými funkcemi a podobně. Z toho důvodu přichází do hry ještě gaussova funkce. Jak můžeme vidět na obrázku 1.9 při vygenerování o něco realističtějších dat, pracujeme s dosti zašuměnou funkcí  $f(x)$ . Pokud bychom nyní aplikovali první nebo i druhou derivaci, nedostaneme žádnou zajímavou informaci.



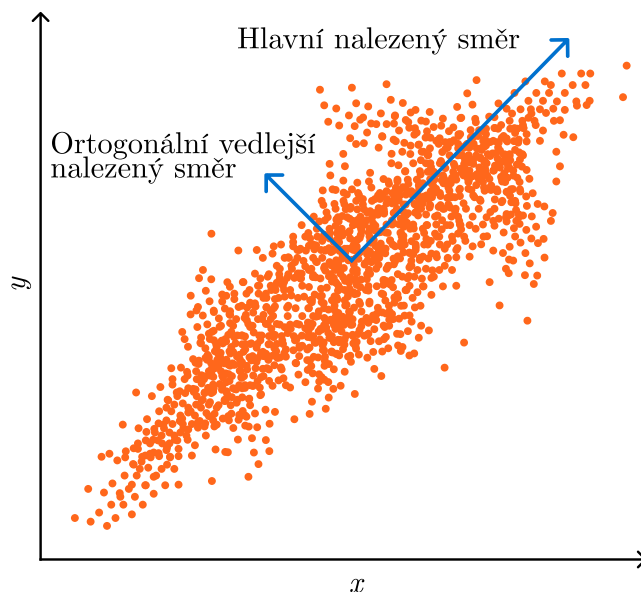
Obrázek 1.9: Působení derivací na funkci reprezentující objekt v obraze.

To lze vyřešit tak, že prvně obraz dostatečně vyhladíme pomocí konvoluce s gaussovským jádrem a na tento obraz následně aplikujeme laplacián (neboli druhou derivaci dle polohy). Tím pak dostáváme ve středu objektu maximum druhé derivace i přes to, že se jedná o zašuměná data. Abychom navíc nemuseli neustále aplikovat gaussovský filtr na obraz a poté dělat jeho druhou derivaci, lze matematicky převést gaussovu funkci a druhou derivaci do jednoho jediného jádra, se kterým nám pak stačí udělat konvoluci s obrazem. Tomuto se říká *LoG*, neboli *Laplacian of gaussian*. Ve výsledku nám nad takovými daty stačí spustit algoritmus pro hledání lokálních maxim, jelikož po aplikaci *LoG* metody pracujeme s hladkými daty (mělo by být proto zajištěno, že lokální maximum bude téměř vždy existovat pouze jedno). Ten nám řekne pozice všech detekovaných objektů.[25] Tento postup ovšem trpí ještě jedním problémem. Při konvoluci obrazu s gaussovským jádrem spoléháme na to, že je velikost (co do šířky) gaussovy funkce (typicky vyjádřeno číslem  $\sigma$ ) podobná velikosti funkce reprezentující objekt v obraze  $f(x)$ . Pokud bychom měli například objekt řádově širší než gaussovu funkci, nebude mít pak konvoluce téměř žádný efekt a stejně tak uplatnění laplaciánu. Nedostali bychom tedy zdaleka tak silný signál, který by indikoval detekci objektu. Proto se v praxi typicky zadává rozsah čísla  $\sigma$ , se kterým chceme konvoluci gaussovským jádrem provádět. Algoritmus pak provede výše zmíněný postup pro všechny  $\sigma$  z daného intervalu a je schopen zachytit veškeré objekty. Toto je také jeden z hlavních důvodů výpočetní a časové náročnosti tohoto algoritmu.[38]

### 1.2.8. Analýza hlavních komponent

Analýza hlavních komponent (*Principal Component Analysis - PCA*) se obecně užívá k redukci dimenze určité množiny statistických dat. Touto redukcí získáme právě ty dimenze (jinými slovy parametry), na kterých je množina dat nejvíce závislá. Jedná se proto o velmi abstraktní metodu, která lze využít pro mnoho statistických aplikací.[39]

U analýzy obrazu využíváme tuto metodu pro určení orientace detekovaného objektu. Tím, že je objekt v obraze reprezentován nějakou množinou bodů, můžeme na této množině pomocí *PCA* určit směry, ve kterých mají body co největší rozptyl (tedy směry, na kterých poloha množiny bodů nejvíce závisí). To je ekvivalent natočení daného objektu. Více názorně situaci popisuje následující obrázek 1.10.



Obrázek 1.10: Nalezení hlavních komponent množiny datových bodů pomocí *PCA* metody.

Matematické pozadí analýzy hlavních komponent je složité. Hlavně pro obecný případ o  $n$  dimenzích. Pokusíme se zde spíše jen nastínit postup této metody a to pouze pro jednodušší 2D případ s předpokladem nulového aritmetického průměru dat (tento předpoklad klademe pouze pro zjednodušení výpočtů).

V případě, že máme množinu dat, můžeme tato data zapsat do jedné jediné matice. Uvažme pro příklad body v 2D prostoru. Můžeme pak jejich souřadnice zapsat jako dva vektory  $\vec{x}$  a  $\vec{y}$ , kde  $x_i$ , respektive  $y_i$  je  $x$ , respektive  $y$  souřadnice  $i$ -tého bodu. Z těchto dvou vektorů tvořící množinu 2D dat můžeme vytvořit matici všech dat  $\mathbf{X}$  následovně

$$\mathbf{X} = \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix}. \quad (1.23)$$

Pro matici  $\mathbf{X}$  můžeme pak získat její kovarianční matici, která má podobu

$$\mathbf{C}_{\mathbf{X}} = \begin{pmatrix} \text{var}(\vec{x}) & \text{cov}(\vec{x}, \vec{y}) \\ \text{cov}(\vec{y}, \vec{x}) & \text{var}(\vec{y}) \end{pmatrix} \propto \mathbf{X}\mathbf{X}^T = \tilde{\mathbf{C}}_{\mathbf{X}}, \quad (1.24)$$

kde  $\text{var}()$  značí rozptyl (varianci) a  $\text{cov}()$  značí kovarianci dat. Bylo by nyní hezké mít možnost transformovat takovou kovarianční matici na jinou kovarianční matici, která bude diagonální. Všechny mimodiagonální prvky (které reprezentují kovariance) budou tedy

### 1.3. DIFUZE

nulové a bude to znamenat, že jsme našli souřadnou soustavu, ve které mezi souřadnicemi dat neexistuje žádný statistický vztah a jsou na sobě nezávislé. Tím pádem budou rozptyly (variance) samostatných souřadnic největší.[40] Hledáme tedy matici transformovaných dat  $\mathbf{Y} = \mathbf{S}\mathbf{X}$ , pro níž platí, že její kovarianční matice  $\mathbf{C}_\mathbf{Y}$  je diagonální.

Víme, že matice  $\tilde{\mathbf{C}}_\mathbf{X}$  lze rozložit pomocí vlastní dekompozice na  $\mathbf{Q}\Lambda\mathbf{Q}^{-1}$  [41], kde matici  $\mathbf{Q}$  tvoří vlastní vektory matice  $\tilde{\mathbf{C}}_\mathbf{X}$  a matice  $\Lambda$  je diagonální matice obsahující vlastní hodnoty matice  $\tilde{\mathbf{C}}_\mathbf{X}$ . Mimo to lze zjistit, že díky symetričnosti matice  $\tilde{\mathbf{C}}_\mathbf{X}$  platí  $\mathbf{Q}^T = \mathbf{Q}^{-1}$ . [40] Pokud nyní za hledanou transformační matici  $\mathbf{S}$  prohlásíme matici  $\mathbf{Q}^T$ , dostáváme následující

$$\begin{aligned}\mathbf{Y} &= \mathbf{S}\mathbf{X} \\ \mathbf{Y} &= \mathbf{Q}^T\mathbf{X} \\ \mathbf{C}_\mathbf{Y} &\propto \mathbf{Y}\mathbf{Y}^T \\ \mathbf{C}_\mathbf{Y} &\propto \mathbf{Q}^T\mathbf{X}(\mathbf{Q}^T\mathbf{X})^T \\ \mathbf{C}_\mathbf{Y} &\propto \mathbf{Q}^T\mathbf{X}\mathbf{X}^T\mathbf{Q} \\ \mathbf{C}_\mathbf{Y} &\propto \mathbf{Q}^T\mathbf{Q}\Lambda\mathbf{Q}^T\mathbf{Q} \\ \mathbf{C}_\mathbf{Y} &\propto \Lambda.\end{aligned}$$

Můžeme si všimnout, že poslední rovnice nám říká, že kovarianční matice  $\mathbf{C}_\mathbf{Y}$  je úměrná diagonální matici  $\Lambda$ . Našli jsme tedy žádanou transformaci.[42] Tato transformace obsahuje pro nás důležité informace právě v transformační matici  $\mathbf{Q}^T$ . Říká nám, jak máme původní souřadnicový systém dat rotovat, abychom se dostali do nového souřadnicového systému s nulovými kovariancemi a největšími rozptyly (variancemi). Jinak řečeno dává nám (vlastní) vektory reprezentující tyto směry, které jsou ekvivalentem směru natočení objektu. Tímto způsobem jsme schopni zjistit natočení objektu v prostoru. Kromě toho se také hodí vlastní hodnoty těchto vektorů skryté v matici  $\Lambda$ . Ty nám říkají, v jakém poměru jsou zastoupeny velikosti vlastních vektorů, neboli v jakém poměru jsou rozptyly (variance) dat vzájemně vůči sobě. To lze v praxi interpretovat jako nesymetričnost dat ve směrech vlastních vektorů.

## 1.3. Difuze

Úlohu, ve které řešíme šíření náboje na *hallbaru*, můžeme z fyzikálního hlediska popsat difuzí.

### 1.3.1. Fickovy zákony

Pro popis difuze slouží dva Fickovy zákony. První z nich hovoří o toku  $\vec{j}$  v závislosti na gradientu veličiny  $c$

$$\vec{j} = -D\vec{\nabla}c, \quad (1.25)$$

kde  $D$  je difuzní koeficient.[43]. Druhý z nich (též zvaný rovnice kontinuity) udává vztah mezi časovou parciální derivací veličiny  $c$  a její divergencí



$$\frac{\partial c}{\partial t} + \vec{\nabla} \cdot \vec{j} = 0. \quad (1.26)$$

Po sloučení obou zákonů, dostáváme obecnou podobu difuzní rovnice

$$\frac{\partial c}{\partial t} = \vec{\nabla} \cdot (D \vec{\nabla} c). \quad (1.27)$$

### 1.3.2. Difuzní rovnice ve 2D

Pro naše účely, kde pracujeme s velmi tenkými vrstvami (i monovrstvami) můžeme difuzní rovnici nadále upravit tak, aby platila pouze pro dvou dimenzionální homogenní struktury a tím pádem popisovala fyziku probíhající na povrchu látek. Zajímá nás šíření náboje, které jsme v experimentu schopni detekovat nepřímo prostřednictvím měření povrchového potenciálu  $\varphi_{\text{povrch}}$ . Ten měříme s pomocí Kelvinovy sondové mikroskopie (*Kelvin probe force microscopy - KPFM*). Proto uvádíme rovnici již ve tvaru, který popisuje šíření povrchového potenciálu  $\varphi_{\text{povrch}}$  závislého na prostoru a čase  $\varphi_{\text{povrch}} = \varphi_{\text{povrch}}(x, y, t)$  s předpokladem konstantního povrchového difuzního koeficientu  $D_{\text{povrch}}$ .

$$\frac{\partial \varphi_{\text{povrch}}}{\partial t} = D_{\text{povrch}} \left( \frac{\partial^2 \varphi_{\text{povrch}}}{\partial x^2} + \frac{\partial^2 \varphi_{\text{povrch}}}{\partial y^2} \right). \quad (1.28)$$

### 1.3.3. Numerická řešení difuzní rovnice

V oblasti řešení parciálních diferenciálních rovnic (mezi které difuzní rovnice spadá) existují tři hlavní přístupy, jak je řešit - metoda konečných diferencí, metoda konečných prvků a metoda konečných objemů. Všechny mají společné to, že jsou definovány na nějaké síti, na které daný problém řešíme.

První a nejvíce intuitivní metodou je metoda konečných diferencí. Tato metoda pracuje s hodnotami zkoumaných funkcí na koncích jednotlivých intervalů sítě. Typicky pak na těchto intervalech aproximuje derivace funkcí podle

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (1.29)$$

To pak vede na soustavu rovnic, které metoda řeší například pomocí algebraických operací s maticemi.

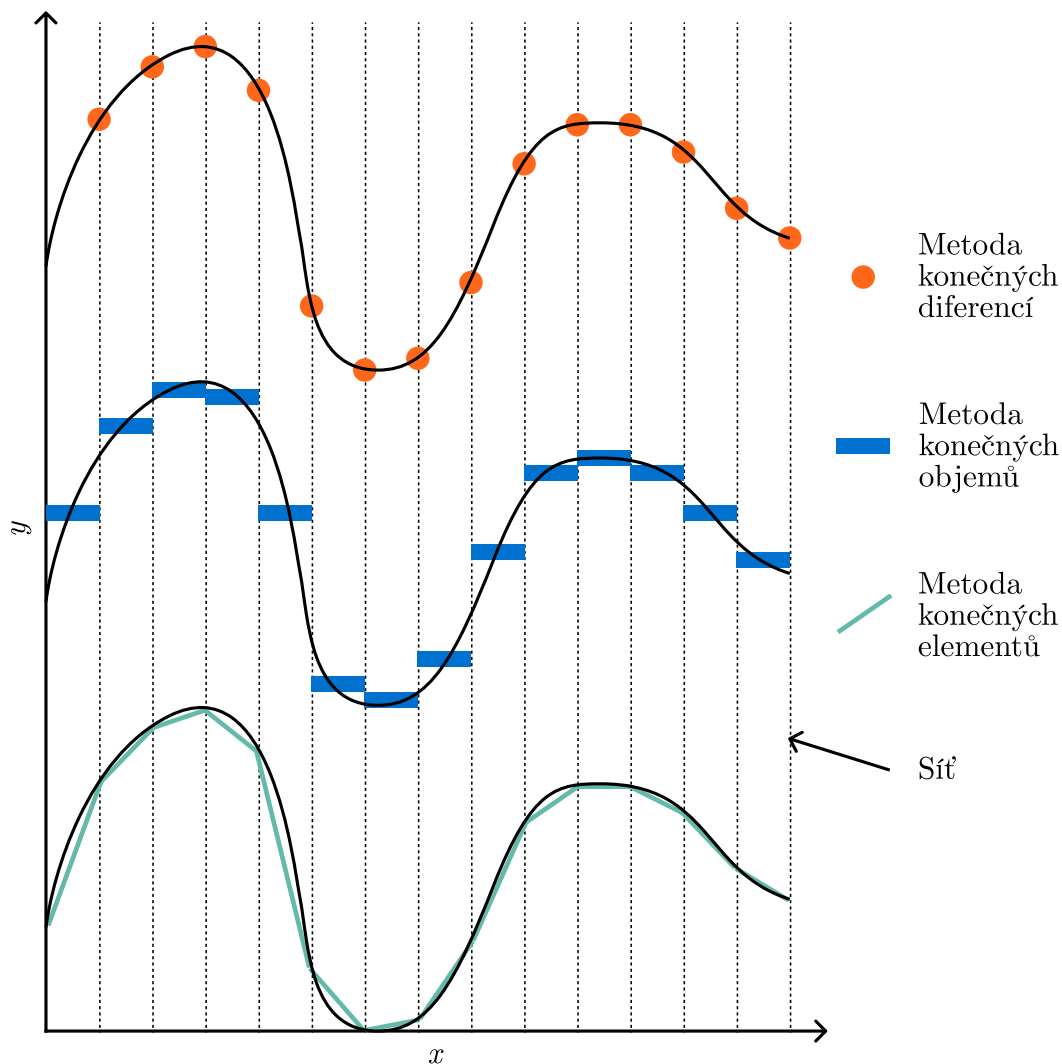
Další metoda - metoda konečných objemů - se na rozdíl od metody konečných diferencí nedívá na hodnoty funkce na koncích intervalů sítě. Místo toho se dívá na průběh funkce na každém intervalu sítě a vyhodnocuje integrál (aritmetický průměr) této funkce. V určitém slova smyslu lze proto prohlásit, že nepracuje s funkcemi samotnými, ale s jejich toky. Stejně tak je metoda odlišná v tom, že se nedívá na hodnotu pro jeden fixní bod na kraji intervalu sítě, ale spíše na hodnotu, která specifikuje celý interval sítě. Tato metoda tedy opět zjednodušuje výpočty díky diskretizaci prostoru, ovšem samotný problém řeší pomocí převedení veškerých veličin na jejich integrální podoby a pracuje nadále s těmito čísly. Je zajímavé si povšimnout vlastnosti konzervativnosti této metody. U metody konečných diferencí máme o funkci představu pouze z množiny omezených bodů. Pokud

### 1.3. DIFUZE

dojde ke zvolení špatné (řídce) sítě, můžeme funkci aproximovat velice špatně. Zatímco u metody konečných objemů vystihujeme integrál celé funkce nezkresleně menšími příspěvky integrálů na všech intervalech sítě. Nepochází tedy k tak velké ztrátě informace o funkci.

Obě již zmíněné metody fungovaly na základě aproximace funkce nějakým způsobem ať už v rámci celého intervalu sítě, nebo pouze na konci intervalů sítě. Metoda konečných elementů oproti tomu volí přístup takový, že se snaží danou funkci na intervalu sítě aproximovat jinou funkcí. Záleží proto na uživateli, jaký řád aproximace si zvolí (lineární, kvadratický a tak dále).<sup>[44]</sup>

Srovnání všech metod ukazujeme pro zjednodušený 1D případ na následujícím obrázku 1.11.



Obrázek 1.11: Porovnání metod konečných diferencí, objemů a elementů a jejich přístupu k diskretizaci zkoumaných funkcí.

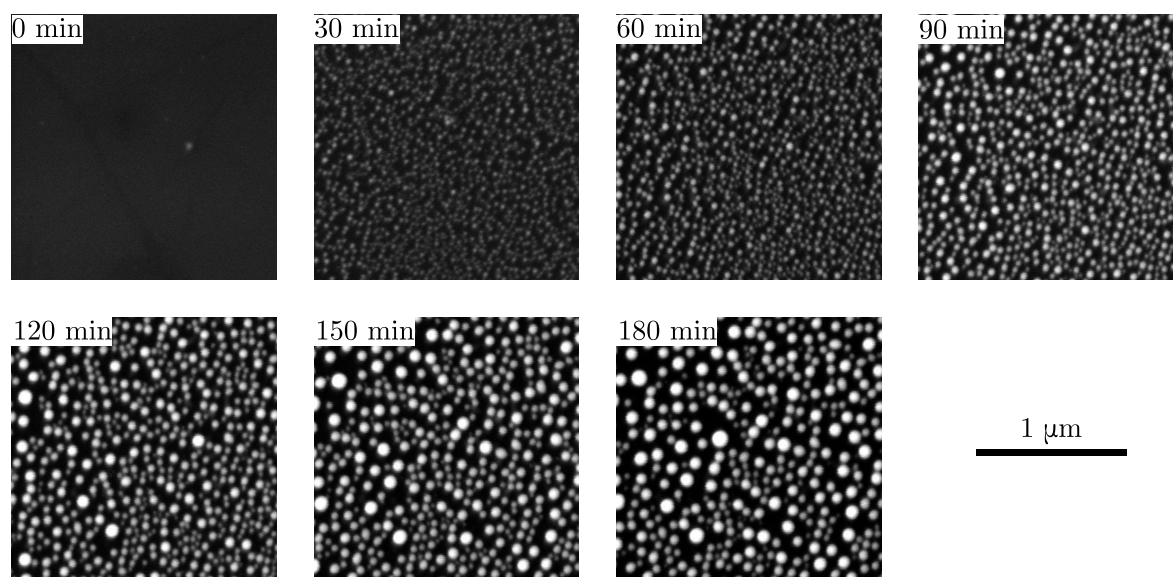
## 2. Praktická část

V praktické části vyhodnocujeme výsledky všech tří úloh postupně. Úkolem k první úloze bylo zpracovat obrazy z rastrovacího elektronového mikroskopu (*Scanning electron microscopy - SEM*), které zaznamenávaly galliové kuličky nadeponované na grafenu. Druhý úkol spočíval v simulaci šíření náboje a časového vývoje potenciálu na struktuře zvané *hallbar*. Poslední úkol se podobal úkolu prvnímu, kde jsme analyzovali snímky nitridu gallia na grafenu. Dodatečný úkol kromě lokalizace a charakterizace zde spočíval i v analýze orientace částic, jelikož částice nejsou symetrické.

### 2.1. Charakterizace galliových kuliček na grafenu

#### 2.1.1. Uvedení problému

Experiment za tímto úkolem probíhá deponováním gallia na grafen v časovém intervalu zhruba 180 minut. Během tohoto časového intervalu je pořízeno několik snímků po 5 minutách z rastrovacího elektronového mikroskopu. Pracujeme proto s obrázky, které obsahují galliové kuličky různých velikostí na grafenu. Veškeré snímky z této části jsou převzaty od Ing. Jakuba Piastka. Pro přiblížení problematiky ukazujeme časový vývoj celého experimentu na následujícím obrázku 2.1.



Obrázek 2.1: Znázornění experimentu deponice gallia na grafen. Intervaly mezi snímky jsou 30 minut. Deponováno za teploty 300° C.

#### 2.1.2. Požadavky na program

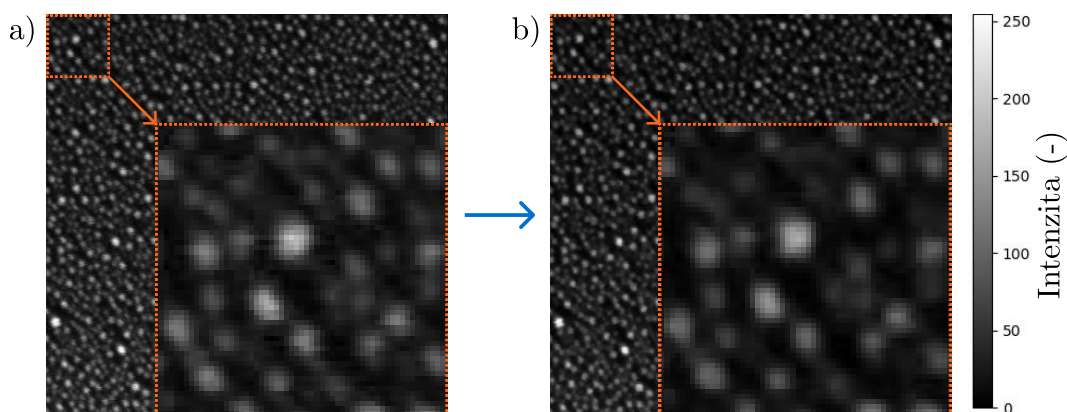
Program by měl zvládat lokalizovat kuličky, stejně jako zjistit jejich velikost a jejich celkový počet. Dále má program za úkol vyhledat nejbližšího souseda každé kuličky. Poslední úlohou je pak zjistit celkové pokrytí grafenu galliovými kuličkami. Zároveň parametry chceme analyzovat pro každý snímek, což nám poskytne časově závislou statistiku všech parametrů.

## 2.1. CHARAKTERIZACE GALLIOVÝCH KULIČEK NA GRAFENU

### 2.1.3. Příprava obrazu

Mnoho snímků často obsahuje body na které se ostří, zaměřovací kříže a nebo i nečistotu na vzorku, která by se do analýzy obrazu neměla dostat. Všechny snímky proto ořezáváme tak, abychom se zbavili všech nechtěných artefaktů.

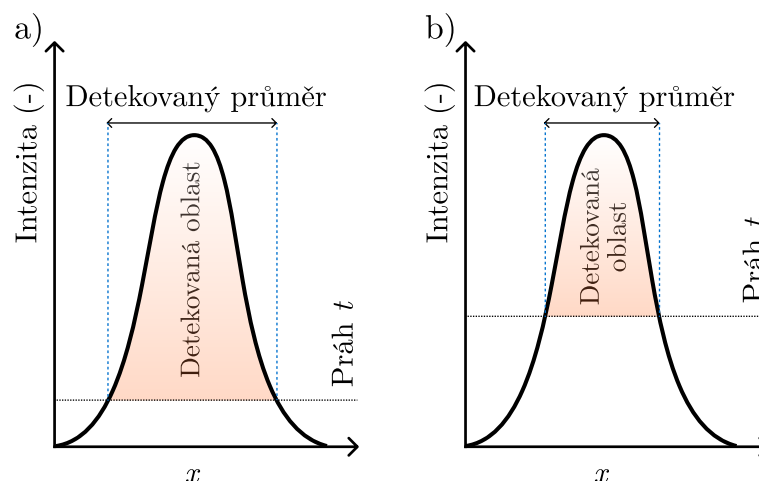
Dalším krokem je odstranění šumu obrazu. První možností, jak se zbavit šumu v obrazu, je konvoluce obrázku s gaussovským jádrem. Toto efektivně rozmaže obrázek a tím pádem i zmírní šum. Problém tohoto postupu ovšem byl, že nikdy nevedl k lepším výsledkům při aplikaci dalších algoritmů, které hledaly kuličky. Obecně se ukázalo, že i menší rozmazání obrázku vede ke špatným výsledkům. Proto jsme tento postup zavrhlí a dále ho v této práci neuplatňujeme. Oproti tomu druhý postup uplatňující dilataci a erozi (viz kapitola 1.2.3) na obraz ihned po sobě prokázal lepší výsledky. To ukážeme na obrázku 2.2.



Obrázek 2.2: Ukázka efektu aplikování dilatace a eroze po sobě na obraz. a) Detail originálního obrázku z mikroskopu. b) Detail upraveného obrázku s řádově sníženým šumem bez výrazného snížení intenzity kuliček.

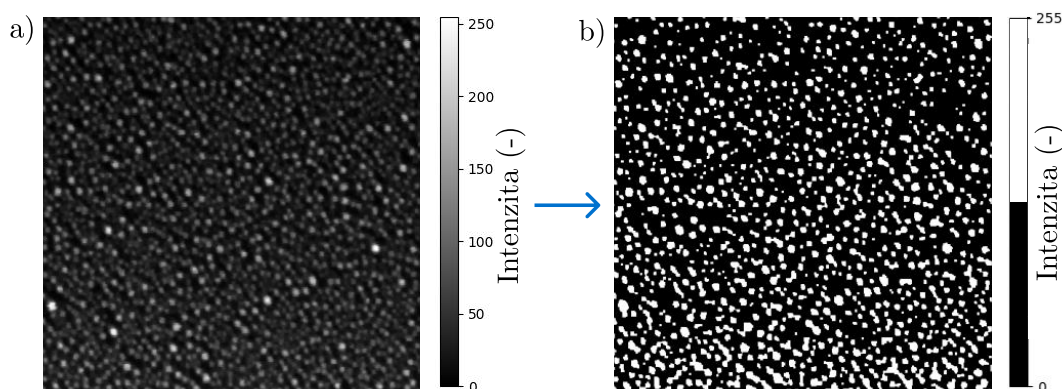
Při následné operaci prahování přicházelo v úvahu nastavení globálního binárního prahování popsaného v kapitole 1.2.1 se stejnou hodnotou prahu pro všechny obrázky v časové sérii. Problém této metody spočívá v tom, že každý pořízený snímek může mít zcela jinou expozici, čímž je tato metoda nepoužitelná. Tento problém jsme zkusili adresovat algoritmem pro vyrovnaní expozice všech obrázků pomocí ekvalizace histogramu. Byla použita funkce `equalize_hist` z knihovny `scikit-image`. Problém této metody ale spočíval v tom, že změna expozice často vedla i k vizuální změně velikosti galliových kuliček. Algoritmy tedy poté vyhodnocovaly kuličky špatně, jelikož se jevíly často větší. Proto tuto metodu nadále v práci neuvažujeme.

Jako nejefektivnější řešení se ukázalo globální binární prahování pomocí Otsuovy metody (kapitola 1.2.2). Užití této metody ale přináší úskalí v tom, že je každý obrázek prahován jinou hodnotou. To může vést k vytvoření chyby při určování velikosti kuliček, neboť vždy interpretujeme obraz s jinou prahovou hodnotou. Situaci znázorňujeme na následujícím obrázku 2.3. Nedochozí ovšem k tak velké chybě jako při použití ekvalizace histogramu.



Obrázek 2.3: Ukázka chybného vyhodnocení velikosti profilů dvou identických kuliček kvůli rozdílnému prahování. a) Nižší hodnota prahu vede na klasifikování většího rozměru kuličky. b) Vyšší hodnota prahování vede na klasifikování menšího rozměru kuličky.

Aplikace Otsuovy metody na obrázek pak vypadá jako na obrázku 2.4.

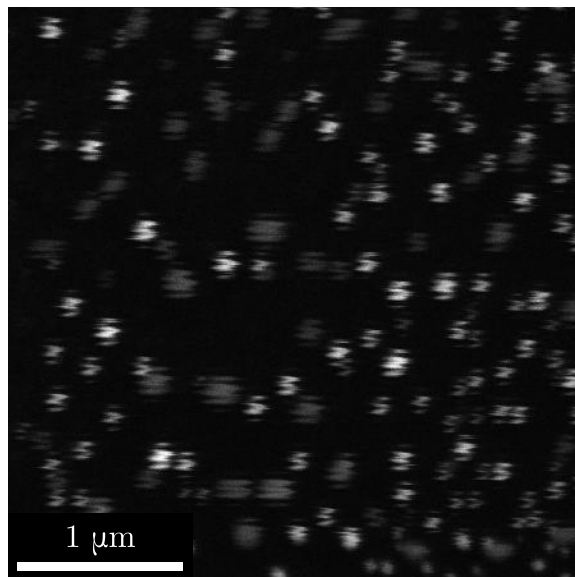


Obrázek 2.4: Prahování obrazu pomocí Otsuovy metody. a) Obrázek vycházející z obrázku 2.2. b) Prahovaný obrázek s binární škálou hodnot připravený k analýze.

#### 2.1.4. Konkrétní kroky analýzy

První analýza proběhla za pomoci použití Houghovy kruhové transformace (kapitola 1.2.5). Primární problém, kterým tato metoda trpěla, byla její neschopnost analyzovat ne zcela perfektní obrázky. Při pohledu na následující obrázek 2.5 si můžeme povšimnout, že se jedná o obrázek galliových kuliček z rastrovacího elektronového mikroskopu, který je ale v důsledku rušících vlivů roztřesený.

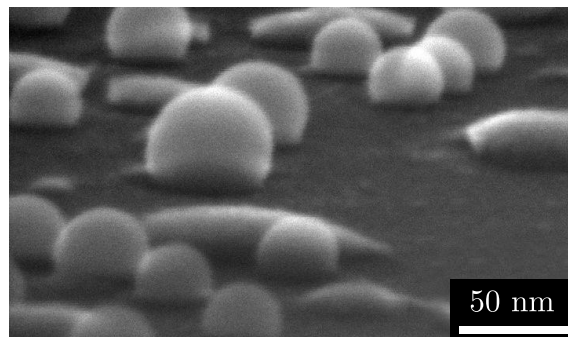
## 2.1. CHARAKTERIZACE GALLIOVÝCH KULIČEK NA GRAFENU



Obrázek 2.5: Detail poukazující na úskalí použití Houghovy kruhové transformace pro data zatížena roztřesením snímku. Houghova metoda si s takovýmito snímky nedokáže příliš poradit.

Tento artefakt kompletně znehodnotil výsledky Houghovy metody a proto tuto metodu nadále již v práci neuvažujeme. Druhý přístup detekce objektů fungoval na principu hledání kontur (kapitola 1.2.6) pomocí funkce *findContours* z knihovny *OpenCV*. Následně jsme proložili kontury kružnicemi pomocí funkce *fitEllipse*. Tím jsme byli schopni zjistit polohy i průměry všech nalezených kuliček. Poslední metoda spočívala v aplikaci funkce *blob\_log* z knihovny *scikit-image*. Tato funkce je přímo určena k hledání útvarů podobných kuličkám v obraze a využívá metody *LoG* popsanou v kapitole 1.8. Pro všechny metody pak bylo možné z množiny detekovaných kuliček dopočítat vzdálenosti mezi nejbližšími sousedy a celkové pokrytí vzorku.

V celé této sekci a i nadále v práci uvažujeme, že je kulička kruhovitěho tvaru při pohledu shora. Občas se může stát, že některé útvary rostly spíše do tvaru elipsy, ale stále bylo možné je nahradit kružnicí. Jako podpůrná informace pro toto tvrzení poslouží i následující obrázek 2.6, na kterém vidíme kuličky z profilu a můžeme prohlásit, že se jedná o útvary ve tvaru kulového vrchlíku, které mají kruhovitý půdorys.



Obrázek 2.6: Pohled na galliové kuličky z boku. Lze pozorovat, že se nejedná o dokonalé kuličky, ovšem půdorysně objekty nabývají kruhovitěho tvaru. Převzato od Ing. Jakuba Piastka.

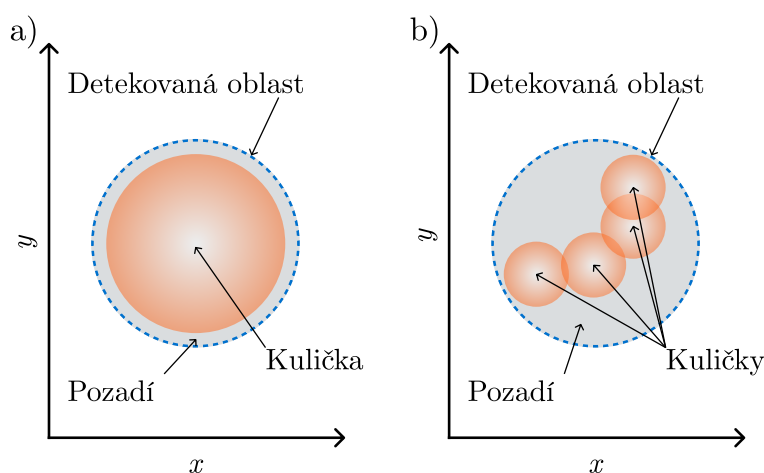


### 2.1.5. Filtrování dat

Může se stát, že se několik kuliček sloučí do sebe například při prahování. To pak může hledací algoritmus chybně vyhodnotit jako jednu větší kuličku a pokazit tím celkovou statistiku. Proto bylo potřeba výsledná data analýzy ještě filtrovat. Zde jsme uplatnili dva postupy.

První spočíval v jednoduché filtraci, která se zbavila všech kuliček s větším průměrem, než pevně nastavená hodnota pro maximální průměr kuličky. Tuto hodnotu jsme vždy zvolili na základě posledního snímku v časové sérii, kde nabývaly kuličky největších rozměrů.

Druhý postup spočíval ve vyhodnocení poměru počtů světlých pixelů (reprezentujících kuličku) a tmavých pixelů (reprezentujících pozadí) v detekované oblasti. Tímto jsme byli schopni odstranit výše zmíněný artefakt splnutí vícero kuliček dohromady. Více situaci popisuje následující obrázek 2.7.



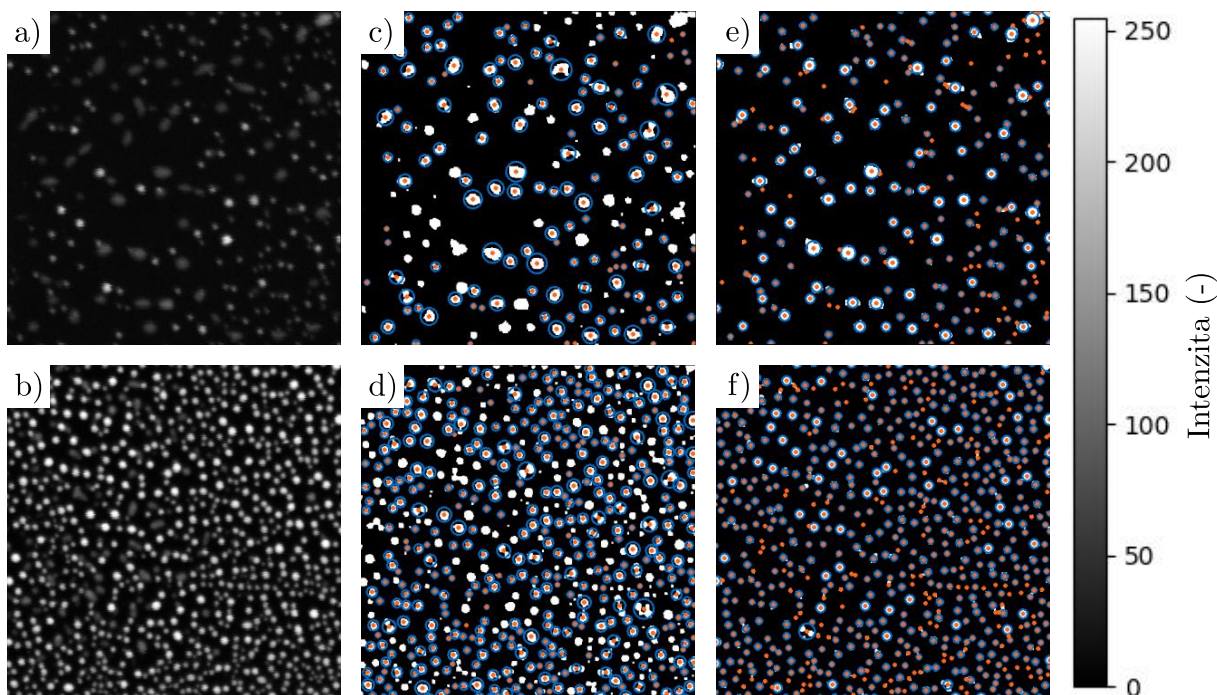
Obrázek 2.7: Znázornění filtrovacího postupu pro filtraci chybně analyzovaných kuliček pomocí porovnávání počtu pixelů v detekované oblasti příslušejících světlým pixelům (kulička) a tmavým pixelům (pozadí). a) Správně detekovaná kulička s většinou světlých pixelů. b) Chybně detekovaná kulička s přibližně stejným počtem světlých i tmavých pixelů.

Nakonec se ukázalo, že pokud má detekovaná (kruhová) oblast 64 % světlých pixelů a zbytek tmavých, je to dobrá hranice pro rozhodnutí mezi tím, zda se jedná o jednu kuličku, nebo o sloučení více kuliček, které pro vyhodnocení již nejsou vhodné.

### 2.1.6. Vyhodnocení výsledků

Jako první ukazujeme výsledek analýzy dvou obrázků pomocí obou metod pro jejich vzájemné porovnání na následujícím obrázku 2.8.

## 2.1. CHARAKTERIZACE GALLIOVÝCH KULIČEK NA GRAFENU



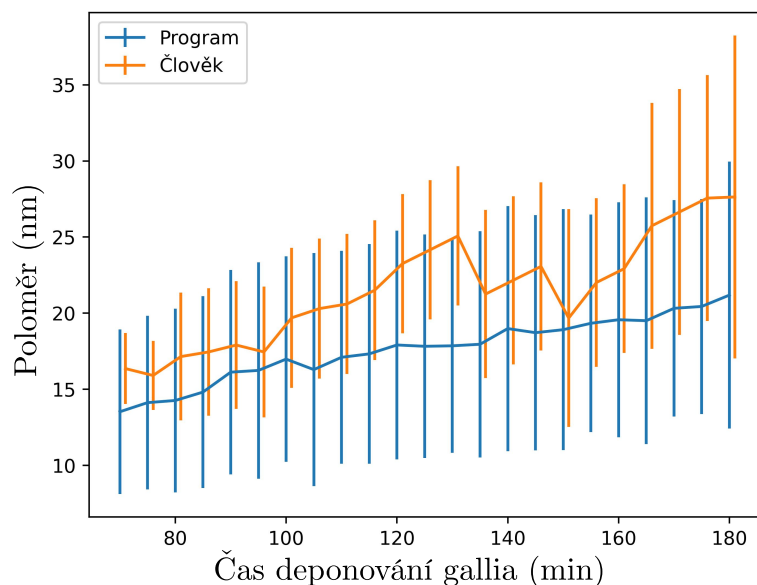
Obrázek 2.8: Porovnání metody hledání kontur a *LoG* pro dvě sady různých obrázků. a), b) Původní obrázky. c), d) Obrázky s nalezenými kuličkami pomocí metody hledání kontur. e), f) Obrázky s nalezenými kuličkami pomocí metody *LoG*. Z obrázků je zřejmé, že metoda *LoG* dosahuje značně lepších výsledků.

Z obrázku 2.8 lze jasně usoudit, že je metoda *LoG* úspěšnější. Pro první obrázek s relativně málo objekty vynechala metoda hledání kontur zhruba 20 % struktur. Na druhém a řadově obsazenějším obrázku vynechala metoda hledání kontur už 42 % struktur. Proto jsme stanovili metodu *LoG* jako nejlepší metodu pro lokalizaci těchto struktur. A to i přes její časovou náročnost, která je zhruba trojnásobek časové náročnosti metody hledání kontur. Konečně pak můžeme vyhodnotit data z experimentu a získat například vývoj průměrného poloměru v čase nebo vývoj jejich distribucí, pokrytí vzorku v čase a podobné závislosti. Proto na závěr této úlohy prezentujeme srovnání vyhodnocení experimentu vyvinutým programem a člověkem. Mimo to ukazujeme několik dalších poznatků z experimentů založených na vyhodnocení dat programem.

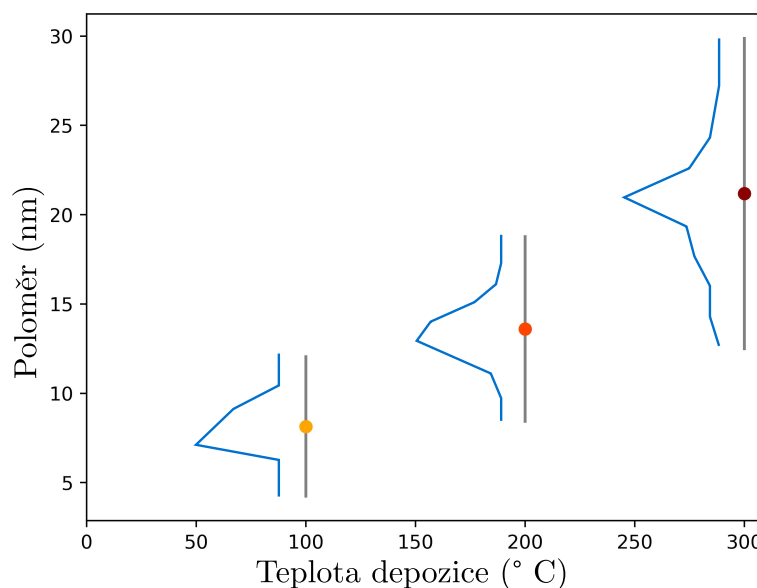
Na obrázku 2.9 můžeme pozorovat vývoj průměrného poloměru galliových kuliček v čase<sup>1</sup>. Data byla vyhodnocena nejprve člověkem a následně i programem pomocí metody *LoG*. Z obou křivek dat lze pozorovat, že průměrný poloměr kuliček s časem roste, což je způsobeno růstem jednotlivých kuliček a jejich následným shlukováním, což je z fyzikálního pohledu očekávané chování. Můžeme také pozorovat, že hodnoty získané programem jsou posunuty oproti hodnotám získaným měřením. To je dle našeho názoru zapříčiněno mírnou změnou velikosti kuliček během prahování. Kromě toho ovšem vypadá rostoucí tendence obou křivek podobně.

<sup>1</sup>O distribucích poloměrů kuliček víme, že by se měly řídit Poissonovým rozdělením, pro které nemá směrodatná odchylka stejný význam jako pro Gaussovo rozdělení. Avšak pro představu o rozptylu dat uvádíme v grafech pro přiblížení směrodatnou odchylku.





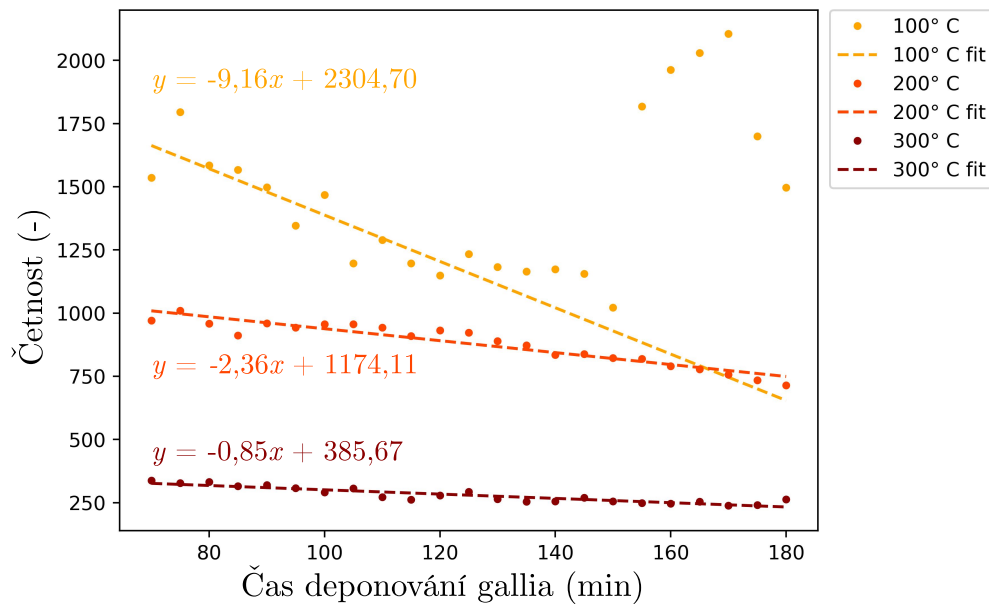
Obrázek 2.9: Porovnání vývoje průměrného poloměru galliových kuliček (pro teplotu 300° C) v čase pozorovaného člověkem a programem. Svislé pruhy značí rozsah směrodatné odchylky. (Oranžové body jsou v grafu mírně horizontálně posunuty kvůli čitelnosti. Ve skutečnosti jsou vyhodnocovány pro identické časy jako modrá křivka)



Obrázek 2.10: Porovnání průměrného finálního poloměru kuliček v posledním stádiu depozice gallia pro různé teploty. Svislé pruhy značí rozsah směrodatné odchylky. Modré křivky značí distribuci poloměrů pro danou teplotu.

Dále na obrázku 2.10 prezentujeme průměrný poloměr v posledním čase depozice gallia pro různé teploty (už pouze vyhodnoceno programem). Můžeme pozorovat, že s rostoucí teplotou depozice roste i průměrná velikost kuliček, což je opět v souladu s očekáváním, neboť pro vyšší teploty dostáváme větší difuzní délky. Na obrázek 2.10 navazuje následující obrázek 2.11, který ukazuje vývoj četností kuliček v čase pro tři výše prezentované teploty.

## 2.2. SIMULACE ŠÍŘENÍ NÁBOJE NA HALLBARU



Obrázek 2.11: Vývoj četností nalezených kuliček v čase pro tři pozorované teploty. Každá z množiny dat je navíc proložena lineárním polynomem. Pouze pro data reprezentující 100° C uvažujeme pro proložení jen první polovinu dat, neboť v druhé polovině dat zjevně došlo k chybnému vyhodnocení.

Opět můžeme pozorovat očekávané chování, že kuliček postupně s časem ubývá, neboť se navzájem slučují. V grafu si můžeme také všimnout, že pro každou teplotu se docela razantně liší absolutní počty kuliček. To opět navazuje na fakt, že pro vyšší teplotu máme větší difuzní délku, takže se kuličky snadněji slučují a proto jich ve výsledku není tolik. Pouze v čase 155 min pro teplotu 100° C vidíme docela značnou výchylku, která značí nějakou chybu. Ta může nastat již na úrovni experimentu, například pohybem vzorku (pohyb vzorku a sledování jednotlivých částic náš program neumí), a nebo například rozmazáním snímku. Chyba může vzniknout ale také na úrovni programu například špatným prahováním. Po čase, kde nastala chyba, můžeme ale pozorovat, že má četnost opět klesající tendenci a proto se domníváme, že se jedná spíše o chybně zachycené snímky, což se i při pohledu na ně samotné potvrzuje.

## 2.2. Simulace šíření náboje na *hallbaru*

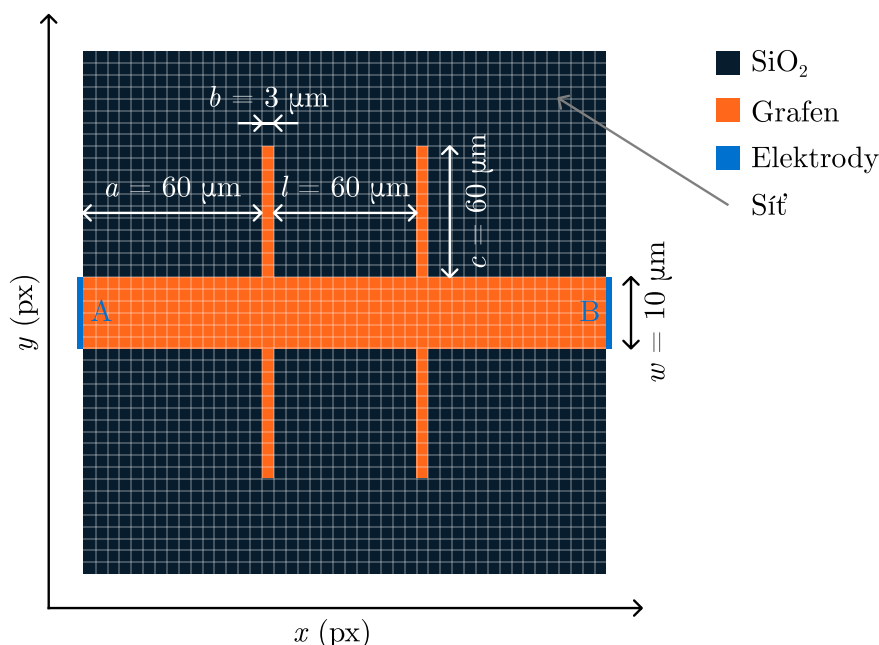
### 2.2.1. Uvedení problému

V druhé části praktické sekce simulujeme šíření náboje v čase na struktuře *hallbaru*, které se projevuje změnou povrchového potenciálu. Typické schéma *hallbaru* užívaného v experimentech můžeme vidět na obrázku 2.12.

### 2.2.2. Realizace výpočtu

Pro řešení difuzní rovnice jsme použili volně dostupnou *FiPy* knihovnu pythonu. Jedná se o knihovnu speciálně určenou k řešení parciálních diferenciálních rovnic. Tato knihovna k výpočtu využívá metodu konečných objemů zmíněnou v kapitole 1.3.3. Celá simulace je navržena pro 2D případ. Řešíme tedy časový vývoj potenciálu v rovinné, ekvidistantní,

čtvercové síti. Na tuto síť aplikujeme pole difuzních koeficientů. Jedná se o 2D pole (o stejném rozměru jako vygenerovaná síť), kde každá hodnota odpovídá jedné buňce v síti (a tuto buňku charakterizuje difuzním koeficientem). Geometrie celého problému tak charakterizujeme tedy pouze 2D polem difuzních koeficientů. Okrajové podmínky jsou zvoleny tak, že uvažujeme nulový tok přes veškeré hranice sítě (neboli Neumannova okrajová podmínka). Stejně tak uvažujeme počáteční podmínky takové, že hodnota potenciálu na celé síti je nulová. Přiložení potenciálu na elektrody *hallbaru* realizujeme hraničními podmínkami, kde na hrany hlavního kanálu *hallbaru* (tedy jeho levou a pravou elektrodu, viz obrázek 2.12) aplikujeme fixní potenciál po celou dobu simulace (dále viz obrázek 2.12). Typické rozměry *hallbaru* pro simulaci jsou znázorněny na následujícím obrázku 2.12. Těmito rozměry jsme se snažili co nejvíce reflektovat reálné rozměry z experimentů.



Obrázek 2.12: Základní geometrie *hallbaru* s typickými rozměry používanými v experimentech i simulaci. Znázorněny jsou i hraniční podmínky pro simulaci elektrod A a B. Z této geometrie následně vycházejí všechny další modifikace *hallbaru*.

### 2.2.3. Požadavky na program

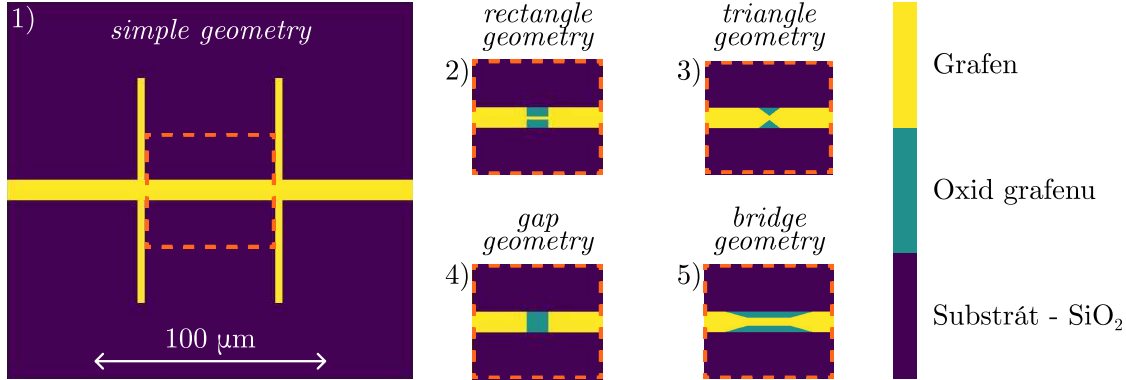
Na obrázku 2.12 můžeme vidět základní strukturu *hallbaru*. Často nás ovšem zajímají i jiné geometrie, a proto bylo prvním požadavkem simulačního programu to, aby uměl vygenerovat pět odlišných geometrií *hallbaru*, které se často používají. Geometrie na obrázku 2.12 znázorňuje první a základní typ geometrie *hallbaru*, kterou v této práci i v programu nazýváme jako *simple geometry*. Dalším požadavkem programu bylo, aby nebyly rozměry *hallbaru* fixní, ale parametrizovatelné, a aby bylo možné simulovat šíření náboje pro různé nastavení velikostí *hallbaru*. Proto na obrázku 2.12 znázorňujeme postupně parametry  $a$ ,  $b$ ,  $c$ ,  $l$  a  $w$ , které může uživatel v programu libovolně nastavit. Následujícím požadavkem bylo, aby bylo možné různým částem *hallbaru* (a to primárně u složitějších geometrií) přiřadit jiný difuzní koeficient. Tento požadavek vznikl z důvodu, že v praxi vytváříme komplikovanější geometrie *hallbaru* například lokální oxidací na struktuře *hallbaru* a vzniká tak místo grafenu oxid grafenu, který je izolantem. Ten má jiný difuzní koeficient, což je po-

## 2.2. SIMULACE ŠÍŘENÍ NÁBOJE NA HALLBARU

třeba zohlednit v simulaci. Konečným cílem programu bylo, aby si mohl uživatel nastavit časový krok a potenciál na elektrodě, se kterým chce simulaci provádět.

### 2.2.4. Konkrétní kroky výpočtu

Na obrázku 2.13 následuje souhrn příkladů pěti různých vygenerovaných geometrií, které program umí vygenerovat spolu se zahrnutím dalšího difuzního koeficientu.



Obrázek 2.13: Přehled geometrií, které umí program vygenerovat. Všechny geometrie mají stejné parametry  $a$ ,  $b$ ,  $c$ ,  $l$  a  $w$  jako na obrázku 2.12. Zleva: 1) *simple geometry*, 2) *rectangle geometry* s parametry *rectangle width* a *rectangle height*, 3) *triangle geometry* s parametry *triangle width* a *triangle height*, 4) *gap geometry* s parametrem *gap*, 5) *bridge geometry* s parametry *bridge outer width*, *bridge inner width* a *bridge height*.

Z popisu obrázku 2.13 si lze všimnout, že jsme zavedli další parametry pro charakterizaci každé geometrie. Každé geometrii přísluší sada dodatečných parametrů, které ji popisují viz následující tabulka 1.

Tabulka 1: Dodatečné parametry pro různé geometrie *hallbaru*

Název geometrie	dodatečné parametry	výchozí hodnoty popořadě ( $\mu\text{m}$ )
<i>simple geometry</i>	-	-
<i>rectangle geometry</i>	<i>rectangle width</i> , <i>rectangle height</i>	10, 4
<i>triangle geometry</i>	<i>triangle width</i> , <i>triangle height</i>	10, 4
<i>gap geometry</i>	<i>gap</i>	10
<i>bridge geometry</i>	<i>bridge outer width</i> , <i>bridge inner width</i> , <i>bridge height</i>	40, 20, 3

Difuzní koeficienty jsme dopočítali dle vztahu z práce [45]

$$D_{\text{povrch}} = \frac{w}{\rho_{\text{povrch}} \epsilon_0 \epsilon_r}, \quad (2.1)$$

kde  $w$  značí tloušťku materiálu a  $\rho_{\text{povrch}}$  povrchovou vodivost. Pro grafen je povrchová vodivost  $\rho_{\text{povrch}}$  zhruba  $100 \Omega$  dle práce [46]. Tím dostáváme povrchový difuzní koeficient o hodnotě  $10 \text{ m}^2/\text{s}$ . Na základě práce [47] jsme schopni dopočítat difuzní koeficient pro oxid grafenu, který vychází orientačně jako  $10^{-4} \text{ m}^2/\text{s}$ . Nakonec pro  $\text{SiO}_2$  se pohybuje difuzní koeficient okolo hodnoty  $10^{-16} \text{ m}^2/\text{s}$  (více v kapitole 2.2.6).

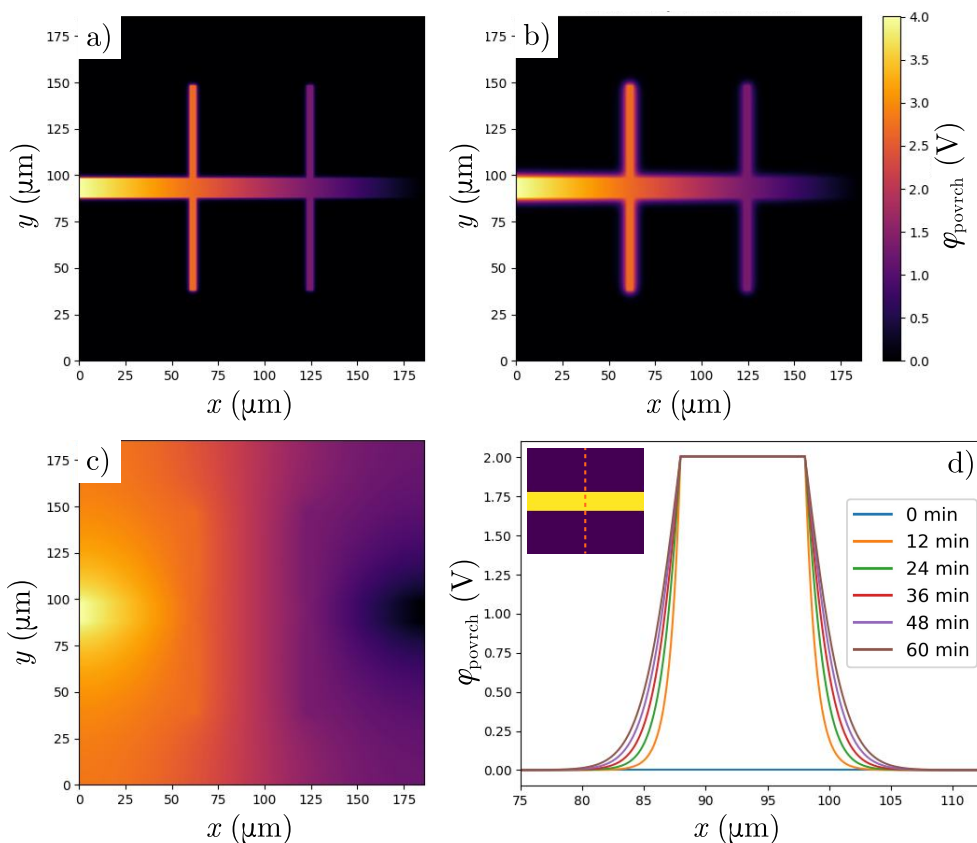
Před spuštěním simulace je nezbytné se zamyslet nad velikostí časového kroku. Program sice umožňuje uživateli časový krok libovolně nastavit, ovšem pro zajištění konvergence řešení je třeba jej omezit. Pro náš případ využíváme podmínku stanovenou ze zdroje [48] v podobě

$$\Delta t \leq \frac{\Delta x^2 \Delta y^2}{2 \max(D_{\text{povrch}})(\Delta x^2 + \Delta y^2)}, \quad (2.2)$$

kde  $\Delta t$  značí časový krok výpočtu a  $\Delta x$  spolu s  $\Delta y$  značí po řadě velikost dílku sítě ve směru  $x$  a ve směru  $y$ . V praxi se ovšem ukázalo, že lze úlohy simulovat s mnohonásobně většími časovými kroky bez ztráty přesnosti výpočtu.

### 2.2.5. Simulace

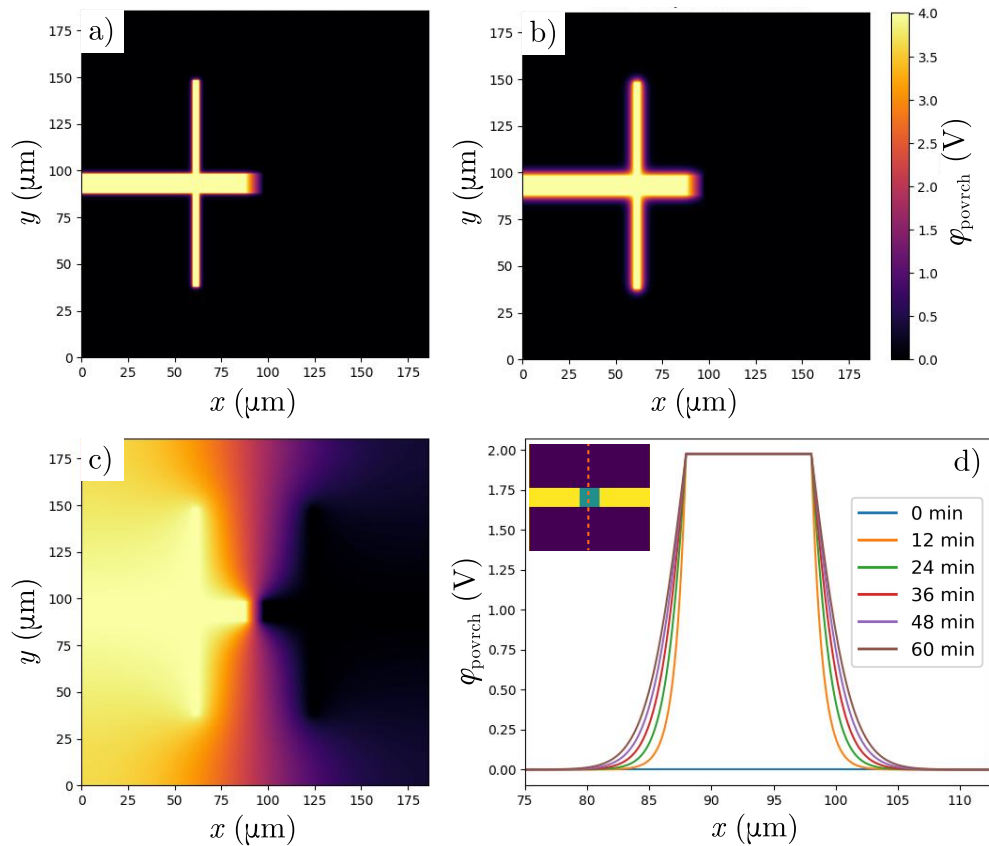
V této části prezentujeme postupně výsledky simulací po každých 12 minutách, u kterých byl užit časový krok  $\Delta t = 1$  s a síť o velikosti  $\Delta x = \Delta y = 0,186 \mu\text{m}$  (neboli 1000 na 1000 dílků sítě). Difuzní koeficienty byly zvoleny  $10^{-15} \text{m}^2/\text{s}$  pro  $\text{SiO}_2$ ,  $10 \text{m}^2/\text{s}$  pro grafen a  $10^{-4} \text{m}^2/\text{s}$  pro oxid grafenu dle kapitoly 2.2.4.



Obrázek 2.14: Simulace šíření náboje na *simple geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Simulované rozložení náboje v čase odpovídá časovému vývoji v experimentu.

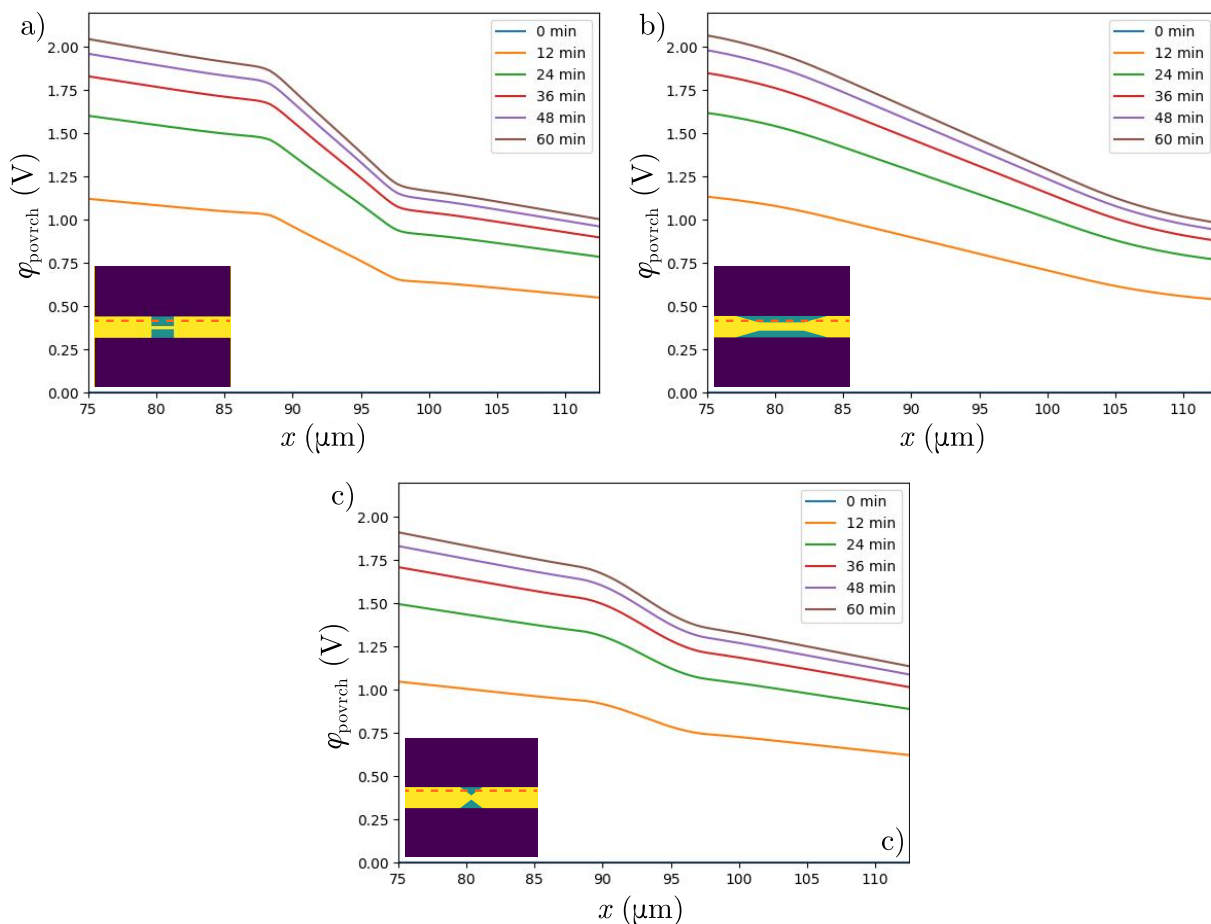
Veškeré další simulace mají velice podobný průběh šíření náboje, ustálený stav i vývoj povrchového náboje v řezu hlavním kanálem. Jediná geometrie, která je značně odlišná, je *gap geometry*, u které můžeme pozorovat, že je potenciál na pravé části hlavního kanálu téměř nulový (viz následující obrázek 2.15).

## 2.2. SIMULACE ŠÍŘENÍ NÁBOJE NA HALLBARU



Obrázek 2.15: Simulace odlišného šíření náboje na *gap geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Nejvíce odlišné rozložení povrchového potenciálu od ostatních geometrií, kde i v ustáleném stavu je téměř nulový potenciál na celé pravé části *hallbaru*.

Pro prezentování dalších geometrií uvádíme obrázek 2.16, který ukazuje jejich významné rozdíly při šíření náboje v řezu hlavním kanálem. Z obrázků můžeme vyvodit dva hlavní poznatky. První je velikost prostoru (uprostřed grafu), na kterém nabývá křivka větší strmosti. Ta nám říká, že v této oblasti dochází k šíření náboje menší rychlostí (neboť dochází k rychlejšímu poklesu potenciálu). Druhý a podstatnější poznatek je, jak moc je křivka zakulacená v bodě jejího zlomu (tedy tam, kde je fyzicky rozhraní mezi oblastmi s jinými difuzními koeficienty). Tento fakt vypovídá o tom, jak moc do oblasti řezu přitéká náboj z okolí. U *rectangle geometry* vidíme, že je zlom velmi razantní a i z geometrie plyne, že do roviny řezu nemůže náboj difundovat odjinud než z levé strany hlavního kanálu. Oproti tomu *bridge geometry* má zlom v grafu velice zakulacený a plynulý. Z toho lze usoudit (a na geometrii je to i patrné), že náboj do řezu nepřitéká jen z levé strany hlavního kanálu, ale i například z prostřední části hlavního kanálu, která se nachází pod oblastí řezu. (Kompletní přehled simulací všech geometrií přikládáme v Příloze A.)



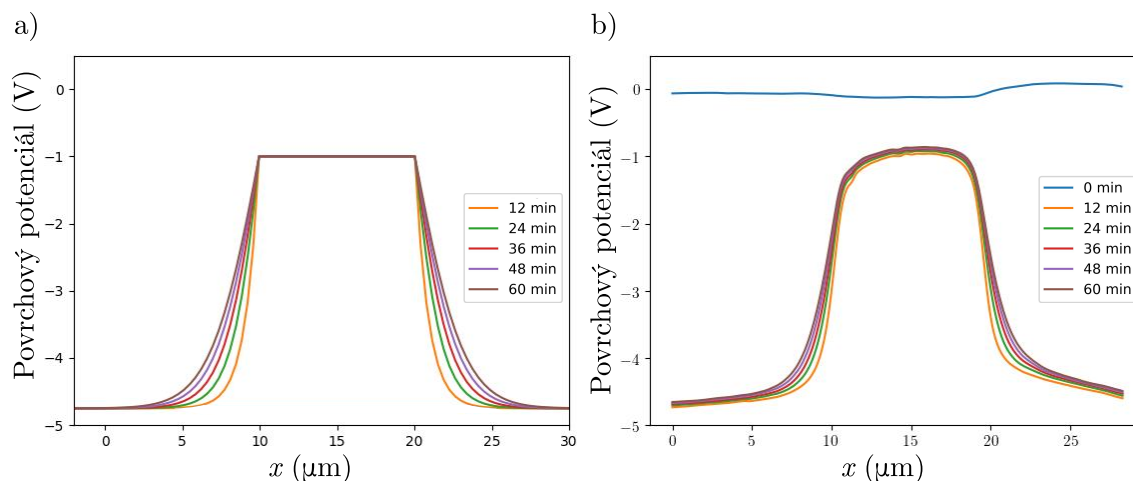
Obrázek 2.16: Porovnání simulací *rectangle geometry* - a), *triangle geometry* - c) a *bridge geometry* - b) pro různé časy v podélném řezu hlavním kanálem. Z grafů lze určit, o jakou geometrii se jedná na základě velikosti intervalu, kde dochází k pomalejšímu šíření náboje a na základě rychlosti změny šíření mezi rozhraními.

## 2.2.6. Porovnání s experimentem

Experimentální data zde používáme jak pro porovnání s výslednými simulacemi, tak pro naladění mnoha parametrů užívaných v simulaci. První využití experimentálních dat proběhlo při porovnání časového měřítka simulace s reálnými daty naměřenými pomocí *KPFM* v uspořádání *simple geometry*. Zde můžeme pozorovat, že dynamika šíření náboje z *hallbaru* na substrát probíhá v řádu desítek minut. Proto prezentujeme výsledky v podobném časovém měřítku. Experimentální data jsme pak použili znovu pro odhad difuzního koeficientu  $\text{SiO}_2$ , jehož hodnota se těžko dohledává. Srovnali jsme proto simulace pro různé difuzní koeficienty  $\text{SiO}_2$  s experimentem, ve kterém je *hallbar* na fixním potenciálu a pouze sledujeme difuzi z něj na substrát v čase (obrázek 2.17). Z porovnání simulace a experimentu jsme určili přibližný difuzní koeficient  $\text{SiO}_2$  o hodnotě  $10^{-15} \text{ m}^2/\text{s}$ . Experiment probíhal za relativní vlhkosti 50 %, což ovlivňuje mnoho faktorů, proto je odhadovaná hodnota difuzního koeficientu spíše orientační.



## 2.3. CHARAKTERIZACE GALLIUM NITRIDOVÝCH OBJEKTŮ NA GRAFENU



Obrázek 2.17: Porovnání šíření náboje v příčném řezu přes hlavní kanál *hallbaru* a) z pohledu simulace a b) z pohledu experimentu měřeného kelvinovou sondovou mikroskopií. Substrát byl připojen na napětí  $-5 \text{ V}$  zatímco *hallbar* byl na fixním potenciálu  $-1 \text{ V}$ . Převzato a upraveno od Ing. Vojtěcha Švarce z práce [23].

### 2.2.7. Vyhodnocení výsledků

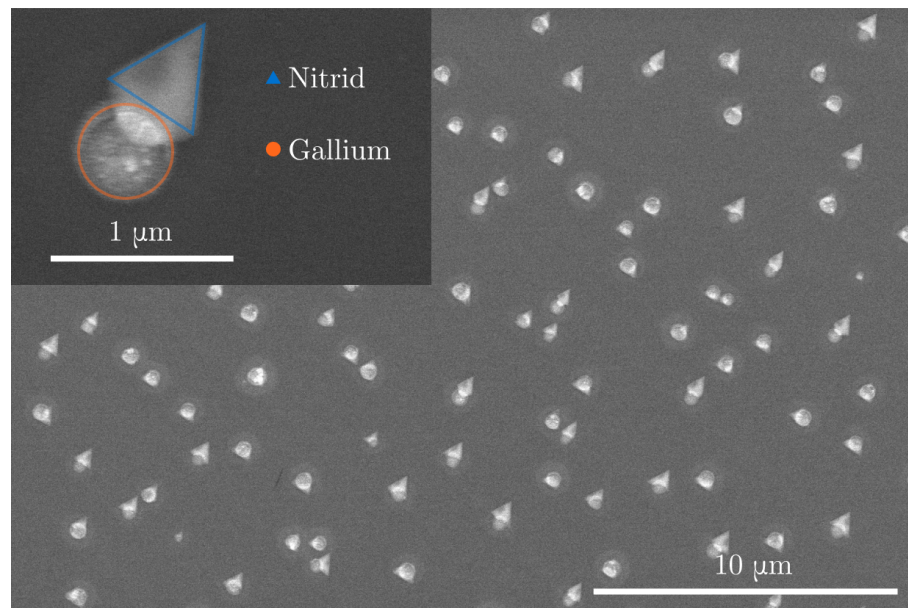
Ze všech obrázků simulací můžeme pozorovat, že se geometrie chovají velmi podobně s výjimkou *gap geometrie*, kde i pro minimální mezeru je náročné rozšířit náboj na pravou část hlavního kanálu *hallbaru*. Rozložení náboje vypadá u všech simulací velice podobně. Stejně tak můžeme u všech simulací pozorovat, že je rozložení náboje symetrické, což je dobrá kontrola toho, že je simulace v pořádku, neboť i celá geometrie problému je symetrická. Co se výpočetní náročnosti týče, výpočet každého kroku všech simulací trval zhruba 3 s při mřížce o velikosti 1000 na 1000 buněk. Tento čas jsme ovšem schopni značně zredukovat zmenšením mřížky na 200 na 200 buněk. Program je také dobře rozšiřitelný, neboť jsme schopni zkonstruovat libovolné 2D pole difuzních koeficientů, což vede k mnoha užitečným aplikacím, jako například import pole z programů pro litografii a podobně.

## 2.3. Charakterizace gallium nitridových objektů na grafenu

### 2.3.1. Uvedení problému

V této části pracujeme s podobnými daty jako v prvním úkolu (kapitola 2.1). Výsledkem experimentu je vzorek plný gallium nitridových struktur, a proto pracujeme se snímky podobnými jako na obrázku 2.18. Za veškeré snímky z rastrovacího elektronového mikroskopu v této úloze vděčím Bc. Jakubovi Štastnému.





Obrázek 2.18: Ukázka gallium nitridových struktur. Vlevo nahoře: detail jedné gallium nitridové struktury, kde kulička se sestává z gallia a trojúhelníček z nitridu. Zbytek obrázku tvoří soubor několika gallium nitridových struktur - jedná se o typický obrázek, který v této kapitole vyhodnocujeme. Hlavní obrázek i detail byly zachyceny v rastrovém elektronovém mikroskopu.

### 2.3.2. Požadavky na program

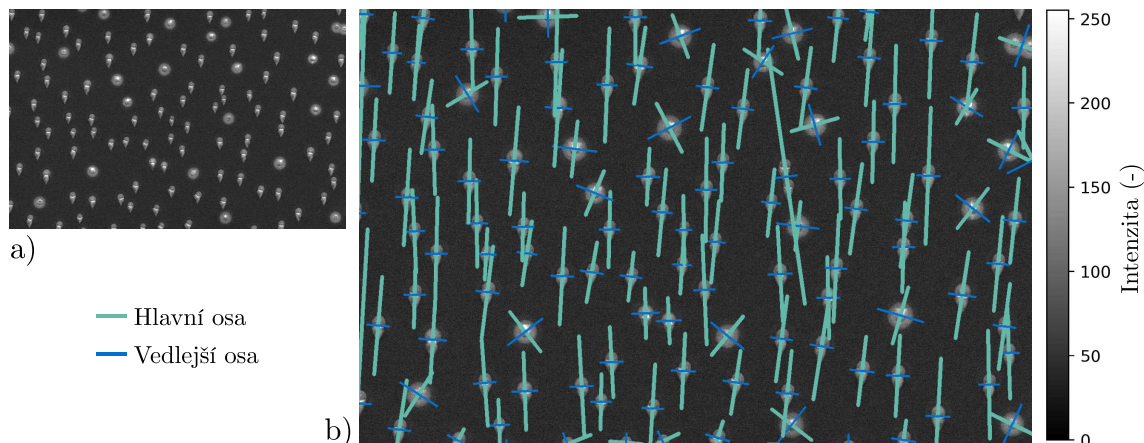
Žádané funkce programu zde byly velice podobné jako u úlohy s detekcí galliových kuliček (kapitola 2.1.2). Na jednu stranu program už nepotřeboval zjišťovat vzdálenosti mezi nejbližšími sousedy a celkové pokrytí vzorku. Jelikož byla tato funkcionalita ale implementována v úloze hledání galliových kuliček, byla přenesena i sem. Na druhou stranu je zde navíc funkcionalita umožňující analýzu orientace objektů. Na obrázku 2.18 si lze všimnout, že většina struktur již není bodově symetrická (ale pouze osově) a lze jim proto přisoudit orientaci natočení.

### 2.3.3. Konkrétní kroky analýzy obrazu

Veškerá příprava dat zůstává v této úloze identická jako u hledání galliových kuliček. Mění se ale užitá metoda pro detekci objektů. V úloze s analýzou galliových kuliček se ukázala jako nejúspěšnější metoda laplaciánu gaussianu. Ta je ale značně časově náročná a data, která zpracováváme v této úloze, jsou dobře analyzovatelná pomocí metody hledání kontur. Je tomu tak primárně z důvodu, že zde nejsou struktury natolik namačkány na sebe a nedochází k jejich překryvu. Dalším důvodem k použití metody hledání kontur byla její schopnost přesně vyčlenit konturu nalezeného objektu, což je pro nás při další práci velmi nápomocné (naproti tomu metoda laplaciánu gaussianu nám určí pouze souřadnice středu nalezeného objektu). Nalezení kontur nám umožnilo každou nalezenou konturu proložit elipsou, což bylo vhodné pro potřeby popsané v dalších kapitolách (to vše opět za použití funkcí *findContours* a *fitEllipse* z knihovny *OpenCV*).

### 2.3.4. Zjišťování orientace objektů

V úvahu zde přicházela metoda, kde by probíhala aplikace masek tvaru nitridu gallia na obrázek a při shodě bychom mohli prohlásit, že byla nalezena orientace objektu. To bylo ovšem zavrženo kvůli technické i časové náročnosti. Nakonec byla použita metoda *PCA* (kapitola 1.10) implementována funkcí *PCACompute2* z knihovny *OpenCV*, která prokázala skvělé výsledky. Touto metodou jsme byli schopni určit hlavní osu (ukazující směr orientace objektu), což můžeme vidět na obrázku 2.19.

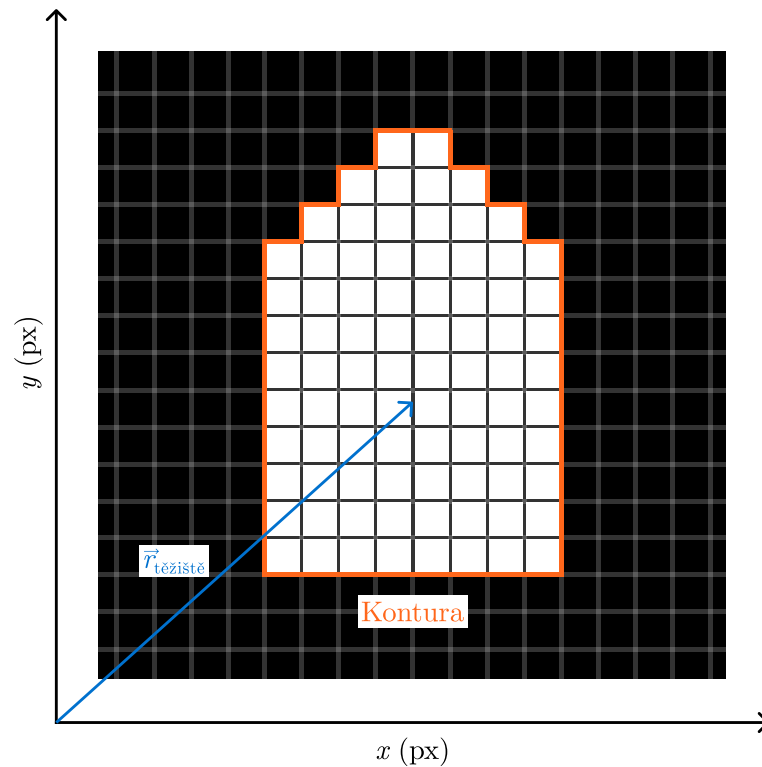


Obrázek 2.19: a) Originální obrázek. b) Detekce orientace pomocí metody *PCA*. Zelená čára ukazuje směr hlavního vlastního vektoru. Modrá ukazuje vedlejší směr (vždy kolmý k hlavnímu vlastnímu vektoru).

Pomocí metody *PCA* bohužel nelze určit, na jakou stranu podél nalezené osy je výsledná orientace. Tento problém bylo třeba adresovat použitím algoritmu pro hledání těžiště kontury. Algoritmus prochází přes veškeré body v nalezené kontuře a počítá jejich celkovou průměrnou souřadnici (analogicky lze říci, že hledá těžiště objektu). Matematicky lze problém formulovat následovně

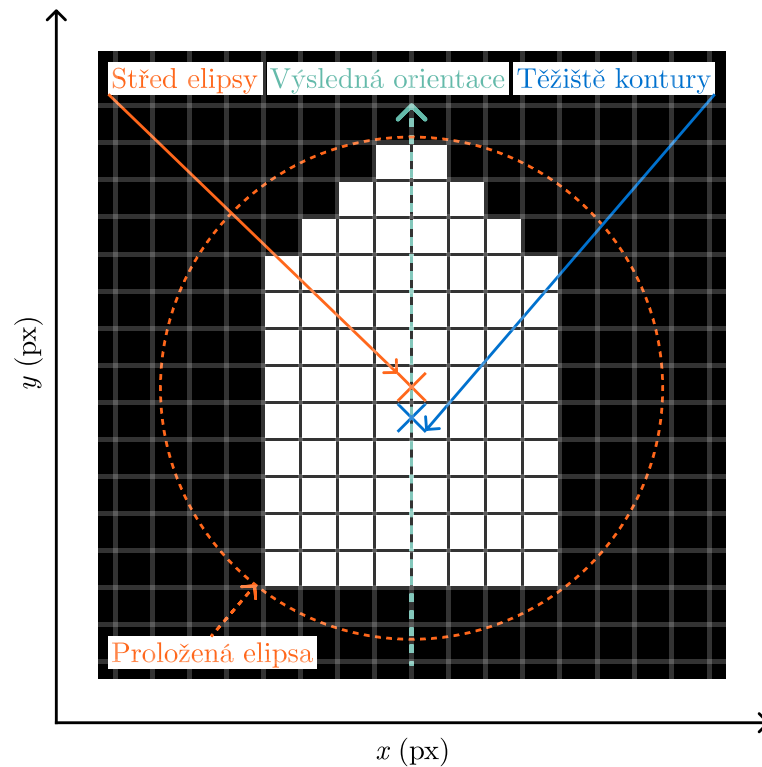
$$\vec{r}_{\text{těžiště}} = \frac{1}{N} \sum_{i=0}^N \vec{r}_i, \quad (2.3)$$

kde  $\vec{r}_{\text{těžiště}}$  je výsledná poloha těžiště kontury,  $\vec{r}_i$  je poloha každého pixelu v dané kontuře a  $N$  je počet pixelů kontury. Znázornění této techniky vidíme na obrázku 2.20.



Obrázek 2.20: Ukázka vektoru těžiště kontury.

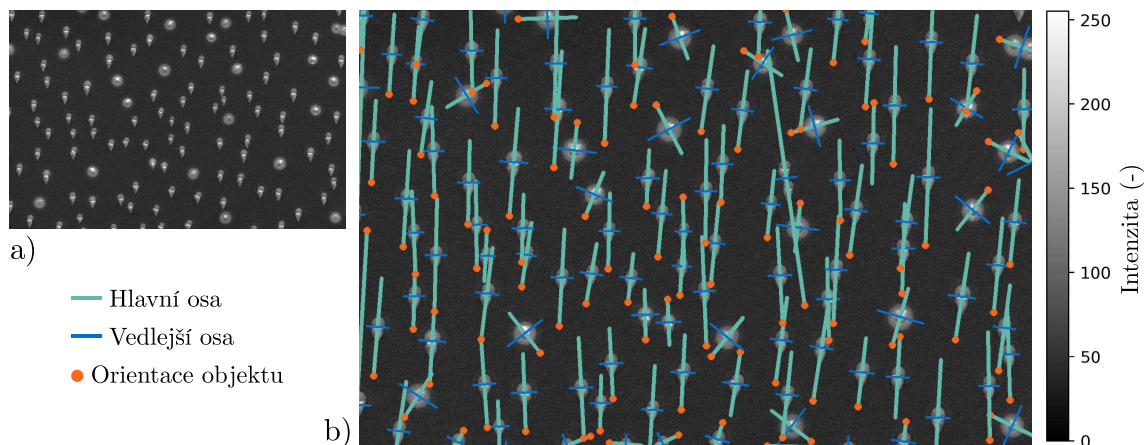
Díky proložení kontury elipsou máme informaci o jejím geometrickém středu. Pokud srovnáme polohu geometrického středu s polohou nalezeného těžiště kontury, můžeme z toho zjistit, na jakou stranu je objekt orientován. Celou situaci demonstruje obrázek 2.21.



Obrázek 2.21: Porovnání spočítané polohy vektoru těžiště kontury se středem proložené elipsy. S pomocí obou informací jsme schopni určit orientaci objektu.

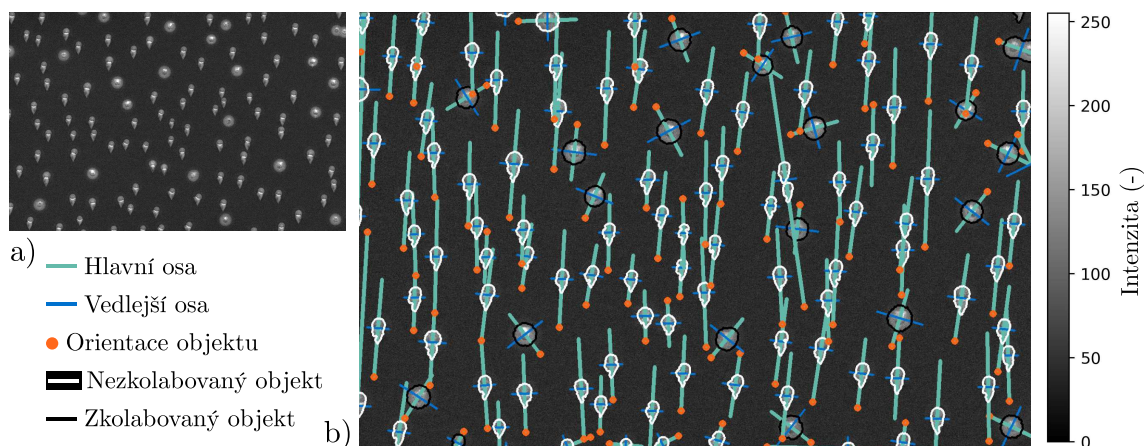
### 2.3. CHARAKTERIZACE GALLIUM NITRIDOVÝCH OBJEKTŮ NA GRAFENU

Aplikaci tohoto postupu a nalezení orientace objektu v praxi vidíme na obrázku 2.22.



Obrázek 2.22: a) Originální obrázek. b) Detekce orientace objektů s implementovanou metodou pro hledání těžiště kontury. Z toho vychází oranžová kulička určující jednoznačnou orientaci objektu.

Dále si můžeme na původním obrázku 2.18 všimnout, že některé objekty jsou tvořeny správně z galliové kuličky a na ní posazeném trojúhelníčku nitridu. Takovým objektům je možné přisoudit orientaci. Některé objekty ovšem zkolabují a nitridový trojúhelníček překrývá galliovou kuličku. V tomto případě pak bývají struktury už téměř symetrické a orientaci jim nelze přisoudit. Proto jsme v programu implementovali další metodu, která dokáže rozlišit, jestli se jedná o správnou, nebo zkolabovanou strukturu. Tato funkcionality vychází z poměru velikostí hlavního a vedlejšího vlastního vektoru, které dostáváme z *PCA* metody. Je logické, že zatímco pro správné struktury budou velikosti obou vektorů v nepoměru, naproti tomu pro zkolabované struktury (které vypadají v podstatě jako kulička) se bude jejich poměr blížit k jedničce, jelikož kruhovitý objekt nemá žádnou preferenční osu symetrie). Zde se jako dobře fungující mezní hodnota ukázalo číslo 1,3 (poměr velikostí hlavního vůči vedlejšímu vlastnímu vektoru). Finální vyhodnocení můžeme pozorovat na následujícím obrázku 2.23.

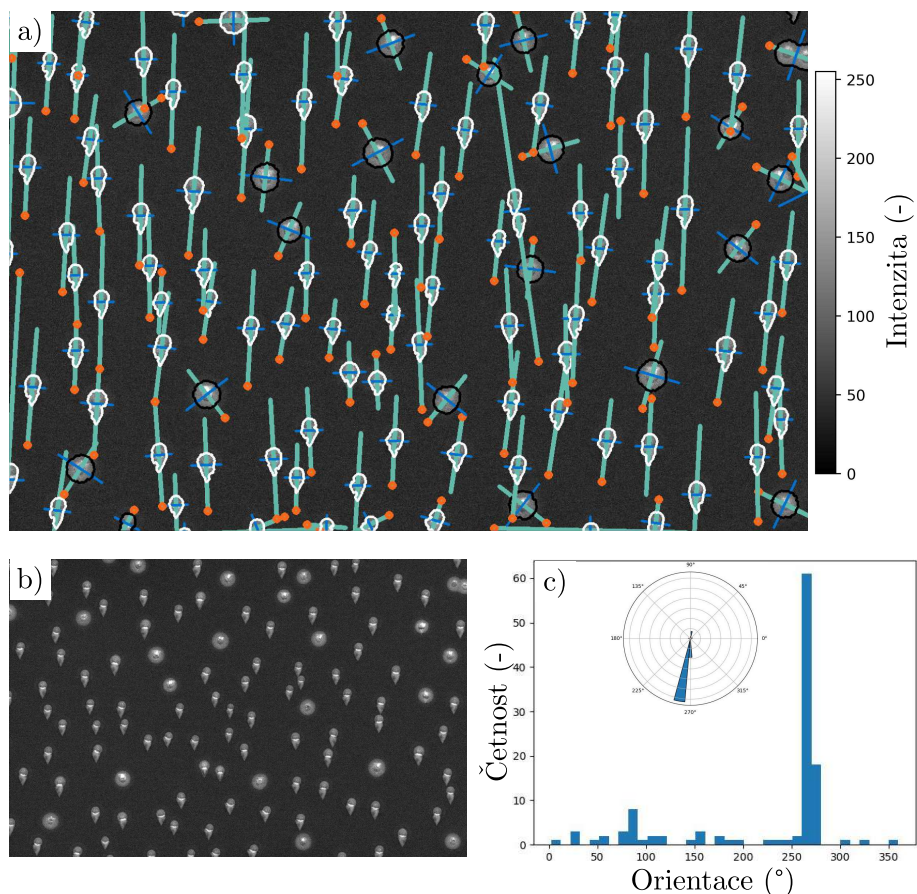


Obrázek 2.23: a) Originální obrázek. b) Detekce orientace objektů obohacena o rozpoznání, zda-li se jedná o běžnou gallium nitridovou strukturu (bílá hranice), nebo o zkolabovanou strukturu (černá hranice).



### 2.3.5. Vyhodnocení výsledků

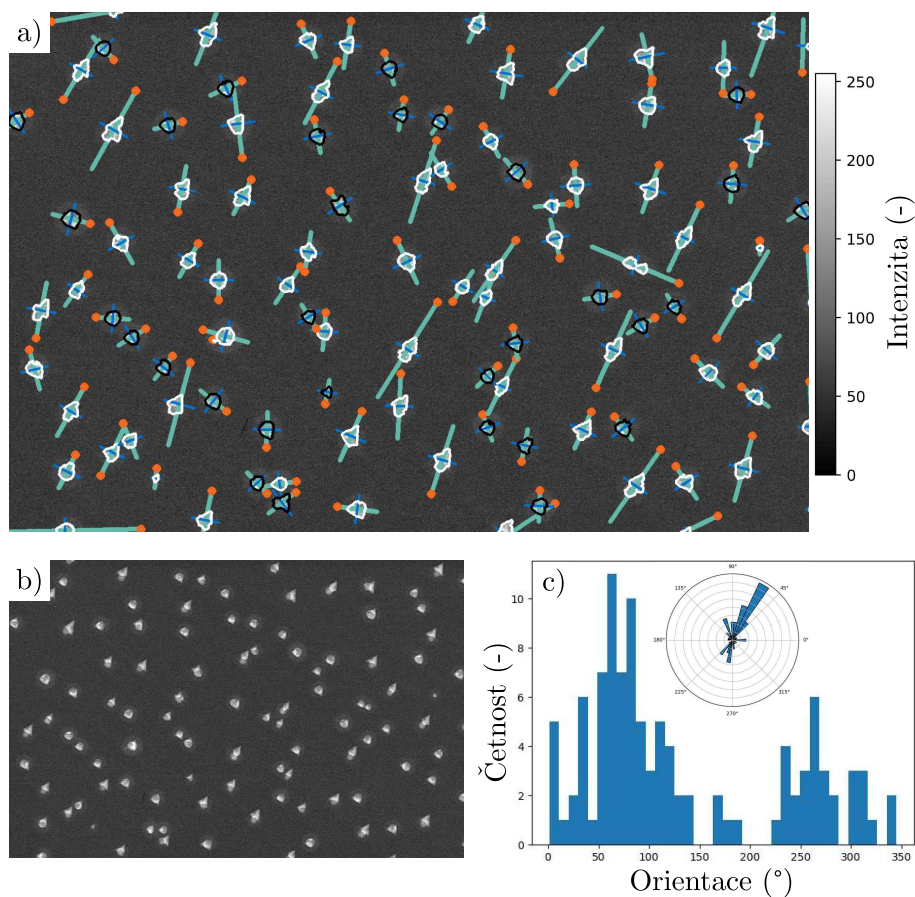
Metoda hledání kontur se i přes svůj neúspěch při aplikaci v první úloze ukázala jako dobrá volba pro tento úkol. Je tomu tak hlavně díky jiným vstupním datům, kde nejsou detekované objekty příliš blízko a obecně se jedná o snímky, které se jednodušeji analyzují. Použití *PCA* metody spolu s algoritmem pro nalezení těžiště kontury se ukázalo jako efektivní kombinace pro nalezení orientace objektů. Nakonec prezentujeme dva snímky vyhodnocené algoritmem pro srovnání (obrázky 2.24 a 2.25).



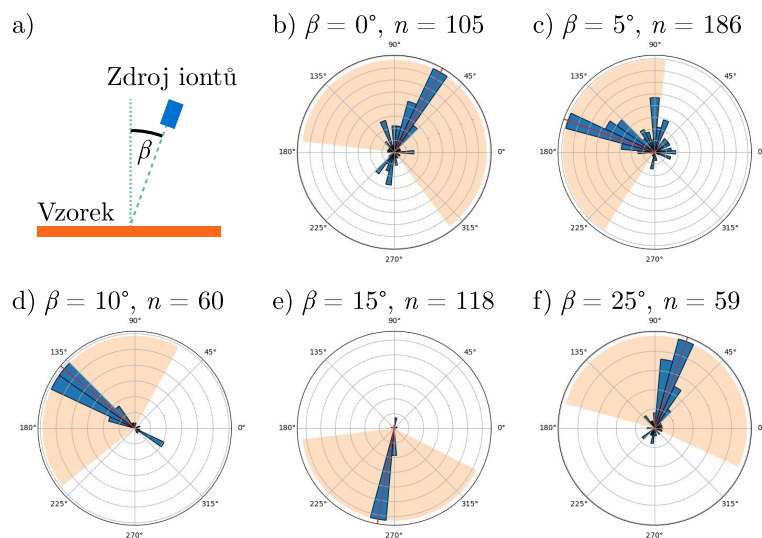
Obrázek 2.24: a) Analyzovaný obrázek gallium nitridových struktur s převážnou orientací pod úhlem  $270^\circ$  od směru vpravo. b) Původní obrázek. c) Histogram orientace objektů ukazující preferenční orientaci.

Bylo prověřeno, že účinnost detekce objektů co do jejich počtu i polohy byla shodná ve srovnání s vyhodnocením člověkem. Stejně tak při vyhodnocení orientace byla shoda výsledků vyhodnocení člověkem a algoritmem téměř perfektní - výsledky se lišily zhruba u 3 % detekovaných struktur. Do odchylky ovšem nezapočítáváme zkolabované struktury. Jejich orientaci člověk z principu vždy vyhodnotí jinak než program (jelikož mají kruhovitý tvar a nelze jim přisoudit jednoznačnou orientaci). Jediný případ, kdy dochází k opravdové chybě ve vyhodnocení snímků programem, je detekce orientace kontur, které jsou na okraji obrázků. (Kompletní přehled analýz všech obrázků gallium nitridových struktur přikládáme v Příloze B.)

### 2.3. CHARAKTERIZACE GALLIUM NITRIDOVÝCH OBJEKTŮ NA GRAFENU

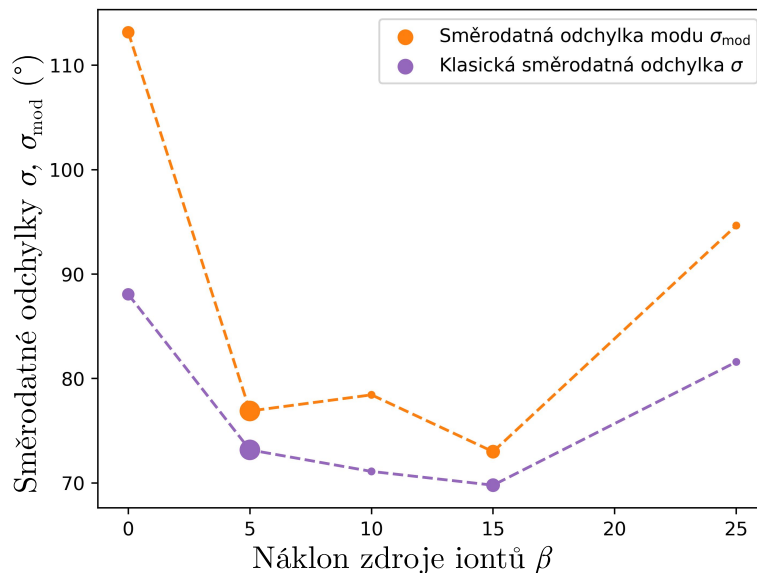


Obrázek 2.25: a) Analyzovaný obrázek gallium nitridových struktur s rovnoměrnější distribucí zastoupených orientací. b) Původní obrázek. c) Histogram orientace objektů ukazující více vyváženou distribuci.



Obrázek 2.26: a) Schéma znázorňující situaci v experimentu. Zdroj iontů svírá se vzorkem definovaný úhel  $\beta$ . b) až f) Vyhodnocení distribuce orientací gallium nitridových struktur pro daný úhel  $\beta$  s nalezeným počtem struktur  $n$ . Modré bloky znázorňují samotný histogram úhlů. Červená úsečka prezentuje modus rozdělení úhlů a oranžové pozadí interval směrodatné odchylky modu  $\sigma_{\text{mod}}$  symetricky na obě strany od modu.

Program pak lze použít pro vyhodnocení například toho, jak a jestli vůbec závisí distribuce detekovaných úhlů na natočení zdroje iontů vůči vzorku. Takové vyhodnocení dat prezentujeme na obrázku 2.26. Z dat na obrázku 2.26 bohužel není patrná žádná přímá závislost kromě toho, že směrodatná odchylka modu  $\sigma_{\text{mod}}$  (což je klasická směrodatná odchylka, ovšem počítána vůči modu a ne vůči aritmetickému průměru) klesá s rostoucím odklonem  $\beta$  až na úhel  $\beta = 25^\circ$ . To znázorňujeme spolu s klasickou směrodatnou odchylkou  $\sigma$  na obrázku 2.27.



Obrázek 2.27: Závislost směrodatných odchylek  $\sigma$  a  $\sigma_{\text{mod}}$  na úhlu odklonu zdroje iontů  $\beta$ . Velikost bodů je závislá na počtu nalezených struktur v daném obraze. Větší bod značí více nalezených bodů a proto i silnější statistický podklad.

Dat bohužel v tomto případě není mnoho a představa o klesající směrodatné odchylce (ať už klasické či modové) s rostoucím úhlem  $\beta$  je proto spíše domněnkou než potvrzeným závěrem.

## Závěr

V práci se věnujeme vývoji tří samostatných programů, které mají za úkol pomoci experimentátorům s vyhodnocováním dat a s jednodušším vytváření závěrů z daných experimentů. U obou úloh se zpracováním obrazu cílí programy na automatizaci zpracování dat, které jinak manuálně zabírá mnoho času (počítání objektů, zjišťování jejich orientace a podobně). U úlohy s numerickou simulací šíření náboje na *hallbaru* dává program experimentátorovi spíše oporu a nástroj pro simulaci chování struktur před tím, než vůbec přejde k samotnému experimentu. Jako hlavní výhodu programů vidíme hlavně jejich možnost rozšiřování a jednoduchost. Například simulaci šíření náboje lze simulovat také v Comsolu a podobných programech, ovšem nakonfigurovat takovou simulaci je pro experimentátora značně časově náročné. Mimo to lze program pro simulaci šíření náboje libovolně rozšiřovat a to ve smyslu další automatizace. Například pro hledání difuzních koeficientů tak, aby byla nalezena nejlepší shoda s experimentálními daty.

Hlavní cíle práce byly úspěšně splněny s přijatelnými výsledky. Co se týče úloh zpracování obrazu, jako nelepší metodu pro hledání objektů v obraze jsme vyhodnotili metodu laplacián gaussianu (*LoG*). Ta je sice zatížena větší časovou náročností, ale přináší skvělé výsledky. Často ovšem nestačí ani sebelepší metoda pro analýzu samotného obrazu, jelikož stále může selhat operace prahování. Proto je potřeba mít na paměti, že prahování je pro úspěšné vyhodnocení obrazu stejně důležitá operace, jako samotné určování polohy objektů. Problematika stanovení obecného algoritmu, který nalezne optimální hodnotu prahu pro každý obrázek, je značně náročná. To je dle našeho mínění oblast, kterou se lze v budoucnu dále zabývat. Výborné výsledky měla analýza hlavních komponent (*PCA*) pro získávání orientace objektu. Je výpočetně rychlá, teoreticky pochopitelná a elegantní. Myslíme si proto, že tato část práce je již spolehlivě vyřešena. V práci jsme nevyužili neuronových sítí, které by taktéž mohly vést k úspěchu. Naš program by mohl sloužit jako dobrá příprava pro trénovací data neuronové sítě. Konvoluční neuronové sítě by mohly dosahovat výborných výsledků vzhledem k tomu, že jsou navrženy právě pro práci s obrazem. Nezodpovězeným zůstává optimální počet snímků pro dobré natrénování sítě a také způsob zajištění dynamického počtu výstupů neuronové sítě k reprezentaci nalezených objektů.

U úlohy se simulací šíření náboje na *hallbaru* jsme dosáhli také uspokojivých výsledků. Knihovna *FiPy* se ukázala jako dobrá volba díky její pohodlnosti pro použití a jednoduché ovládání. Prvotně byl požadavek mít v programu možnost si vybrat z celkem pěti geometrií, které se v experimentech běžně používají. Nakonec se ovšem povedlo přijít s postupem, jak zadefinovat libovolnou geometrii, která je reprezentována obecným 2D polem difuzních koeficientů. To může v budoucnu znamenat velký potenciál například při využití importu geometrií, které jsou výstupem z jiných programů - jako třeba programy pro litografii, iontové svazky a podobně. Mimo to má uživatel volnou ruku ve volbě časového kroku, přičemž simulace s časovým krokem značně převyšujícím povolený limit dosahují dobrých výsledků. V neposlední řadě nám program také dává možnost si parametrizovat jak předvolené difuzní koeficienty, tak kompletní rozměry všech pěti generovaných geometrií.



### 3. Literatura

- [1] VANDERPLAS, Jacob T. *Python data science handbook: essential tools for working with data*. First edition. Beijing: O'Reilly, [2017]. ISBN 978-1-491-91205-8.
- [2] NOVOSELOV, K; GEIM, A. K. Electric Field Effect in Atomically Thin Carbon Films. *Science*. 2004, roč. 2004, č. 306, s. 1–4. ISSN 0036-8075.
- [3] DENG, Bing; LIU, Zhongfan; PENG, Hailin. Toward Mass Production of CVD Graphene Films. *Advanced Materials* [online]. 2019, roč. 31, č. 9, s. 1–25 [cit. 2024-04-07]. ISSN 0935-9648. Dostupné z DOI: [10.1002/adma.201800996](https://doi.org/10.1002/adma.201800996).
- [4] NAUMIS, Gerardo G; BARRAZA-LOPEZ, Salvador; OLIVA-LEYVA, Maurice; TERRONES, Humberto. Electronic and optical properties of strained graphene and other strained 2D materials: a review. *Reports on Progress in Physics* [online]. 2017-09-01, roč. 80, č. 9, s. 1–63 [cit. 2024-04-07]. ISSN 0034-4885. Dostupné z DOI: [10.1088/1361-6633/aa74ef](https://doi.org/10.1088/1361-6633/aa74ef).
- [5] NEZVAL, D; BARTOŠÍK, M; MACH, J; PIASTEK, J; ŠVARC, V; KONEČNÝ, M; ŠIKOLA, T. Density functional study of gallium clusters on graphene: electronic doping and diffusion. *Journal of Physics: Condensed Matter* [online]. 2021-01-13, roč. 33, č. 2, 1 až 15 [cit. 2024-05-08]. ISSN 0953-8984. Dostupné z DOI: [10.1088/1361-648X/abb683](https://doi.org/10.1088/1361-648X/abb683).
- [6] BEISER, Arthur. *Úvod do moderní fyziky: vysokoškolská učebnice*. 2. vyd. Praha: Academia, 1978.
- [7] FUCHS; OLIVER, Mark; GOERBIG, Jean-Noel. Introduction to the physical properties of graphene. *Lecture notes*. 2008, roč. 2008, č. 10, s. 1–69.
- [8] Graphene, new physics in two dimensions. *Europhysics News*. 2009, roč. 2009, č. 40, s. 17–20.
- [9] D'SOUZA, Francis; KADISH, Karl M. *Handbook of carbon nano materials*. 1. vyd. New Jersey: World Scientific, [2014]. ISBN 978-981-4566-69-8.
- [10] TORRES, Luis E. F. Foa; ROCHE, Stephan; CHARLIER, Jean-Christophe. *Introduction to graphene-based nanomaterials: from electronic structure to quantum transport*. 1st pub. New York: Cambridge University Press, 2014. ISBN 978-1-107-03083-1.
- [11] *Electronic band structure of graphene* [online]. 2020. [cit. 2024-05-09]. Dostupné z: [https://en.wikipedia.org/wiki/Graphene%5C#/media/File:Electronic\\_band\\_structure\\_of\\_graphene.svg](https://en.wikipedia.org/wiki/Graphene%5C#/media/File:Electronic_band_structure_of_graphene.svg).
- [12] CHEN, Zhong; GANDHI, Umesh; LEE, Jinwoo; WAGONER, R.H. Variation and consistency of Young's modulus in steel. *Journal of Materials Processing Technology* [online]. 2016, roč. 227, č. 2, s. 227–243 [cit. 2024-04-07]. ISSN 09240136. Dostupné z DOI: [10.1016/j.jmatprotec.2015.08.024](https://doi.org/10.1016/j.jmatprotec.2015.08.024).
- [13] MAYOROV, Alexander S.; GORBACHEV, Roman V.; MOROZOV, Sergey V.; BRITNELL, Liam; JALIL, Rashid; PONOMARENKO, Leonid A.; BLAKE, Peter; NOVOSELOV, Kostya S.; WATANABE, Kenji; TANIGUCHI, Takashi; GEIM, A. K. Micrometer-Scale Ballistic Transport in Encapsulated Graphene at Room Temperature. *Nano Letters* [online]. 2011-06-08, roč. 11, č. 6, s. 2396–2399 [cit. 2024-04-07]. ISSN 1530-6984. Dostupné z DOI: [10.1021/nl200758b](https://doi.org/10.1021/nl200758b).

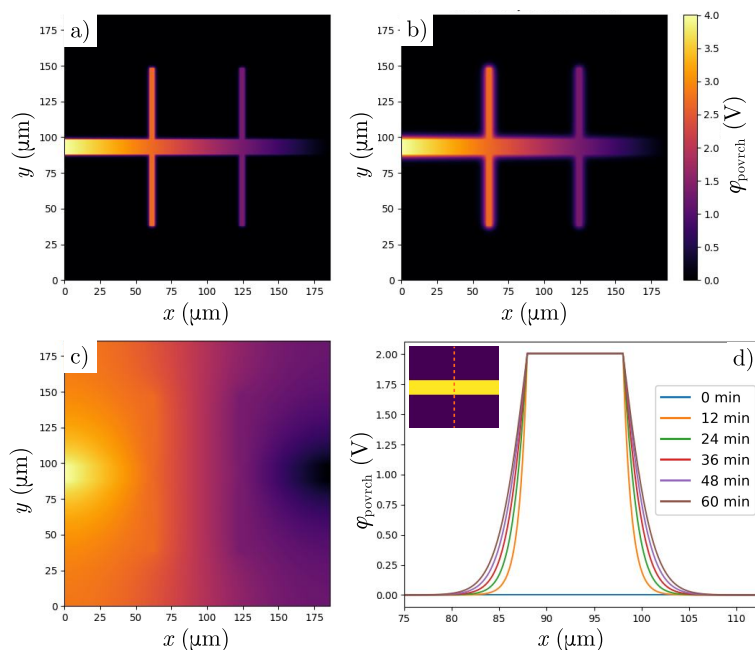
- [14] JIANG, Z.; ZHANG, Y.; TAN, Y.-W.; STORMER, H.L.; KIM, P. Quantum Hall effect in graphene. *Solid State Communications* [online]. 2007, roč. 143, č. 1-2, s. 14–19 [cit. 2024-04-07]. ISSN 00381098. Dostupné z DOI: [10.1016/j.ssc.2007.02.046](https://doi.org/10.1016/j.ssc.2007.02.046).
- [15] *Příprava grafenových vrstev pokrytých Ga atomy a charakterizace jejich elektrických vlastností*. Brno, 2015. Diplomová práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství.
- [16] YU, Young-Jun; ZHAO, Yue; RYU, Sunmin; BRUS, Louis E.; KIM, Kwang S.; KIM, Philip. Tuning the Graphene Work Function by Electric Field Effect. *Nano Letters* [online]. 2009-10-14, roč. 9, č. 10, s. 3430–3434 [cit. 2024-05-09]. ISSN 1530-6984. Dostupné z DOI: [10.1021/nl901572a](https://doi.org/10.1021/nl901572a).
- [17] MACH, J; PROCHÁZKA, P; BARTOŠÍK, M; NEZVAL, D; PIASTEK, J; HULVA, J; ŠVARC, V; KONEČNÝ, M; KORMOŠ, L; ŠIKOLA, T. Electronic transport properties of graphene doped by gallium. *Nanotechnology* [online]. 2017-10-13, roč. 28, č. 41, 1 až 23 [cit. 2024-05-09]. ISSN 0957-4484. Dostupné z DOI: [10.1088/1361-6528/aa86a4](https://doi.org/10.1088/1361-6528/aa86a4).
- [18] O'HARE, A.; KUSMARTSEV, F. V.; KUGEL, K. I. A Stable “Flat Form of Two-Dimensional Crystals: Could Graphene, Silicene, Germanene Be Minigap Semiconductors? *Nano Letters* [online]. 2012-02-08, roč. 12, č. 2, s. 1045–1052 [cit. 2024-04-06]. ISSN 1530-6984. Dostupné z DOI: [10.1021/nl204283q](https://doi.org/10.1021/nl204283q).
- [19] BAHRI, Mohamed; GEBRE, Shushay Hagos; ELAGUECH, Mohamed Amin; DAJAN, Fekadu Tsegaye; SENDEKU, Marshet Getaye; TLILI, Chaker; WANG, Deqiang. Recent advances in chemical vapour deposition techniques for graphene-based nanoarchitectures: From synthesis to contemporary applications. *Coordination Chemistry Reviews* [online]. 2023, roč. 475, č. 3, s. 1–4 [cit. 2024-04-06]. ISSN 00108545. Dostupné z DOI: [10.1016/j.ccr.2022.214910](https://doi.org/10.1016/j.ccr.2022.214910).
- [20] CHEN, Xiangping; ZHANG, Lili; CHEN, Shanshan. Large area CVD growth of graphene. *Synthetic Metals* [online]. 2015, roč. 210, č. 2, s. 95–108 [cit. 2024-05-12]. ISSN 03796779. Dostupné z DOI: [10.1016/j.synthmet.2015.07.005](https://doi.org/10.1016/j.synthmet.2015.07.005).
- [21] ZHU, Jinfeng; LIU, Qing Huo; LIN, Timothy. Manipulating light absorption of graphene using plasmonic nanoparticles. *Nanoscale* [online]. 2013, roč. 5, č. 17, 3 až 7 [cit. 2024-05-08]. ISSN 2040-3364. Dostupné z DOI: [10.1039/c3nr02660d](https://doi.org/10.1039/c3nr02660d).
- [22] ZHANG, Dandan; ZHAN, Zaiji. Strengthening effect of graphene derivatives in copper matrix composites. *Journal of Alloys and Compounds* [online]. 2016, roč. 654, č. 3, s. 226–233 [cit. 2024-05-09]. ISSN 09258388. Dostupné z DOI: [10.1016/j.jallcom.2015.09.013](https://doi.org/10.1016/j.jallcom.2015.09.013).
- [23] ŠVARC, Vojtech; BARTOŠÍK, Miroslav; KONEČNÝ, Martin; PIASTEK, Jakub; NEZVAL, David; MACH, Jindrich; ŠIKOLA, Tomas. Side charge propagation in simultaneous KPFM and transport measurement of humidity exposed graphene FET sensor. *Carbon* [online]. 2023, roč. 215, č. 1, 1 až 5 [cit. 2024-05-08]. ISSN 00086223. Dostupné z DOI: [10.1016/j.carbon.2023.118471](https://doi.org/10.1016/j.carbon.2023.118471).

- [24] GOH, Ta Yang; BASAH, Shafriza Nisha; YAZID, Haniza; SAFAR, Muhammad Juhairi Aziz; SAAD, Fathinul Syahir Ahmad. Performance analysis of image thresholding: Otsu technique. *Measurement* [online]. 2018, roč. 114, č. 1, s. 298–307 [cit. 2024-05-12]. ISSN 02632241. Dostupné z DOI: [10.1016/j.measurement.2017.09.052](https://doi.org/10.1016/j.measurement.2017.09.052).
- [25] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. 1. vyd. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-233-6.
- [26] MORSE, Brian. *Lecture 4: Thresholding*. 2000. Dostupné také z: [https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/threshold.pdf](https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf).
- [27] *Dilation (morphology)* [online]. Wikipedia, 2004. [cit. 2024-02-22]. Dostupné z: [https://en.wikipedia.org/wiki/Dilation\\_\(morphology\)](https://en.wikipedia.org/wiki/Dilation_(morphology)).
- [28] *Eroding and Dilating* [online]. 2024. [cit. 2024-05-08]. Dostupné z: [https://docs.opencv.org/3.4/db/df6/tutorial\\_erosion\\_dilatation.html](https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html).
- [29] *Erosion (morphology)* [online]. Wikipedia, 2004. [cit. 2024-02-22]. Dostupné z: [https://en.wikipedia.org/wiki/Dilation\\_\(morphology\)](https://en.wikipedia.org/wiki/Dilation_(morphology)).
- [30] JAMIL, Nursuriati; SEMBOK, Tengku Mohd Tengku; BAKAR, Zainab Abu. Noise removal and enhancement of binary images using morphological operations. In: *2008 International Symposium on Information Technology* [online]. IEEE, 2008, s. 1–6 [cit. 2024-03-11]. ISBN 978-1-4244-2327-9. Dostupné z DOI: [10.1109/ITSIM.2008.4631954](https://doi.org/10.1109/ITSIM.2008.4631954).
- [31] CHOW, James C.L. Chapter 2 - Photon and electron interactions with gold nanoparticles: A Monte Carlo study on gold nanoparticle-enhanced radiotherapy. In: GRUMEZESCU, Alexandru Mihai (ed.). *Nanobiomaterials in Medical Imaging*. William Andrew Publishing, 2016, s. 45–70. ISBN 978-0-323-41736-5. Dostupné z DOI: <https://doi.org/10.1016/B978-0-323-41736-5.00002-9>.
- [32] *Finding local maxima* [online]. Scikit-image, 2013. [cit. 2024-03-11]. Dostupné z: [https://scikit-image.org/docs/stable/auto\\_examples/segmentation/plot\\_peak\\_local\\_max.html](https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_peak_local_max.html).
- [33] LEAVERS, V.F. Which Hough Transform? *CVGIP: Image Understanding* [online]. 1993, roč. 58, č. 2, s. 250–264 [cit. 2024-05-12]. ISSN 10499660. Dostupné z DOI: [10.1006/ciun.1993.1041](https://doi.org/10.1006/ciun.1993.1041).
- [34] MALIK, Jitendra; BELONGIE, Serge; LEUNG, Thomas; SHI, Jianbo. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision* [online]. 2001, roč. 43, č. 1, s. 7–27 [cit. 2024-04-07]. ISSN 09205691. Dostupné z DOI: [10.1023/A:1011174803800](https://doi.org/10.1023/A:1011174803800).
- [35] ARCELLI, Carlo. Pattern thinning by contour tracing. *Computer Graphics and Image Processing* [online]. 1981, roč. 17, č. 2, s. 130–144 [cit. 2024-05-12]. ISSN 0146664X. Dostupné z DOI: [10.1016/0146-664X\(81\)90021-6](https://doi.org/10.1016/0146-664X(81)90021-6).
- [36] WAKAYAMA, Tadao. A Core-Line Tracing Algorithm Based on Maximal Square Moving. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 1982, roč. PAMI-4, č. 1, s. 68–74 [cit. 2024-05-03]. ISSN 0162-8828. Dostupné z DOI: [10.1109/TPAMI.1982.4767198](https://doi.org/10.1109/TPAMI.1982.4767198).

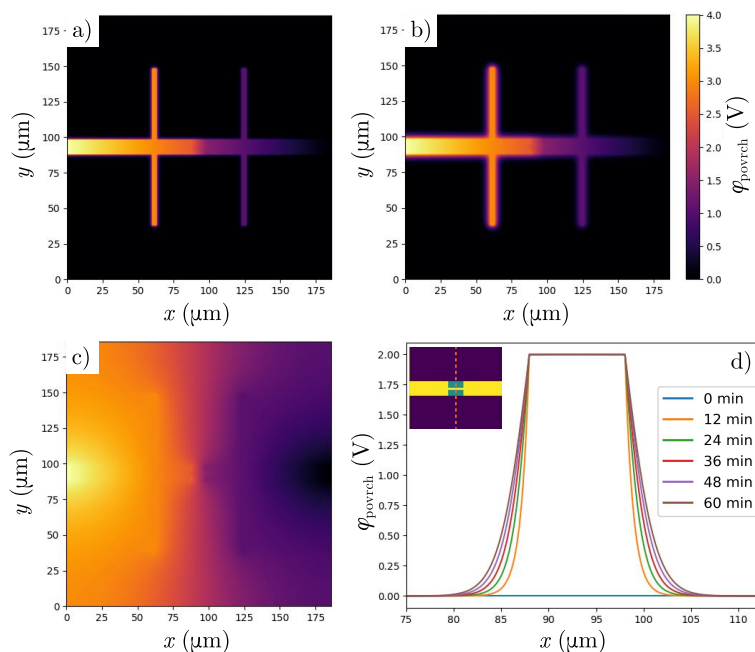
- [37] SUZUKI, Satoshi; BE, KeiichiA. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* [online]. 1985, roč. 30, č. 1, s. 32–46 [cit. 2024-04-07]. ISSN 0734189X. Dostupné z DOI: [10.1016/0734-189X\(85\)90016-7](https://doi.org/10.1016/0734-189X(85)90016-7).
- [38] KONG, Hui; AKAKIN, Hatice Cinar; SARMA, Sanjay E. A Generalized Laplacian of Gaussian Filter for Blob Detection and Its Applications. *IEEE Transactions on Cybernetics* [online]. 2013, roč. 43, č. 6, s. 1719–1733 [cit. 2024-05-12]. ISSN 2168-2267. Dostupné z DOI: [10.1109/TSMCB.2012.2228639](https://doi.org/10.1109/TSMCB.2012.2228639).
- [39] GÉRON, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Second edition. Beijing: O'Reilly, 2019. ISBN 978-1-492-03264-9.
- [40] SHLENS, Jonathon. *A Tutorial on Principal Component Analysis*. 2014. Dostupné z arXiv: [1404.1100](https://arxiv.org/abs/1404.1100) [cs.LG].
- [41] HORN, Roger A.; JOHNSON, Charles R. *Matrix analysis*. 2. vyd. New York: Cambridge University Press, 1985. ISBN 978-0-521-38632-6.
- [42] *Principal Component Analysis* [online]. 2. vyd. New York: Springer-Verlag, 2002 [cit. 2024-05-08]. ISBN 0-387-95442-2. Dostupné z DOI: [10.1007/b98835](https://doi.org/10.1007/b98835).
- [43] PAUL, Alope; LAURILA, Tomi; VUORINEN, Vesa; DIVINSKI, Sergiy V. Fick's Laws of Diffusion. In: *Thermodynamics, Diffusion and the Kirkendall Effect in Solids*. Cham: Springer International Publishing, 2014, s. 115–139. ISBN 978-3-319-07461-0. Dostupné z DOI: [10.1007/978-3-319-07461-0\\_3](https://doi.org/10.1007/978-3-319-07461-0_3).
- [44] MAZUMDER, Sandip. *Numerical Methods for Partial Differential Equations*. 1. vyd. Academic Press, 2015. ISBN 9780128498941.
- [45] KONEČNÝ, Martin; BARTOŠÍK, Miroslav; MACH, Jindřich; ŠVARC, Vojtěch; NEZVAL, David; PIASTEK, Jakub; PROCHÁZKA, Pavel; CAHLÍK, Aleš; ŠIKOLA, Tomáš. Kelvin Probe Force Microscopy and Calculation of Charge Transport in a Graphene/Silicon Dioxide System at Different Relative Humidity. *ACS Applied Materials & Interfaces* [online]. 2018-04-11, roč. 10, č. 14, s. 11987–11994 [cit. 2024-04-11]. ISSN 1944-8244. Dostupné z DOI: [10.1021/acsami.7b18041](https://doi.org/10.1021/acsami.7b18041).
- [46] NIRMALRAJ, Peter N.; LUTZ, Tarek; KUMAR, Shishir; DUESBERG, Georg S.; BOLAND, John J. Nanoscale Mapping of Electrical Resistivity and Connectivity in Graphene Strips and Networks. *Nano Letters* [online]. 2011-01-12, roč. 11, č. 1, s. 16–22 [cit. 2024-04-11]. ISSN 1530-6984. Dostupné z DOI: [10.1021/nl101469d](https://doi.org/10.1021/nl101469d).
- [47] JAAFAR, E.; KASHIF, Muhammad; SAHARI, S.K.; NGAINI, Z. Study on Morphological, Optical and Electrical Properties of Graphene Oxide (GO) and Reduced Graphene Oxide (rGO). *Materials Science Forum* [online]. 2018, roč. 917, č. 1, s. 112–116 [cit. 2024-04-10]. ISSN 1662-9752. Dostupné z DOI: [10.4028/www.scientific.net/MSF.917.112](https://doi.org/10.4028/www.scientific.net/MSF.917.112).
- [48] *The two-dimensional diffusion equation* [online]. 2020. [cit. 2024-04-10]. Dostupné z: <https://scipython.com/book/chapter-7-matplotlib/examples/the-two-dimensional-diffusion-equation/>.

# Seznam příloh

## Příloha A: Souhrn simulací šíření náboje

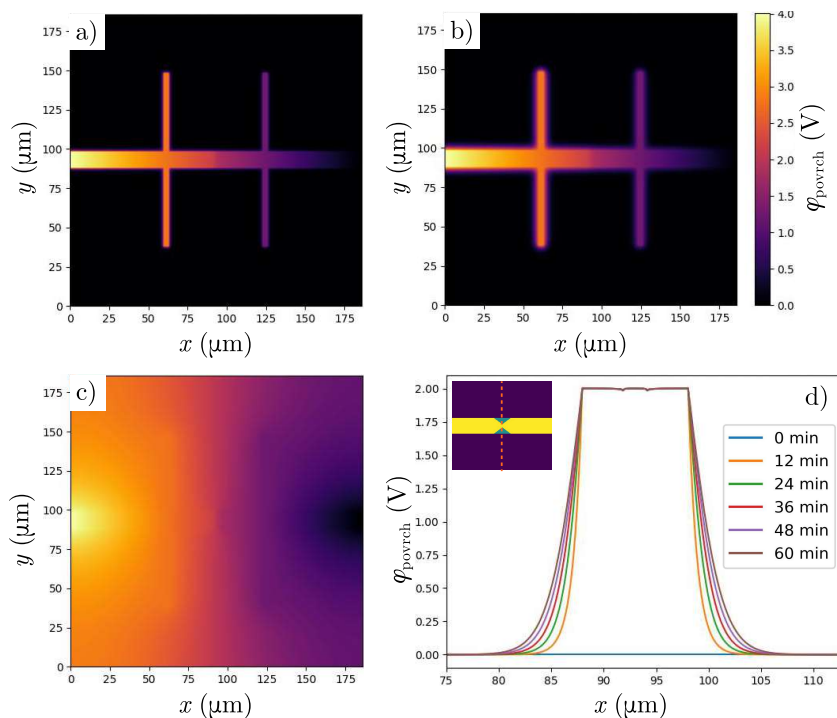


Obrázek 3.1: Simulace šíření náboje na *simple geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Simulované rozložení náboje v čase odpovídá časovému vývoji v experimentu.

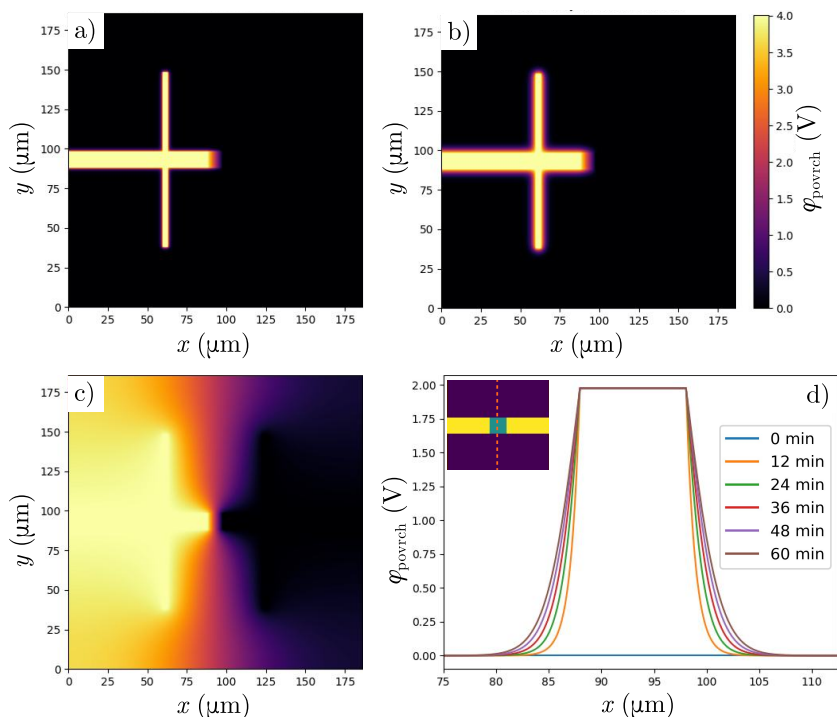


Obrázek 3.2: Simulace šíření náboje na *rectangle geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Hlavní změna nastává v ustáleném stavu a oblasti oxidovaného grafenu.

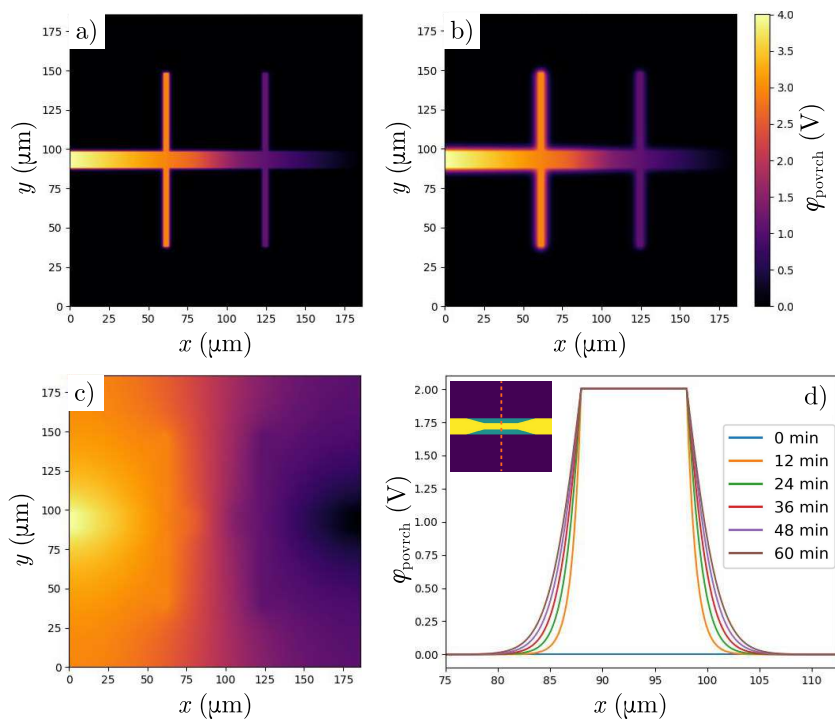




Obrázek 3.3: Simulace šíření náboje na *triangle geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Pozorujeme drobný výkyv v průběhu potenciálu v řezu hlavním kanálem.

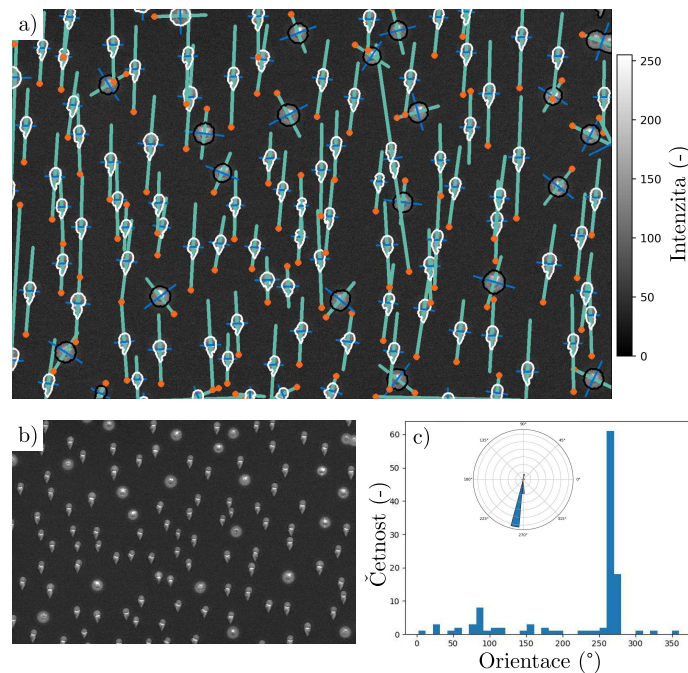


Obrázek 3.4: Simulace odlišného šíření náboje na *gap geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Nejvíce odlišné rozložení povrchového potenciálu od ostatních geometrií, kde i v ustáleném stavu je téměř nulový potenciál na celé pravé části *hallbaru*.

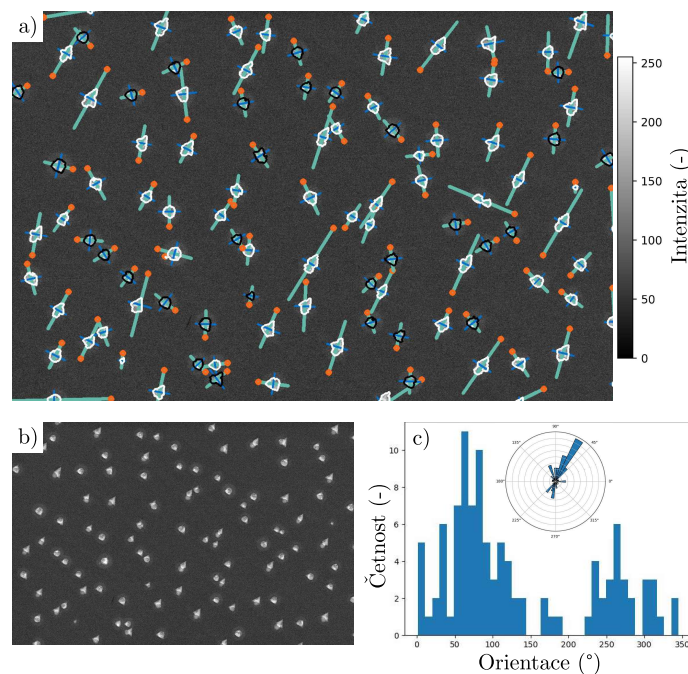


Obrázek 3.5: Simulace šíření náboje na *bridge geometry hallbaru* a) v čase 12 min, b) v čase 72 min, c) ustálený stav, d) řez hlavním kanálem v různých časech simulace. Simulace je velice podobná simulaci *simple geometry hallbaru*.

## Příloha B: Souhrn analýz obrázků gallium nitridových struktur

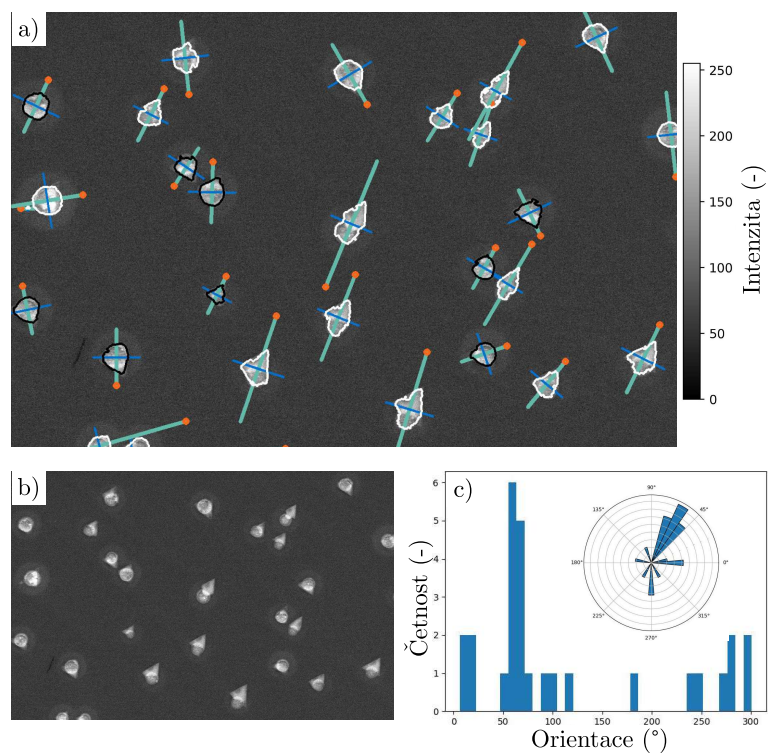


Obrázek 3.6: a) Analyzovaný obrázek gallium nitridových struktur s převážnou orientací pod úhlem  $270^\circ$  od směru vpravo. b) Původní obrázek. c) Histogram orientace objektů ukazující preferenční orientaci.



Obrázek 3.7: a) Analyzovaný obrázek gallium nitridových struktur s rovnoměrnější distribucí zastoupených orientací. b) Původní obrázek. c) Histogram orientace objektů ukazující více vyváženou distribuci.





Obrázek 3.8: a) Analyzovaný obrázek gallium nitridových struktur s nízkým počtem objektů. b) Původní obrázek. c) Histogram orientace objektů s nedostatečným počtem dat.