

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Mgr. Ján Dugáček



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## AUTOMATIZOVANÉ OPTICKÉ MĚŘENÍ KONTAKTNÍHO ÚHLU KAPKY

AUTOMATED OPTICAL MEASUREMENT OF DROPS CONTACT ANGLE

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Mgr. Ján Dugáček

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Petyovský, Ph.D.

BRNO 2018

# Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**  
Ústav automatizace a měřicí techniky

**Student:** Mgr. Ján Dugáček

**ID:** 195986

**Ročník:** 2

**Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## **Automatizované optické měření kontaktního úhlu kapky**

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem této diplomové práce je navrhnout a implementovat algoritmy, které budou schopny automatické detekce kapky na snímku a budou dále schopny vypočítat kontaktní úhel mezi kapkou a podkladovým materiálem.

- 1) Nastudujte problematiku měření povrchového napětí a proveďte v této oblasti literární rešerši.
- 2) Vyberte a popište vhodné metody sloužící k detekci kapky na povrchu měřeného materiálu.
- 3) Vytvořte vhodnou databázi snímků, která bude složít ověření funkčnosti navrhovaných algoritmů.
- 4) Navrhněte a otestujte metodu výpočtu kontaktního úhlu na základě detekované kapky ve snímku.
- 5) Navrhněte a implementujte program, který bude schopný automaticky detekovat kapku a vyhodnotit velikost kontaktního úhlu.
- 6) Navrhněte a vhodným způsobem otestujte úspěšnost vámi vytvořeného programu.
- 7) Zhodnoťte a diskutujte dosažené výsledky. Navrhněte možná vylepšení.

**DOPORUČENÁ LITERATURA:**

[1] KWOK, D.Y. and NEUMANN, A.W., Contact angle measurement and contact angle interpretation. 1991. 82p. DOI: 10.1016/S0001-8686(98)00087-6. ISSN 0001-8686.

[2] JAN, J., Medical Image Processing, Reconstruction and Restoration: Concepts and Methods. Boca Raton: CRC Press, 2005, ISBN 0-8247-5849-8.

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 14.5.2018

**Vedoucí práce:** Ing. Petr Petyovský, Ph.D.

**Konzultant:** Ing. Branislav Hesko

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Meranie povrchového napätia pevných látok zväčša vyžaduje ručné spracovanie fotografií kvapiek kvapalín so známymi vlastnosťami na povrchu meranej látky, pri ktorom sa ručne stanoví kontaktný uhol. Cieľom tejto práce je navrhnúť a implementovať algoritmus, ktorý dokáže tieto kontaktné uhly určiť z fotografií a nevyžaduje fotografie zostrojené za presne daných svetelných podmienok. Implementácia nesmie závisieť na platených knižniciach a musí byť jednoduché ju zapracovať do existujúceho softvéru.

## KLÚČOVÉ SLOVÁ

počítačové videnie, povrchové napätie, kontaktné uhly, metóda kvapiek

## ABSTRACT

Measurement of surface tension of solids usually requires manual processing of photographs of droplets of liquids with known properties on the measured material's surface, during which the contact angle is manually determined. The objective of this work is to implement an algorithm that can determine these contact angles from photographs and does not require the photographs to be made under specific light conditions. The implementation cannot depend on paid libraries and must be simple to incorporate into existing software.

## KEYWORDS

computer vision, surface tension, contact angles, droplet method

DUGÁČEK, Ján. *Automatizované optické měření kontaktního úhlu kapky*. Brno, 2018, 74 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedúci práce: Ing. Petr Petyovský, Ph.D.

## VYHLÁSENIE

Vyhlasujem, že som svoju diplomovú prácu na tému „Automatizované optické měření kontaktního úhlu kapky“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora

## POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Petrovi Petyovskému, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno .....

.....

podpis autora

# Obsah

Úvod	9
<b>1 Povrchové napätie</b>	<b>10</b>
1.1 Kontaktné uhly	10
1.2 Zariadenie na meranie kontaktných uhlov SeeSystem	11
1.3 Výpočet povrchovej energie	12
1.3.1 Zismanova teória	12
1.3.2 Neumannova teória	13
1.3.3 Teória OWRK	13
1.3.4 Acidobázická teória	14
1.4 Kvapaliny používané pri meraní povrchového napätia	14
1.5 Zhrnutie	15
<b>2 Metódy používané pre detekciu kvapky na fotografii</b>	<b>16</b>
2.1 Filtrovanie	16
2.1.1 Gaussov filter	16
2.1.2 Gradientné operátory	17
2.1.3 Binárne prahovanie	18
2.1.4 Cannyho detektor	19
2.2 Metódy hľadania oblúkov	19
2.2.1 Metódy najmenších štvorcov	20
2.2.2 Metódy založené na hlasovaní	20
2.2.3 Ďalšie metódy hľadania kružníc	21
2.3 Zhuková analýza	22
2.4 Zhrnutie	22
<b>3 Sada snímok pre overenie funkčnosti navrhovaných algoritmov</b>	<b>24</b>
<b>4 Návrh a implementácia algoritmu na meranie kontaktného uhlu</b>	<b>28</b>
4.1 Pokus o použitie štandardného postupu hľadania kružníc	28
4.2 Filtrovanie	29
4.2.1 Odstránenie šumu	31
4.2.2 Smerové derivácie	31
4.2.3 Cannyho detektor	32
4.2.4 Viacškálový Cannyho detektor	32
4.2.5 Vyplňanie oblastí	33
4.2.6 Čistenie vnútra kvapky	33
4.3 Hľadanie kvapky	34

4.3.1	Záložné riešenie . . . . .	38
4.4	Nájdenie dolného okraja kvapky . . . . .	39
4.4.1	Záložný postup . . . . .	41
4.5	Neperspektívne prístupy . . . . .	41
4.5.1	Bežná Houghova transformácia . . . . .	41
4.5.2	Gradientná Houghova transformácia . . . . .	42
4.5.3	Hľadanie roviny podložky pomocou Houghovej transformácie . . . . .	42
4.5.4	Hľadanie okrajov v obraze transformovanom do polárnych súradníc . . . . .	43
4.5.5	Hľadanie elíps . . . . .	47
4.5.6	Hľadanie kružníc aj elíps . . . . .	48
4.6	Automatické testovanie a doladovanie parametrov . . . . .	49
4.7	Zhrnutie . . . . .	51
<b>5</b>	<b>Automatické zhodnotenie kvality výsledku</b>	<b>54</b>
5.1	Kritériá na overenie správnosti výsledku . . . . .	54
5.1.1	Kritérium príslušnosti . . . . .	54
5.1.2	Kritérium približného odhadu kontaktného uhla . . . . .	56
5.1.3	Kritérium šumu . . . . .	56
5.1.4	Kritérium sústredenia . . . . .	56
5.1.5	Kritérium súvislosti . . . . .	57
5.1.6	Kritérium znečistenia . . . . .	57
5.2	Hlasovanie medzi kritériami . . . . .	57
5.3	Zvolené riešenie . . . . .	58
5.4	Využitie v algoritme . . . . .	59
5.5	Zhrnutie . . . . .	59
<b>6</b>	<b>Automatizovaný test na veľkej vzorke snímok</b>	<b>60</b>
<b>7</b>	<b>Konečné zhodnotenie algoritmu</b>	<b>62</b>
<b>8</b>	<b>Záver</b>	<b>67</b>
	<b>Literatúra</b>	<b>70</b>
	<b>Zoznam príloh</b>	<b>73</b>
<b>A</b>	<b>Obsah priloženého DVD</b>	<b>74</b>



# Zoznam obrázkov

1.1	Schématické znázornenie zariadenia SeeSystem . . . . .	11
1.2	2D znázornenie princípu určovania kontaktného uhla z fotografie . . . . .	12
1.3	3D znázornenie určovania kontaktného uhla z fotografie . . . . .	12
1.4	Trojfázové rozhranie kvapky . . . . .	13
2.1	Pôvodný obrázok (foto: autor) . . . . .	16
2.2	Obrázok po aplikácii Gaussovho filtra . . . . .	17
2.3	Obrázok po aplikácii Sobelovho operátora a prahovania . . . . .	18
2.4	Cannyho detektor . . . . .	19
3.1	Prvá fotografia používaná pri tvorbe algoritmu . . . . .	24
3.2	Rôzne vady, ktoré sa vyskytovali na fotografiách . . . . .	26
3.3	Rôzne vady, ktoré sa vyskytovali na fotografiách (pokračovanie) . . . . .	27
4.1	Úspešné nájdenie kružnice pomocou detekcie elíps v OpenCV . . . . .	29
4.2	Takmer úspešné nájdenie kružnice pomocou detekcie elíps s jednou sadou bodov v OpenCV . . . . .	29
4.3	Takmer úspešné nájdenie kružnice pomocou detekcie elíps s viacerými sadami bodov v OpenCV . . . . .	30
4.4	Neúspešné hľadanie kružnice pomocou detekcie elíps s jednou sadou bodov v OpenCV . . . . .	30
4.5	Neúspešné hľadanie kružnice pomocou detekcie elíps s viacerými sadami bodov v OpenCV . . . . .	30
4.6	Fotografia po aplikácii viacškálového Cannyho detektora . . . . .	33
4.7	Obrázok po vyplnení súvislých oblastí. Koniec ľavej časti oblúku nie je spojený so zvyškom, preto je zaradený do inej oblasti a nie je prepojený s ostatnými. . . . .	34
4.8	Obrázok bez čistenia vnútra kvapky . . . . .	34
4.9	Obrázok po neagresívnom čistení vnútra kvapky . . . . .	35
4.10	Obrázok po agresívnom čistení vnútra kvapky . . . . .	35
4.11	Diagram krokov zvoleného riešenia . . . . .	36
4.12	Maximum klasickej Houghovej transformácie na dvadsaťkrát zmenšenom obrázku (červená kružnica) . . . . .	37
4.13	Maximá klasickej Houghovej transformácie na desaťkrát zmenšenom obrázku v okolí výsledkov predchádzajúceho kroku (obrázok 4.12) . . . . .	37
4.14	Maximá po fitovaní v okolí výsledkov predchádzajúceho kroku (obrázok 4.13) . . . . .	38

4.15	Použitie metódy dotyčníc - nájdené kolmice na kružnicu sú vyznačené fialovou farbou, prieniky dvojíc týchto kolmíc ako zelené body, potenciálne kružnice červenou a vybraná kružnica žltou (pre lepšiu viditeľnosť sú farby invertované). Prvotné odhady kružníc sú zelené, pravdepodobná kružnica je fialová. . . . .	39
4.16	Nájdenie dolného okraja oblúku kvapky . . . . .	40
4.17	Houghova transformácia na lokalizáciu priamok. Maximum je jasne viditeľné. . . . .	43
4.18	Prípad, keď Houghova transformácia korektne našla dolný okraj oblúka kružnice (pre lepšiu viditeľnosť sú farby invertované) . . . . .	43
4.19	Prípad, keď bola podložka príliš neostrá na nájdenie roviny podložky (pre lepšiu viditeľnosť sú farby invertované) . . . . .	44
4.20	Prípad, keď sa oblúk kružnice nekončí na hrane podložky (pre lepšiu viditeľnosť sú farby invertované) . . . . .	44
4.21	Polárna transformácia výsledku detekcie hrán . . . . .	45
4.22	Kvalita úsečky v závislosti od polohy začiatku a konca . . . . .	45
4.23	Nájdenie koncov kružnice pomocou polárnej transformácie . . . . .	46
4.24	Problémy spôsobené mierne eliptickým tvarom kvapky . . . . .	46
4.25	Problémy spôsobené odrazom kvapky od substrátu . . . . .	46
4.26	Úspech pri nájdení malého kontaktného uhlu pomocou hľadania elíps (pre lepšiu viditeľnosť sú farby invertované) . . . . .	47
4.27	Prípad, kde vyšší počet voľných parametrov elipsy spôsobil, že fitovanie skončilo v lokálnom minime a hľadanie kružnice uspelo (pre lepšiu viditeľnosť sú farby invertované) . . . . .	48
4.28	Zablúdenie hľadania elíps fitovaním v nesúvisiacich štruktúrach (pre lepšiu viditeľnosť sú farby invertované) . . . . .	48
4.29	Nechcené hľadanie elíps aj s odrazom kvapky od substrátu (pre lepšiu viditeľnosť sú farby invertované) . . . . .	49
4.30	Veľmi častý prípad, kde hľadanie elipsy viedlo na rovnaký výsledok ako hľadanie kružnice (pre lepšiu viditeľnosť sú farby invertované) . . . . .	49
4.31	Nežiadané preferovanie lepšie vyhovujúcej ale neúplnej elipsy, keď bola kružnica nájdená správne (pre lepšiu viditeľnosť sú farby invertované) . . . . .	50
4.32	Prípad, kde hľadanie elipsy viedlo na lepší výsledok ako hľadanie kružnice (pre lepšiu viditeľnosť sú farby invertované) . . . . .	50
4.33	Ukážka závislosti úspešnosti na parametri, kde existuje nápadná optimálna hodnota . . . . .	51
4.34	Ukážka závislosti úspešnosti na parametri, kde je optimálna hodnota viditeľne v nejakom intervale . . . . .	52

4.35	Ukážka závislosti úspešnosti na parametri, keď má konkrétna hodnota parametra minimálny vplyv na úspešnosť . . . . .	52
5.1	Lahko odhaliteľná chyba (pre lepšiu viditeľnosť sú farby invertované)	55
5.2	Ťažko odhaliteľná chyba 1 (pre lepšiu viditeľnosť sú farby invertované)	55
5.3	Ťažko odhaliteľná chyba 2 (pre lepšiu viditeľnosť sú farby invertované)	55
6.1	Ukážky fotografií z testovania na veľkej vzorke (pre lepšiu viditeľnosť sú kvapky orezané) . . . . .	61
7.1	Úspešne vypočítaný kontaktný uhol na fotografii kvapky s malým kontaktným uhlom a neviditeľným okrajom podložky (pre lepšiu viditeľnosť sú farby invertované) . . . . .	62
7.2	Úspešne vypočítaný kontaktný uhol na fotografii kvapky s množstvom odrazov okolitých svetiel (pre lepšiu viditeľnosť sú farby invertované)	63
7.3	Úspešne vypočítaný kontaktný uhol na fotografii s viacerými kvapkami (pre lepšiu viditeľnosť sú farby invertované) . . . . .	63
7.4	Úspešne ale nepresne vypočítaný kontaktný uhol na fotografii s kvapkou nie celkom kruhového tvaru (pre lepšiu viditeľnosť sú farby invertované) . . . . .	64
7.5	Neúspešné spracovanie neostrej fotografie (pre lepšiu viditeľnosť sú farby invertované) . . . . .	64
7.6	Neostrá fotografia, ktorej kontaktný uhol bol vypočítaný vďaka záložnému algoritmu (pre lepšiu viditeľnosť sú farby invertované) . . . .	64
7.7	Neúspešné spracovanie fotografie kvapky, pri na ktorej sa našla hrana nad koncom oblúka (pre lepšiu viditeľnosť sú farby invertované) . . .	65
7.8	Neúspešné spracovanie fotografie kvapky, ktorá sa odrážala od substrátu (pre lepšiu viditeľnosť sú farby invertované) . . . . .	65
7.9	Príklad fotografie kvapky, ktorej kontaktný uhol sa podarí vypočítať iba s istou pravdepodobnosťou (pre lepšiu viditeľnosť sú farby invertované) . . . . .	66

# Zoznam tabuliek

1.1	Parametre bežne používaných kvapalín . . . . .	15
4.1	Rýchlosť výpočtu rôznych implementácií odmocniny (100% je rýchlosť štandardnej implementácie) a úspešnosť celého algoritmu pri použití tejto implementácie . . . . .	38
4.2	Zoznam doladovaných parametrov . . . . .	53
5.1	Úspešnosti jednotlivých kritérií a ich váhy pri hlasovaní . . . . .	58
5.2	Hodnotenie výsledku podľa súčtu hlasov s váhami zo sady D . . . . .	58
5.3	Hodnotenie výsledkov pri úspešnom spracovaní a pri neúspešnom spracovaní, v percentách z celku . . . . .	59

# Úvod

Hlavnou metódou určovania povrchového napätia povrchov je meranie kontaktných uhlov kvapiek na týchto povrchoch. Používajú sa rôzne kvapaliny s presne známym povrchovým napätím, napríklad destilovaná voda alebo dijódometán. Metóda spočíva v umiestnení drobnej kvapky kvapaliny na meraný povrch, jej odfotozobrazením a určením kontaktného uhlu z fotografie.

Vytvoriť algoritmus, ktorý z fotografie určí kontaktný uhol je už dávno vyriešené, ak sú vhodné svetelné podmienky. Avšak dosiahnutie správneho osvetlenia zvyšuje cenu meracieho zariadenia. Bez neho sa na fotografiách kvapiek objavujú odrazy blízkych predmetov a svetiel, kvapka máva podobnú farbu ako okolie, kvapka sa odráža od meraného povrchu alebo v pozadí sa môžu vyskytovať neostre objekty.

Cieľom tejto práce bolo napísať algoritmus, ktorý určí kontaktný uhol z fotografie kvapky aj za menej priaznivých svetelných podmienok. Pre minimalizáciu ceny algoritmus nesmie používať knižnice, ktoré nie sú voľne šíriteľné. Algoritmus by mal byť dostatočne rýchly na to, aby naň nemusela obsluha čakať. Tiež by pre jednoduchosť zapracovania do hlavného programu algoritmus mal používať čo najmenej knižníc.

Práca sa začína pojednaním o povrchovom napätí a problematike merania kontaktných uhlov. Nasleduje popis algoritmov používaných pre hľadanie kružníc na fotografiách. Ďalšia kapitola sa venuje zbierke fotografií s manuálne určenými vzorovými výsledkami používaných pre testovanie programu. Po nej nasleduje časť o tvorbe algoritmu a skúšaní rôznych metód. Ďalšia kapitola sa venuje algoritmu pre kontrolu správnosti výsledku. Po nej nasleduje kapitola o testovaní algoritmu na veľkej sade fotografií, teraz už používajúc automatizovanú kontrolu správnosti. Posledná časť hodnotí úspešnosť algoritmu na rôznych typoch fotografií.

Vytvorený algoritmus mal pomerne vysokú úspešnosť, približne 80%, a ak boli fotografie dostatočne kvalitné, tak sa úspešnosť blížila k 100%. Vďaka možnosti spracovávať kontaktné uhly v reálnom čase je možné, aby experimentátor dosiahol vhodné podmienky a potom začal s meraním.

# 1 Povrchové napätie

Povrchové napätie je jav tvorený kohéznymi silami medzi molekulami kvapaliny alebo pevnej látky. Molekuly na povrchu telesa nemajú na všetkých stranách ďalšie molekuly, preto je medzi nimi zosilnená kohézna sila. Tento jav v prípade kvapalín spôsobí, že povrch odoláva vplyvu vonkajších síl. To umožňuje drobným predmetom neponoriť sa do kvapaliny, ak ich tiaž je príliš slabá na to, aby oddelili molekuly povrchu od seba.

Tento jav vysvetľuje napríklad ako dokáže malý hmyz chodiť po vodnej hladine, prečo sú bubliny guľaté, ako mydlo čistí povrchy alebo prečo majú kvapky svoj typický tvar.[3]

## 1.1 Kontaktné uhly

Ak je medzi kvapalinou a pevnou látkou dotyk, uhol medzi povrchom kvapaliny a povrchom pevnej látky popisuje kontaktný uhol  $\theta$ . Kontaktný uhol je mierou zmáčavosti pevnej látky kvapalinou.

Dostatočne malé kvapky majú tvar veľmi podobný časti gule, pretože sily povrchového napätia výrazne prevyšujú tiaž. Pri meraní kontaktných uhlov sa nepoužívajú väčšie kvapky.

Ak je kontaktný uhol povrchu s vodou menší ako  $90^\circ$ , hovoríme, že povrch je hydrofilný, v opačnom prípade hovoríme, že je hydrofóbny. Teoretické maximum je  $180^\circ$ . Ak je kvapalina úplne rozliata po povrchu a kvapky vôbec nevytvára, kontaktný uhol je  $0^\circ$ .

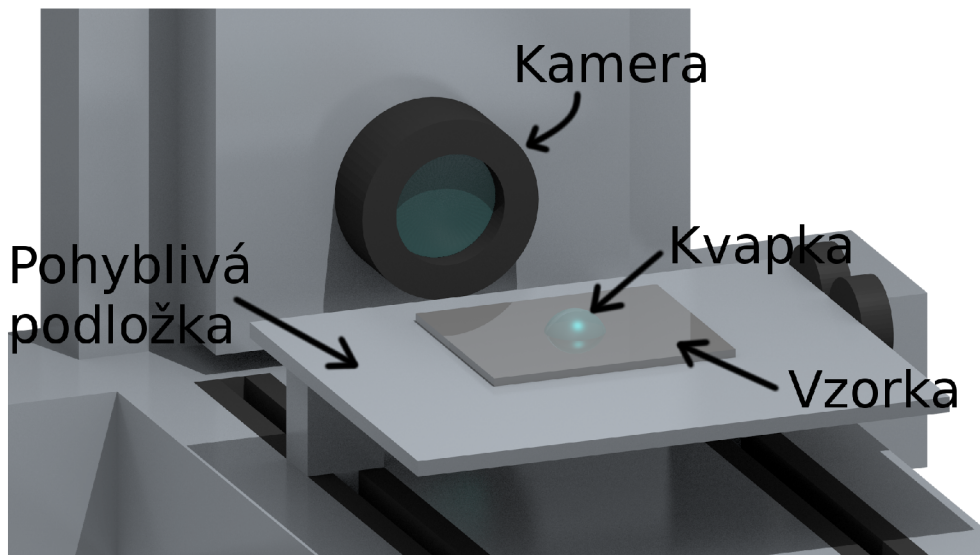
Vzťah medzi povrchovými napätiami a kontaktným uhlom popisuje Youngova rovnica [4]:

$$\sigma_s = \sigma_{sl} + \sigma_l \cdot \cos \theta, \quad (1.1)$$

kde  $\sigma_s$  je povrchové napätie pevnej látky (v  $\text{J}\cdot\text{m}^{-2}$ ),  $\sigma_l$  je povrchové napätie kvapaliny a  $\sigma_{sl}$  je napätie rozhrania.

Na určenie kontaktného uhla látky sú zvyčajne použité kvapky viacerých kvapalín s dobre známym povrchovým napätím, kde je kontaktný uhol vypočítaný napríklad pomocou lineárnej regresie medzi výsledkami získanými s rôznymi kvapalinami.

Kontaktný uhol je užitočné poznať pri aplikáciách vyžadujúcich kontakt kvapaliny s pevnou látkou, napríklad náteroch, čistení, hydrofóbnych alebo hydrofilných úpravách povrchov materiálov, tlačí... Aplikácia merania kontaktných uhlov na určenie povrchového napätia je do hĺbky popísaná v [1].



Obr. 1.1: Schématické znázornenie zariadenia SeeSystem

## 1.2 Zariadenie na meranie kontaktných uhlov SeeSystem

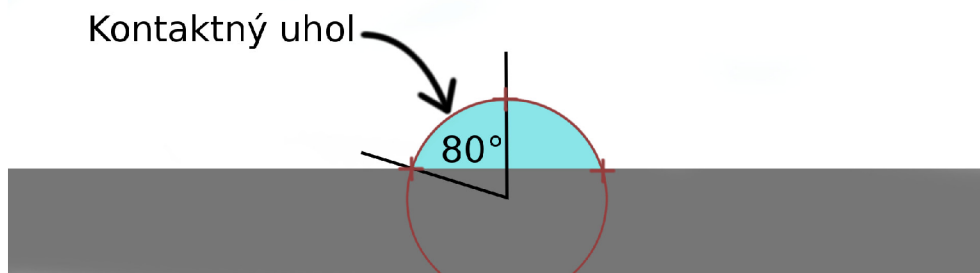
SeeSystem (skratka *Surface Energy Evaluation System*) je prenosný nástroj pre meranie kontaktného uhla a určovanie povchovej energie pevných látok. Je tiež použiteľný pre určovanie samočistiacich schopností fotokatalytických materiálov.

Zariadenie je používané na meranie kontaktných uhlov kvapiek na vzorkách položených na podložke polohovateľnej v dvoch osiach pomocou 2 Mpix UVC kamery. Kameru je tiež možné polohovať. Zariadenie využíva denné svetlo namiesto vlastného zdroja svetla, čo pomáha riešiť problém s vyparovaním kvapalín spôsobeným vytvoreným teplom. Obraz z kamery je prenášaný cez USB 2.0 port do počítača, v ktorom beží ovládací softvér. Zariadenie je schématicky zakreslené na obrázku 1.1.

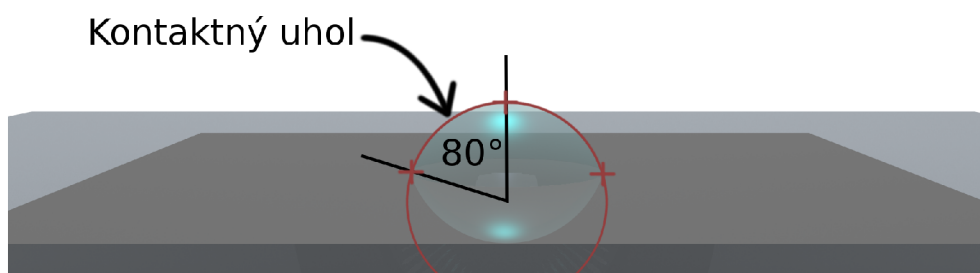
Doteraz používaný softvér umožňuje používateľovi odfotografovať kvapku, zobrazí ju a prakticky na nej označiť oblúk kvapky, z čoho je vypočítaný kontaktný uhol a pomocou rôznych modelov je z neho vypočítané povrchové napätie.

K nájdeniu kontaktného uhla na fotografii je potrebné označiť na nej tri body, dva na miestach, kde oblúk kvapky končí a dotýka sa povrchu a jeden na ľubovoľnom mieste na oblúku kvapky (znázornené na obrázkoch 1.2 a 1.3). Pretože tri body jednoznačne definujú kružnicu a dva body jednoznačne definujú priamku, stačí to na výpočet kontaktného uhla [5].

Nedostatkom zariadenia je potreba manuálne vyznačiť kvapku, čo zaberá netriviálnu časť času stráveného meraním.



Obr. 1.2: 2D znázornenie princípu určovania kontaktného uhla z fotografie



Obr. 1.3: 3D znázornenie určovania kontaktného uhla z fotografie

## 1.3 Výpočet povrchovej energie

Kvapka sa ustáli v stave s najmenšou povrchovou energiou. Zmenu povrchovej energie  $\Delta E$  možno vyjadriť nasledovne:

$$\Delta E = \delta \cdot \Delta A, \quad (1.2)$$

kde  $\delta$  je povrchové napätie (v  $\text{J}\cdot\text{m}^{-2}$ ) a  $\Delta A$  rozdiel veľkosti povrchu (v  $\text{m}^{-2}$ ). Rovnováhu medzi povrchovým napätím a trojfázovým rozhraním kvapky popisuje Youngova rovnica [6]:

$$\sigma_{BC} - \sigma_{AB} = \sigma_{AC} \cdot \cos\Phi. \quad (1.3)$$

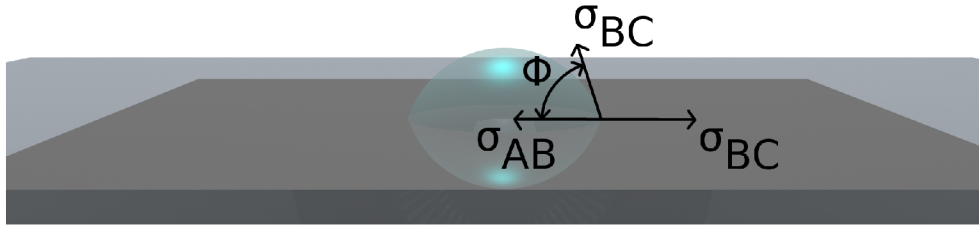
Geometrický význam povrchových napätí  $\sigma_{BC}$ ,  $\sigma_{AB}$  a  $\sigma_{AC}$  a uhla  $\Phi$  je na obrázku 1.4.

Určenie povrchového napätia využíva rôzne modely vyplývajúce z rôznych teórií.

### 1.3.1 Zismanova teória

Najstaršou teóriou je Zismanova teória. Používa empiricky zistenú lineárnu závislosť kosínu kontaktného uhla na povrchovej energii. Používajú sa výsledky niekoľkých meraní s rôznymi povrchovými energiami. Nájdené body preložíme lineárnym fitom





Obr. 1.4: Trojfázové rozhranie kvapky

$f(\sigma)$  a podľa neho nájdeme hodnotu  $\sigma_{krit}$  takú, že  $f(\sigma_{krit}) = 1$ . Hodnota  $\sigma_{krit}$  sa považuje za povrchové napätie pevnej látky [7].

### 1.3.2 Neumannova teória

Neumannova, alebo Li-Neumannova teória, je rozšírením Berthelotovej teórie o exponenciálny faktor [8]:

$$\sigma_{sl} = \sigma_{lg} + \sigma_{sg} - 2\sqrt{\sigma_{lg}\sigma_{sg}}e^{-0.0001247(\sigma_{lg}-\sigma_{sg})^2}, \quad (1.4)$$

kde  $\sigma_{sl}$  je povrchové napätie medzi pevnou látkou a kvapalinou,  $\sigma_{lg}$  je povrchové napätie medzi a vzduchom a  $\sigma_{sg}$  je povrchové napätie medzi pevnou látkou a vzduchom.

Pridaním tejto rovnice do Young-Dupreho rovnice vznikne vzťah [6]:

$$\cos \Phi = -1 + 2\sqrt{\frac{\sigma_{lg}}{\sigma_{sg}}}e^{-0.0001247(\sigma_{lg}-\sigma_{sg})^2}. \quad (1.5)$$

Povrchové napätie získame hľadaním parametrov  $\sigma_{lg}$  a  $\sigma_{sg}$  metódou najmenších štvorcov.

### 1.3.3 Teória OWRK

Teória OWRK, alebo Owens-Wendtova teória predpokladá, že voľná povrchová energia je súčet disperzného a polárneho komponentu [9]. Medzifázová energia je vyjadrená ako ich geometrický priemer:

$$\sigma_{sl} = \sigma_s + \sigma_l - 2(\sqrt{\sigma_s^d\sigma_l^d} + \sqrt{\sigma_s^p\sigma_l^p}), \quad (1.6)$$

kde  $\sigma_{sl}$  je povrchové napätie medzi pevnou látkou a kvapalinou,  $\sigma_s$  je povrchové napätie pevnej látky,  $\sigma_l$  je povrchové napätie kvapaliny,  $\sigma_s^d$  je disperzná zložka povrchového napätia pevnej látky,  $\sigma_l^d$  je disperzná zložka povrchového napätia kvapaliny,

$\sigma_s^p$  je polárna zložka povrchového napätia pevnej látky a  $\sigma_s^d$  je polárna zložka povrchového napätia kvapaliny.

Dosadením do Youngovej rovnice dostaneme vzťah [6]:

$$\sigma_l(1 + \cos \Theta) = -2(\sqrt{\sigma_s^d \sigma_l^d} + \sqrt{\sigma_s^p \sigma_l^p}), \quad (1.7)$$

kde  $\Theta$  je nameraný uhol. Na výsledky meraní viacerých kvapalín použijeme lineárnu regresiu.

### 1.3.4 Acidobázická teória

Acidobázická teória vychádza z acidobázických interakcií [6]. Používa nepolárnu zložku označovanú  $LW$  a polárnu zložku označovanú  $AB$  (interakcia donorov a akceptorov  $\sigma^{AB} = 2\sqrt{\sigma^+ \sigma^-}$ , kde  $\sigma^{AB}$  je polárna zložka povrchového napätia,  $\sigma^+$  je povrchové napätie od donorov a  $\sigma^-$  je povrchové napätie od akceptorov). Regresný model má tvar:

$$\frac{1 + \cos \Phi}{2} \cdot \frac{\sigma_l}{\sqrt{\sigma_l^{LW}}} = \sqrt{\sigma_s^{LW}} + \sqrt{\sigma_s^+} \cdot \sqrt{\frac{\sigma_l^-}{\sigma_l^{LW}}} + \sqrt{\sigma_s^-} \cdot \sqrt{\frac{\sigma_l^+}{\sigma_l^{LW}}}, \quad (1.8)$$

kde  $\Phi$  je kontaktný uhol,  $\sigma_l$  je povrchové napätie kvapaliny,  $\sigma_l^{LW}$  je nepolárna zložka povrchového napätia kvapaliny,  $\sigma_s^{LW}$  je nepolárna zložka povrchového napätia pevnej látky,  $\sigma_l^-$  je povrchové napätie od akceptorov kvapaliny a  $\sigma_s^-$  je povrchové napätie od akceptorov pevnej látky.

Na určenie povrchového napätia potrebujeme aspoň tri kvapaliny so známymi všetkými tromi zložkami povrchového napätia, z čoho musia byť aspoň dve polárne.

## 1.4 Kvapaliny používané pri meraní povrchového napätia

Pri meraní povrchového napätia pomocou kontaktných uhlov je zvyčajne používaná tá istá skupina kvapalín so známymi potrebnými parametrami, kvôli množstvu rôznych parametrov, ktoré treba o nich poznať [10]. So SeeSystemom je bežne používaná voda, dijódmetán, formamid, glycerol a ethylénglykol [6]. Ich dôležité parametre sú uvedené v tabuľke 1.1.

V niektorých prípadoch, zvlášť pri disperznom a polárnom povrchovom napätí, rôzne zdroje uvádzajú rôzne parametre. V takýchto prípadoch bola vybratá najčastejšia hodnota.

kvapalina	povrchové napätie [mJ/m <sup>2</sup> ]				
	celkové ( $\sigma_l$ )	disperzné ( $\sigma_l^d$ )	polárne ( $\sigma_l^p$ )	kyselinové ( $\sigma_l^+$ )	zásadité ( $\sigma_l^-$ )
voda	72.8	22	51	25.5	25.5
dijódometán	50.8	48	2	0.01	0.0
formamid	58	39	19	2.28	39.6
glycerol	64	37	26	3.92	57.4
etylénglykol	48	29	19	3.0	30.1

Tab. 1.1: Parametre bežne používaných kvapalín

## 1.5 Zhrnutie

Povrchové napätie látky je možné určiť pomocou jednej z popísaných metód, pričom ako vstupné dáta slúžia namerané kontaktné uhly rôznych kvapalín na tomto povrchu a ich fyzikálne vlastnosti.

## 2 Metódy používané pre detekciu kvapky na fotografii

Spracovávanie obrazového signálu je v súčasnosti veľmi rozšírenou oblasťou spracovania dát. Na riešenie mnohých úloh sú známe vysoko úspešné algoritmy, popísané napríklad v [2]. Táto kapitola popisuje iba metódy používané na hľadanie kvapiek na fotografii.

### 2.1 Filtrovanie

Obraz získaný napríklad z kamery je zväčša vo forme nevhodnej na priame použitie algoritmov. Typickými problémami sú šum, variácie v osvetlení alebo slabý kontrast. Tieto problémy treba vyriešiť na zlepšenie kvality obrazu a jeho úpravu do formy vyžadovanej algoritmi [11].

Filtre uvedené v tejto podkapitole sú aplikované na obrázok 2.1.



Obr. 2.1: Pôvodný obrázok (foto: autor)

#### 2.1.1 Gaussov filter

Gaussov filter je používaný na rozostrenie obrázkov a odstránenie šumu a detailov. V dvoch rozmeroch je definovaný funkciou [12]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.1)$$



Obr. 2.2: Obrázok po aplikácii Gaussovho filtra

kde  $\sigma$  je štandardná odchýlka a  $x$  a  $y$  sú súradnice pixelov.

Táto funkcia je bežne používaná na potlačenie náhodných odchýlok v dvoch rozmeroch.

Táto funkcia má hodnotu všade väčšiu ako 0, avšak jej hodnota rýchlo klesá s rastúcou vzdialenosťou od stredu, pri veľkom  $\sigma$  klesá pomalšie, pri malom  $\sigma$  klesá prudšie. 99% objemu pod funkciou leží do vzdialenosti  $3\sigma$ , takže hodnoty ležiace vo väčšej vzdialenosti môžeme zanedbať.

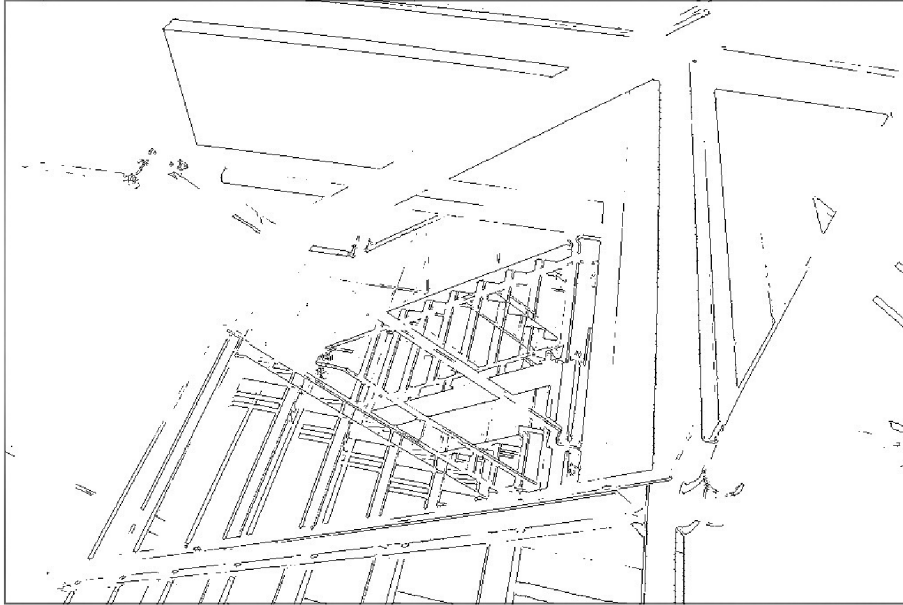
### 2.1.2 Gradientné operátory

Na samotné nájdenie hrán sa používa napríklad Prewittovej operátor alebo Sobelov operátor. Prewittovej operátor spočíva v konvolúcii obrázka s maticami:

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \text{ a } \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.2)$$

Výsledkom sú jasové rozdiely v obrázku v horizontálnom smere pri použití prvej matice a rozdiely vo vertikálnom smere pri použití druhej matice. [13]

Častejšie sa používa Sobelov operátor, ktorý aplikuje konvolúciu s inými mati-



Obr. 2.3: Obrázok po aplikácii Sobelovho operátora a prahovania

camí [14]:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ a } \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.3)$$

Výsledkom sú jasové rozdiely v horizontálnom smere (vertikálnom smere v prípade druhej matice). Po nájdení rozdielov v rôznych smeroch je nájdená celková veľkosť gradientu podľa vzorca:

$$G = \sqrt{G_x^2 + G_y^2}, \quad (2.4)$$

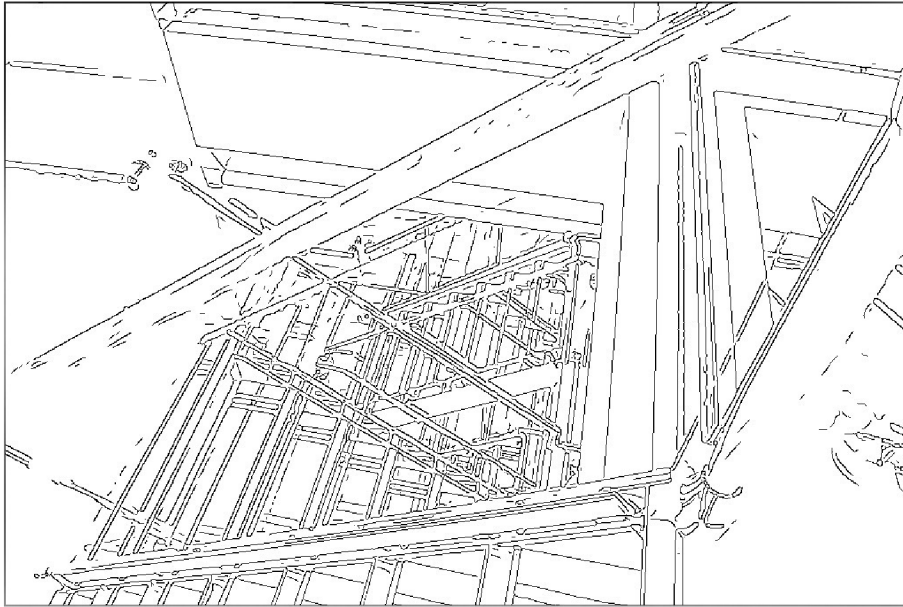
kde  $G_x$  je hodnota rozdielu v smere  $x$  a  $G_y$  je hodnota rozdielu v smere  $y$ .

Smer zmeny jasú určíme podľa vzorca:

$$G = \arctan \frac{G_y}{G_x}. \quad (2.5)$$

### 2.1.3 Binárne prahovanie

Binárne prahovanie je operátor, ktorého výsledkom je 0 na mieste, kde bola hodnota jasú pixelu pod prahovou hodnotou a 1 na mieste, kde bola nad prahovou hodnotou. Práhová hodnota je zvyčajne získaná z histogramu obrázku na začiatku tohoto kroku.



Obr. 2.4: Cannyho detektor

Často je používané aj prahovanie s hysteréziou, kde sa hodnota výsledných pixelov nastaví na 1 tam, kde je hodnota vyššia ako vyšší prah a potom sa rekurzívne nastaví na 1 aj vo všetkých bodoch s hodnotou vyššou ako nižší prah, ak susedia s nejakým bodom s hodnotou 1.

#### 2.1.4 Cannyho detektor

Cannyho detektor je viackrokový algoritmus na detekciu hrán a potlačanie šumu. Pozostáva z nasledujúcich krokov:

1. Aplikácia Gaussovho filtra na obrázok na potlačenie vplyvu nechcených detailov a šumu (pôvodná fotografia 2.1, výsledok 2.2)
2. Vypočítanie gradientu pomocou gradientných operátorov
3. Aplikácia binárneho prahovania (obrázok 2.3)
4. Potlačenie pixelov, ktoré nie sú maximami gradientov
5. Prahovanie s dvoma rôznymi prahmi tak, aby vznikli dva binárne obrázky
6. Pridávanie pixelov z obrázku s nižším prahom, ktoré s jeho pixelmi susedia dovedy, kým sa niečo mení, do obrázku s vyšším prahom (obrázok 2.4) [15]

## 2.2 Metódy hľadania oblúkov

Hľadanie oblúkov či už kruhových alebo všeobecne eliptických, má široké využitie v mnohých oblastiach, ako sledovanie ľudských pohybov, triedenie semienok, seg-

mentovanie buniek v biológii, hľadanie slnečnej koróny v astronómii, atď. Výpočtová zložitosť naivného riešenia je príliš veľká, pretože kružnica má tri voľné parametre (dve súradnice stredu a polomer) a elipsa päť (dve súradnice stredu, dve poloosi, orientácia). Existuje množstvo efektívnych metód pre hľadanie elíps, s rôznymi výhodami a nevýhodami.

### 2.2.1 Metódy najmenších štvorcov

Metódy najmenších štvorcov (LS, least-square) vo všeobecnosti minimalizujú súčet kvadrátov nejakých rozdielov. V tomto prípade je kritérium nastavené tak, aby jeho minimom bol útvar prechádzajúci čo najbližšie k bodom, o ktorých sa predpokladá, že sú na hľadanom útvere.

Pri aplikácii na detekciu kružníc alebo elíps sú zvyčajne rýchle, ale vyžadujú kvalitný odhad počiatočného bodu, sú citlivé na body ležiace ďaleko mimo útvaru, šum a sú málo efektívne, ak je útvar neúplný alebo sa na ňom nachádzajú odrazy. Používané metódy sú [16] [17]:

- DLS (direct least-square) - dokáže nájsť elipsu bez iterovania, eliptické podmienky sú v normalizačnom faktore, dokonalý útvar vedie k singularitám vo výpočte
- Enhance LS - upravené DLS, mierne pomalšie, ale bez problémov so singularitami
- Hyper LS - oveľa lepšie ako predchádzajúce, ale vyžaduje aspoň 1/4 celého oblúku elipsy
- Geometrické fitovanie - zvyčajne sa hľadajú parametre získané nejakou transformáciou, aby zmyslupnejšie popisovali krivky a konvergovali rýchlejšie

### 2.2.2 Metódy založené na hlasovaní

Základnou metódou založenou na hlasovaní je Houghova transformácia. Okrem nej sa používajú najmä rôzne jej úpravy.

#### Houghova transformácia

Houghova transformácia (HT) je postup pre izoláciu prvkov konkrétneho tvaru v obrázku [18]. Ak je možné tvar vyjadriť parametricky, hovoríme o klasickej Houghovej transformácii. Inak hovoríme o zovšeobecnenej Houghovej transformácii. Ďalej sa bude hovoriť iba o klasickej Houghovej transformácii. Houghova transformácia je použiteľná v mnohých aplikáciách, kde sa nachádzajú hrany popísateľné krivkami. Jej veľkou výhodou je tolerancia medzier v okrajoch a malý problém so šumom.



Pri Houghovej transformácii je obrázok najprv predspracovaný Cannyho detektorom. Potom je pre všetky kombinácie parametrov krivky vypočítaná priemerná hodnota pixelov ležiacich na krivke.

V prípade hľadania priamky sa používa rovnica:

$$x \cos \theta + y \sin \theta = r, \quad (2.6)$$

kde  $x$  a  $y$  sú súradnice bodov tejto priamky a  $r$  a  $\theta$  sú parametre priamky, poloha a náklon.

V prípade hľadania kružníc sa používa rovnica:

$$(x - a)^2 + (y - b)^2 = r^2, \quad (2.7)$$

kde  $x$  a  $y$  sú súradnice bodov tejto kružnice a  $r$ ,  $a$  a  $b$  sú parametre kružnice, polomer a súradnice stredu.

Pretože v obraze Houghovej transformácie nás zaujímajú iba maximá a v blízkosti maxim Houghovej transformácie sú tiež zvýšené hodnoty, je možné výpočet zrýchliť získaním počiatočného odhadu a posunu do maxima po gradiente.

## Úpravy Houghovej transformácie

Pretože potrebný počet výpočtov pre vypočítanie Houghovej transformácie rastie s počtom parametrom exponenciálne, metóda je vhodná pre hľadanie priamok (2 parametre), ale menej vhodná pre hľadanie kružníc (3 parametre) alebo elíps (5 parametrov) [16] [19]. Houghova transformácia je ale veľmi výhodná kvôli svojej robustnosti, preto sa používajú rôzne metódy na zníženie jej výpočtovej náročnosti:

- Zovšeobecnená HT (GHT) - redukuje dimenziu priestoru parametrov tým, že ho delí na menšie kúsky nižšieho rádu
- Priamočiara HT (SLHT) - vypočíta Houghovu transformáciu obrazu na priamky a v nich hľadá prejavy eliptických útvarov
- Rýchla elipsová HT (FEHT) - používa dopytový proces na rekurzívnom zaostracom algoritme, dokáže nájsť veľký počet elíps rôznych veľkostí
- Náhodná HT (RHT) - náhodne vyberie pixely, s ktorými bude ďalej počítať a počíta iba v týchto pixeloch
- Eliptická HT - kombinuje obrázok s pyramídovou dátovou štruktúrou s Houghovou transformáciou, postupne zvyšuje rozlíšenie a počíta iba v okolí kandidátov z predchádzajúceho rozlíšenia; kvôli nejasnému názvu bude táto metóda ďalej nazývaná *viacstupňová Houghova transformácia*

### 2.2.3 Ďalšie metódy hľadania kružníc

Okrem metód založených na najmenších štvorcoch alebo Houghovej transformácie existuje aj množstvo iných metód [16]:

- Rozšírený Kalmanov Filter (EKF) - je rovnako závislý na počiatočnom odhade ako lineárne metódy, avšak existuje metóda na odstránenie tohoto problému
- Konsenzus náhodnej vzorky (RANSAC) - rozdelí obrázok na malé podmnožiny a z každej určí najpravdepodobnejšiu množinu útvarov, potom sú vybrané tie s najväčším výskytom (zohľadní sa pritom počet úsekov, ktoré sa v ich blízkosti vyskytli); má problémy nájsť väčšie počty útvarov a je náchylný na šum, existujú variácie, ktoré tieto obmedzenia redukovujú
- Genetický algoritmus - biológiiu inšpirovaný algoritmus simulujúci evolúciu pomocou mutácie, kríženia a výberu
- Rast elíps - nájde kružnice, postupne ich zväčšuje do elíps, kým je nájdená hrana, pomalé
- Metódy sledujúce hranu - nájdu hranu, snažia sa ju sledovať a pri jej sledovaní spresňovať parametre, často potrebujú celý útvar
- Hybridné metódy - kombinujú viaceré metódy, často majú lepšie výsledky ako jednotlivé metódy - napríklad použitie Houghovej transformácie na odhad hodnôt a následne použitie metódy najmenších štvorcov

## 2.3 Zhuková analýza

Zhluková analýza je proces zoskupenia súboru objektov do zhukov tak, že objekty v tom istom zhuku sú si navzájom podobnejšie ako s prvkami iných zhukov [20]. Tento proces je dôležitý v rôznych oblastiach ako strojové učenie, rozpoznávanie vzorov v obraze, spracovanie obrazu...

Existuje množstvo metód pre zhukovú analýzu, uvedená je iba tá, ktorá je použitá v algoritme, Unweighted Pair Group Method with Algorithmic Mean (UPGMA).

Algoritmus UPGMA [21]:

1. V každom bode zdefinovať zhuk
2. Pokiaľ je zhukov viac, ako je žiadané (argument funkcie)
  - (a) Vypočítať polohy stredov zhukov (priemery súradníc)
  - (b) Vypočítať vzdialenosti medzi stredmi zhukov (podľa zvolenej metriky)
  - (c) Nájsť dva zhuky, ktorých stredy sú najbližšie k sebe
  - (d) Zlúčiť tieto dva zhuky

## 2.4 Zhrnutie

Pri hľadaní kružníc na fotografii sa fotografia najprv upraví tak, aby sa našli hrany. Na hľadanie hrán sa používa hlavne Cannyho detektor pozostávajúci z potlačenia šumu, operátora na vyznačenie rozdielov a prahovania s hysteréziou. Kružnicu je

možné spoľahlivo nájsť pomocou Houghovej transformácie, ktorá je však príliš výpočtovo náročná a používajú sa namiesto nej rôzne menej spoľahlivé, ale rýchlejšie metódy, zvyčajne založené na Houghovej transformácii alebo metóde najmenších štvorcov. Ak je výsledkom veľký počet potenciálnych kružníc, podobné je možné zlúčiť pomocou zhlukovej analýzy.

### 3 Sada snímok pre overenie funkčnosti navrhovaných algoritmov

Pri hľadaní prvého použiteľného algoritmu bola používaná fotografia jednej kvapky. Bola vybraná tak, aby sa jednoducho spracovávala pri tvorbe prvých prototypov a nebola ničím neobvyklá (3.1).



Obr. 3.1: Prvá fotografia používaná pri tvorbe algoritmu

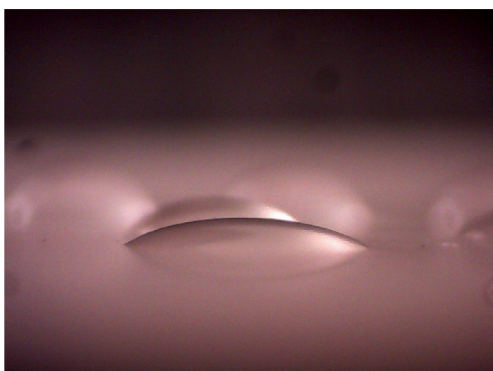
Po vytvorení funkčného prototypu algoritmu bol používaný adresár s 27 fotografiami kvapiek a úspešnosť hľadania kvapky bola posudzovaná ručne. Tieto fotografie boli vybrané z výsledkov rôznych meraní tak, aby boli čo najrozmanitejšie. Táto metóda bola síce dobrá na analýzu prípadov, keď hľadanie kvapky zlyhalo, ale nebola vhodná na hľadanie optimálnych parametrov pre niektoré časti algoritmu.

Preto bola vytvorená sada 161 fotografií, ktorých kontaktný uhol bol určený ručne pomocou programu SeeSystem a zapísaný to tabuľky čitateľnej počítačom. Fotografie boli opäť vybrané tak, aby pochádzali z čo najrôznejších podmienok. Táto sada bola dostatočne veľká na kvantifikáciu úspešnosti algoritmu a automatizované doladovanie parametrov algoritmu. Nedostatkom tejto sady je stále nedostatočný počet fotografií, pretože drobné zlepšenia úspešnosti nebolo možné odlíšiť od náhodného úspechu na jedinej fotografii, avšak ručné spracovávanie stoviek fotografií

by bolo príliš nepraktické.

Významná časť fotografií obsahovala rôzne vady, ktoré značne komplikovali riešenie, vyžadovali rôzne filtre, znemožňovali využiť niektoré predpoklady, vyžadovali rôzne kompenzácie a podobne. Na niektorých fotografiách bolo viac kvapiek, pričom tá správna bola najostrejšia (obrázok 3.2a). Bežným problémom bol malý kontaktný uhol kvapky, kvôli čomu bola kvapka na fotografii iba malým výsekom kružnice (obrázok 3.2b). Ďalším problémom bol tvar líšiaci sa od predpokladaného (obrázok 3.2c). Tiež sa niekedy stalo, že kvapka obsahovala odraz veľmi podobný oblúku kvapky, ktorý bol výraznejší ako skutočný oblúk kvapky (obrázok 3.2d). Niekedy boli kvapky veľmi neostre (obrázok 3.2e). Vyskytovali sa aj prípady, kde sa kvapka odrážala od povrchu a odraz dotváral s kvapkou kompletný kruh (obrázok 3.2f). Na niektorých kvapkách bol veľký počet odrazov okolitých svetiel približne kruhového tvaru (obrázok 3.3a). Niekedy sa ako povrch substrátu pod kvapkou javili rôzne farebné prechody, pričom ani jeden z nich nemusel byť správny (obrázok 3.3b). Tiež sa stávalo, že sa substrát odrážal od kvapky a vytváral hranu pripomínajúcu spodný okraj kvapky (obrázok 3.3c). Aspoň jednu z týchto väd obsahovalo približne 55% fotografií v testovanej vzorke, takže nebolo vhodné sa zameriavať iba na vyhovujúce fotografie.

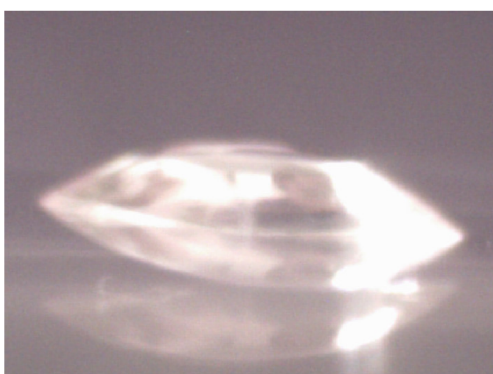
Neskôr bola vytvorená sada 11221 fotografií. Viac o nej v kapitole 6.



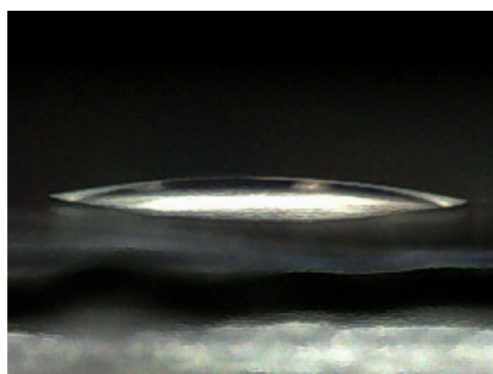
(a) Fotografia obsahujúca viac kvapiek



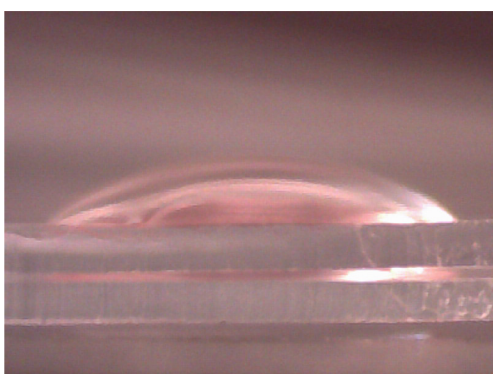
(b) Kvapka s veľmi malým kontaktným uhlom



(c) Kvapka so tvarom značne sa odchyľujúcim od predpokladaného



(d) Kvapka obsahujúca odraz podobný oblúku kvapky

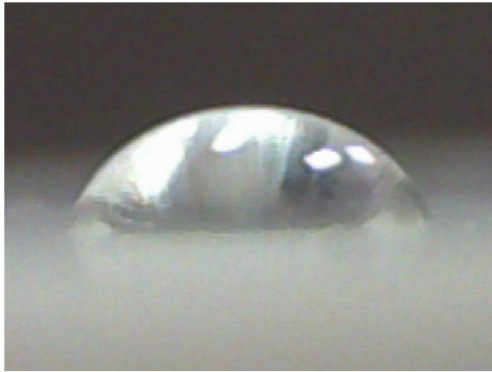


(e) Neostrá kvapka

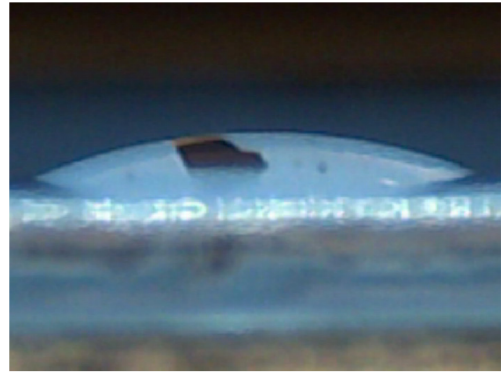


(f) Kvapka odrážajúca sa od povrchu tak, že výsledný tvar je takmer kruh

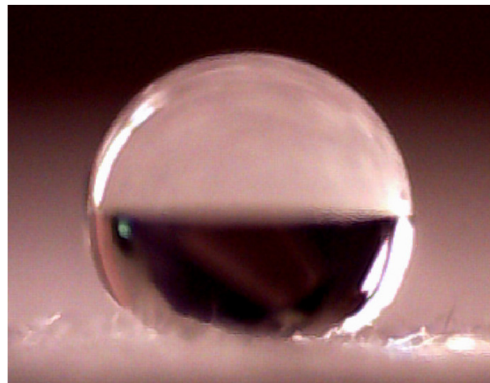
Obr. 3.2: Rôzne vady, ktoré sa vyskytovali na fotografiách



(a) Fotografia s veľkým množstvom odrazov svetiel približne kruhového tvaru



(b) Fotografia, na ktorej je viac prechodov vyzerajúcich ako povrch substrátu pod vzorkou, pričom ani jeden z nich nie je správny



(c) Kvapka, na ktorej sa odráža substrát tak, že pôsobí ako povrch substrátu pod kvapkou

Obr. 3.3: Rôzne vady, ktoré sa vyskytovali na fotografiách (pokračovanie)

## 4 Návrh a implementácia algoritmu na meranie kontaktného uhlu

Spracovávanú fotografiu kvapky na povrchu bolo treba najprv spracovať, aby bolo možné nájsť okraje kvapky. Tento krok spravidla vyžadoval odstránenie šumu a nejakú formu detekcie hrán. Z nájdených hrán bolo následne potrebné nájsť oblúk kvapky. Po nájdení polomeru a súradníc stredu bolo potrebné zistiť, kde sa tento oblúk končí. Z koncových bodov a ľubovoľného ďalšieho bodu na oblúku bolo možné priamo vypočítať kontaktný uhol.

Riešenie je teda možné rozdeliť na štyri hlavné časti:

1. Filtrovanie obrázku, aby sa ďalej dobre spracovával
2. Nájdenie kvapky na obrázku
3. Nájdenie dolného okraja kvapky
4. Vypočítanie kontaktného uhla z nájdených parametrov (počítal obslužný program zariadenia)

Program bol napísaný v C++ kvôli jeho rýchlosti a všestrannosti. MATLAB nebolo možné použiť, pretože jeho cena prevyšuje cenu celého zariadenia.

Kvôli požiadavke výrobcu zariadenia pre čo najjednoduchšiu integráciu programu nebola použitá ani bežná knižnica pre spracovanie obrazu OpenCV (viac v [22]), všetky algoritmy boli v programe reimplementované. Okrem toho sa ukázalo, že vlastná reimplementácia algoritmov bola nutná kvôli potrebe experimentovania s rôznymi modifikáciami algoritmov.

Z týchto dôvodov bolo v rámci riešenia úlohy napísaných vyše 2000 riadkov kódu v C++. Zvolené riešenie pozostáva z približne 1500 funkčných riadkov kódu. Kvôli jeho dĺžke nie je v texte uvedený použitý kód.

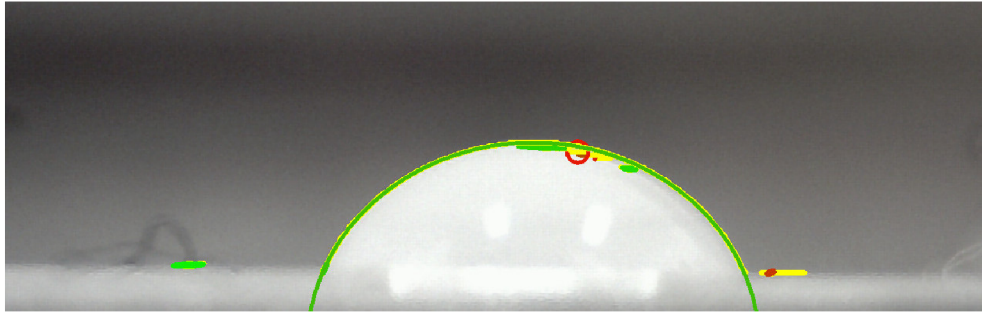
### 4.1 Pokus o použitie štandardného postupu hľadania kružníc

Hľadanie kružníc je častou úlohou počítačového videnia, preto nie je problém nájsť riešenie na Internete. Nemožnosť využitia knižníc by bolo možné obísť reimplementáciou funkcií, ktoré sa v záverečnom riešení zvolili.

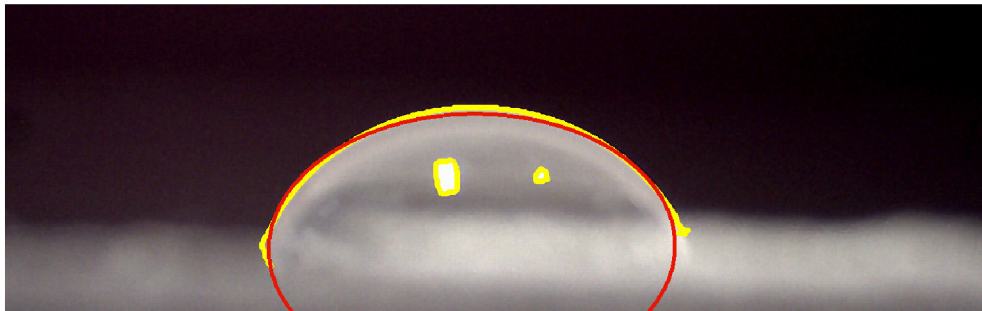
Najčastejšie používanou knižnicou počítačového videnia je OpenCV a využíva ju množstvo nájdených riešení. Ponúka širokú škálu funkcií pre prácu s obrázkami a implementácie väčšiny bežných algoritmov počítačového videnia.

Prvým vyskúšaným postupom bolo použitie prahovania a následne Houghovej transformácie. Algoritmus nebol úspešný pri analýze žiadnej z testovacích fotografií.





Obr. 4.1: Úspešné nájdenie kružnice pomocou detekcie elíps v OpenCV



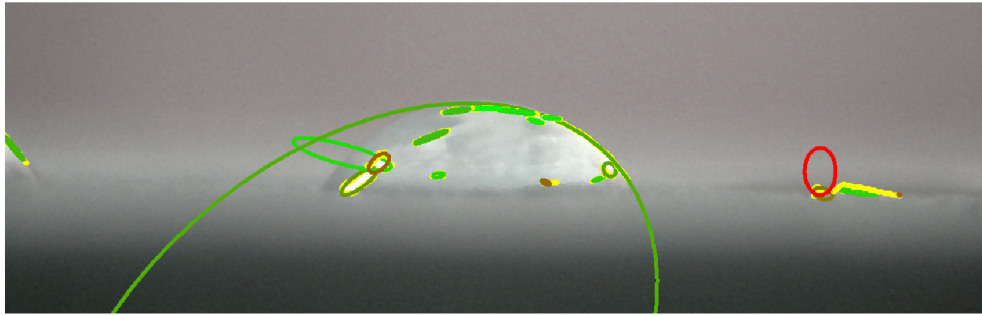
Obr. 4.2: Takmer úspešné nájdenie kružnice pomocou detekcie elíps s jednou sadou bodov v OpenCV

Použitie Cannyho operátora pred Houghovou transformáciou tiež nevedlo k úspešnému nájdeniu kvapky na niektorej fotografii. Pravdepodobnou príčinou bola optimalizácia Houghovej transformácie pre zníženie zložitosti použitá v OpenCV.

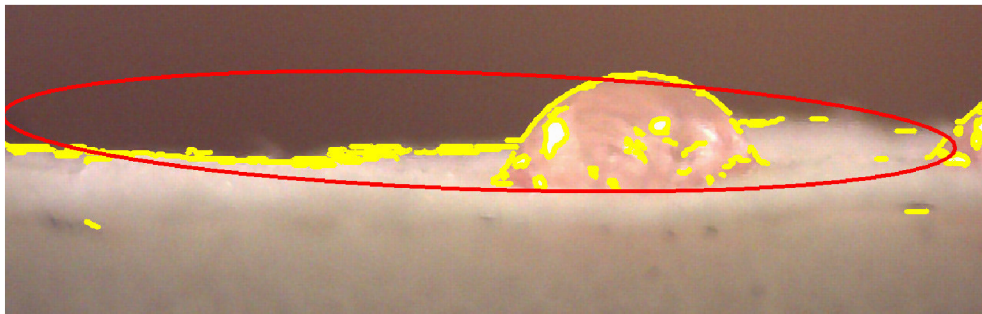
Ďalšou možnosťou bola detekcia elíps. Patričná funkcia v OpenCV pracuje s množinou bodov, pre ktoré nájde pomocou metódy najmenších štvorcov takú elipsu, ku ktorej sú body najbližšie. Body boli získané pomocou Cannyho detektora. Nie všetky body boli spojené, takže v úvahu pripadalo použitie všetkých bodov na nájdenie jednej elipsy alebo nájdenie elipsy pre každú množinu spojených bodov. Algoritmus v ojedinelých prípadoch, zväčša s veľmi ostrým oblúkom bez odrazov, viedol na správny výsledok (obrázok 4.1). Niekedy sa stávalo, že výsledok bol blízko správneho (obrázky 4.2 a 4.3), čo by pravdepodobne bolo možné korigovať, ale vo väčšine prípadov bol výsledok úplne nesprávny (obrázky 4.4 a 4.5).

## 4.2 Filtrovanie

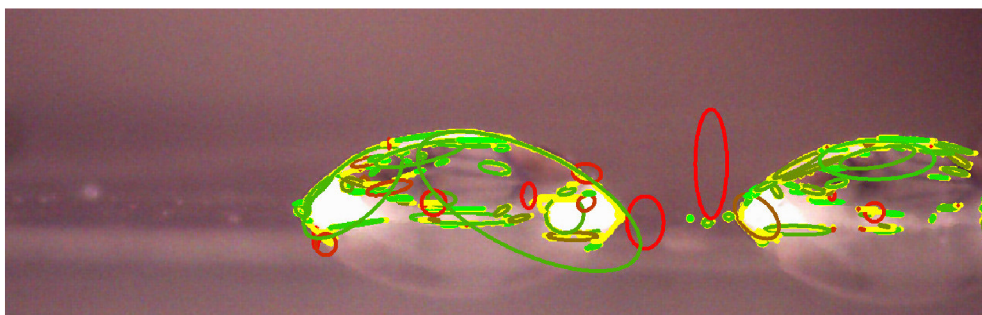
Úlohou filtrovania bolo zvýrazniť obrys kvapky a znevýrazniť iné prvky, aby nezhoršovali výsledky ďalšieho výpočtu. Pre potlačenie vplyvu nežiadúcich prvkov bol



Obr. 4.3: Takmer úspešné nájdenie kružnice pomocou detekcie elíps s viacerými sadami bodov v OpenCV



Obr. 4.4: Neúspešné hľadanie kružnice pomocou detekcie elíps s jednou sadou bodov v OpenCV



Obr. 4.5: Neúspešné hľadanie kružnice pomocou detekcie elíps s viacerými sadami bodov v OpenCV

zvolený Gaussov filter. Na zvýraznenie hrán bol nakoniec použitý Cannyho filter, boli vyskúšané však aj iné prístupy. Ďalším potrebným filtrom bolo odstránenie objektov nad kvapkou, odrazov na kvapke a odrazov kvapky na povrchu. Na tento problém bol zvolený algoritmus založený na vyplňaní oblasti.

### 4.2.1 Odstránenie šumu

Na odstránenie šumu bol spočiatku použitý Gaussov filter s konvolučnou maticou:

$$\begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix} \quad (4.1)$$

Výsledky bolo treba na konci normalizovať delením súčtom prvkov v matici. Pretože objekt záujmu sa nenachádzal na okraji obrázka, mohli sme okraj obrázka doplniť čiernymi pixelmi.

Pretože algoritmus má vysokú zložitosť,  $\mathcal{O}(h \cdot w \cdot m^2)$ , kde  $h$  je výška obrázka,  $w$  je šírka obrázka a  $m$  je veľkosť matice, jeho vykonávanie trvalo približne sekundu. Pre zrýchlenie bol rozdelený na viac vlákien, z ktorých každé dostalo inú časť obrázka.

Časom sa ukázalo, že ak sa pri výpočte neskúšal veľký počet potenciálnych kandidátov na oblúk, tento algoritmus trval zo všetkých algoritmov najdlhšie. Pre tento algoritmus existujú efektívnejšie implementácie odvodené z rýchlej Fourierovej transformácie (FFT) so zložitosťou  $\mathcal{O}(h \cdot w \cdot m)$  a jedna verzia so zložitosťou iba  $\mathcal{O}(h \cdot w)$  [23]. Ukázková implementácia bola v JavaScripte, preto bol kód preložený do C++.

Oveľa rýchlejším spôsobom odstránenia šumu bolo zmenšenie obrázka, čo však zhoršovalo presnosť výsledku a zvyšovalo pravdepodobnosť chybného nájdenia kružnice.

Nakoniec bol použitý algoritmus využívajúci FFT.

### 4.2.2 Smerové derivácie

Tento filter bol podobný podobný Prewittovej alebo Sobelovmu operátoru, avšak pre zistenie zmien vo viac ako dvoch základných smeroch. Operátor mal byť súčasne dostatočne rýchlo vypočítateľný, pretože zmeny bolo treba počítat v mnohých smeroch a čas výpočtu tohoto filtra sa výrazne prejavoval na celkovej dobe výpočtu.

Pre tento účel boli použité operátory analogické nasledujúcej matici:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.2)$$

Pri výpočte nebolo nutné násobiť maticou, stačilo iba odčítať nenulové hodnoty. Tým boli získané intenzity a smery zmien lepšie, ako Sobelovým operátorom, vďaka ktorým bolo možné odhadnúť polohy a smery dotyčníc.

Nevýhodou bola vysoká výpočtová náročnosť. Zložitosť bola  $\mathcal{O}(h \cdot w \cdot n)$ , kde  $h$  je výška obrázka,  $w$  je šírka obrázka a  $n$  je množstvo smerov. Preto bol obrázok pri tejto operácii zmenšený (čo tiež potlačilo šum).

### 4.2.3 Cannyho detektor

Cannyho detektor je algoritmus pre nájdenie hrán pozostávajúci z Gaussovho filtra, Sobelovho operátora a prahovania s hysteréziou. Viac je uvedené v kapitole 2.1.4.

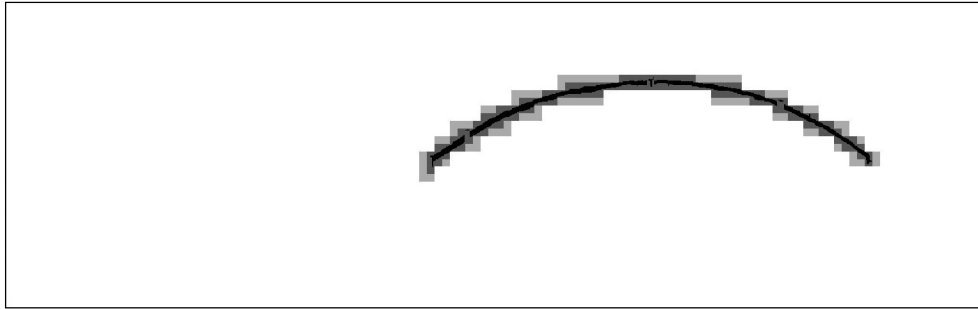
Správne prahy značne záviseli na charaktere obrázku, preto sa ich konkrétne hodnoty určovali z histogramu obrázku po aplikácii Sobelovho operátora. Ako lokálne maximum boli brané hodnoty s intenzitou zmeny väčšou ako intenzita na 99.8% plochy obrázku. Prepojené body boli akceptované, ak bola ich intenzita zmeny vyššia ako 98% alebo 95% plochy obrázku (obidve verzie boli užitočné, ako bude neskôr spomenuté).

### 4.2.4 Viacškálový Cannyho detektor

Kvôli neodstrániteľným vadám na obrázku sa často vyskytoval problém, že metódy hľadania oblúkov na základe Houghovej transformácie viazli v lokálnych minimách, aj keď odhadovaný oblúk bol značne blízko oblúka na fotografii.

Pre ich navedenie na skutočný oblúk bolo potrebné hrany nájdené Cannyho detektorom rozostriť, podobne ako Gaussovým filtrom. Bolo však potrebné ho rozostriť tak, že blízkosť hrán navádzala stred gradientnej Houghovej transformácie k hranám. Preto bol obrázok zmenšený desaťnásobne a dvadsaťnásobne (zmenšené obrázky boli

využitú aj neskôr) a následne naspäť zväčšenú (bez interpolácie). Hodnoty jasú pôvodného obrázku boli spriemerované s hodnotami dvoch takto upravených spôsobov. Výsledný obrázok je na obrázku 4.6.



Obr. 4.6: Fotografia po aplikácii viacškálového Cannyho detektora

#### 4.2.5 Vypĺňanie oblastí

Tento algoritmus tiež upravoval obrázky tak, aby sa v nich lepšie hľadali oblúky. V obrázku hľadal skupiny spojených bodov, v ktorých bola nájdená hrana (bielych bodov), a oblasti obklopené týmito bodmi zafarboval.

Algoritmus iteroval bodmi na obrázku a hľadal body nad prahom. Akonáhle takýto bod našiel, začal preskúmať okolie, kým nevytvoril zoznam bodov v celej oblasti bodov nad prahom. Takto nájdené body označil ako preskúmané a ďalej ich ignoroval.

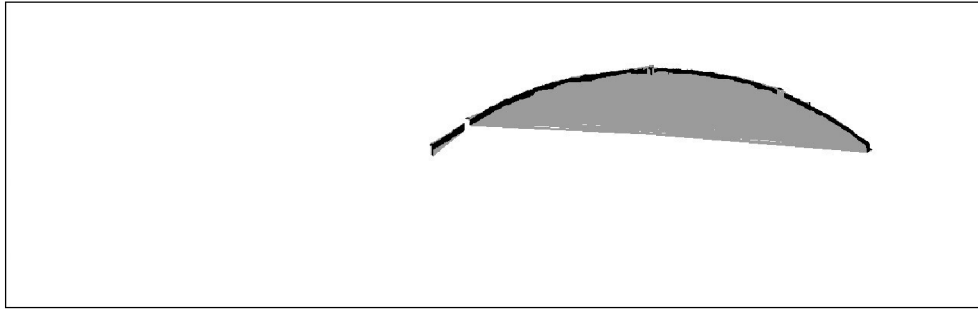
Po preskúmaní celého obrázku vybral prepojené útvary obsahujúce viac ako 10 bodov. Potom v každom útvere náhodne vyberal dvojice bodov. Pre útvar s  $n$  bodmi vybral  $\frac{n^2}{4}$  dvojíc bodov. Ak vyšiel počet dvojíc bodov nad 1000, vybral iba 1000. Tieto dvojice prepojil úsečkou. Jas čiary bol polovicou jasú bodov nad prahom, avšak jas v bodoch nad prahom pretnutých úsečkou nebol znížený. Výsledok algoritmu je na obrázku 4.7.

Tento algoritmus síce nie je najefektívnejším riešením tejto úlohy, je však jednoduchý na implementáciu a čas jeho behu je zanedbateľnou časťou celého času výpočtu.

Algoritmus používa náhodne vygenerované čísla a spôsobuje, že výpočet neprebieha vždy rovnako a niektorých vstupoch celý program niekedy uspie a niekedy nie.

#### 4.2.6 Čistenie vnútra kvapky

Na kvapke sa typicky nachádzajú odrazy svetiel, je cez ňu vidieť okolité objekty a môžu sa v nej nachádzať bubliny. Všetky tieto objekty zachytí Cannyho detektor.

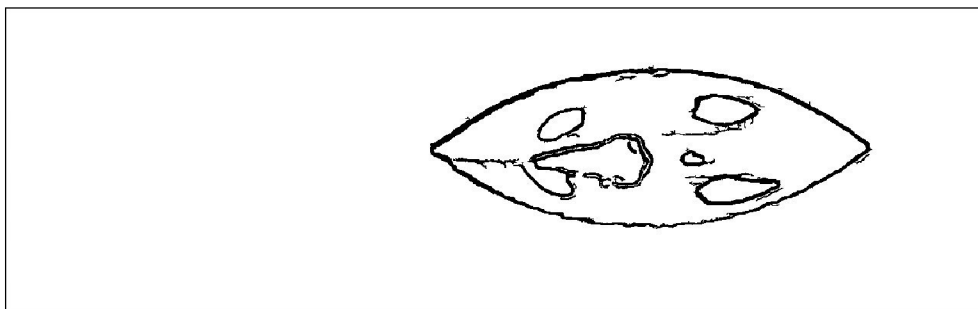


Obr. 4.7: Obrázok po vyplnení súvislých oblastí. Koniec ľavej časti oblúku nie je spojený so zvyškom, preto je zaradený do inej oblasti a nie je prepojený s ostatnými.

Často majú kruhový tvar, často úplnejší ako oblúk kvapky. Aby ich nepreferovala Houghova transformácia, je potrebné ich odstrániť.

Použitý algoritmus je podobný hľadaniu ciest, avšak začína na hornej časti obrázka a postupuje po stĺpcoch iba nadol. Ak sa pri tomto prieskume vyskytne bod nad prahom, prieskum v danom stĺpci sa zastaví. V tomto štádiu má algoritmus dva varianty, jeden zastaví v bode nad prahom a už nepokračuje (nazvaný *agresívny*), druhý umožňuje znova začať v zastavených stĺpcoch ak prieskum v okolitých stĺpcoch pokračuje (nazvaný *neagresívny*). Každý z nich bol užitočný v iných častiach.

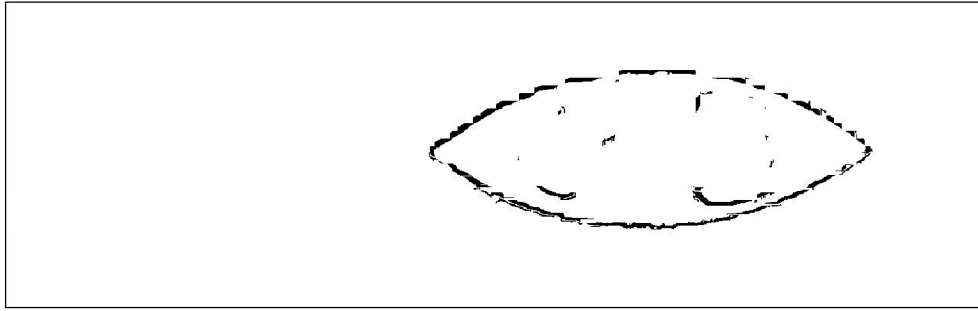
Výsledok čistenia neagresívnym algoritmom je na obrázku 4.9. Výsledok čistenia agresívnym algoritmom je na obrázku 4.10. Pôvodný obrázok pred čistením vnútra je na obrázku 4.8.



Obr. 4.8: Obrázok bez čistenia vnútra kvapky

### 4.3 Hľadanie kvapky

Tento algoritmus je založený na Houghovej transformácii, avšak dokáže nájsť aj mierne deformované oblúky a nemá problém s lokálnymi maximami (schéma 4.11).



Obr. 4.9: Obrázok po neagresívnom čistení vnútra kvapky

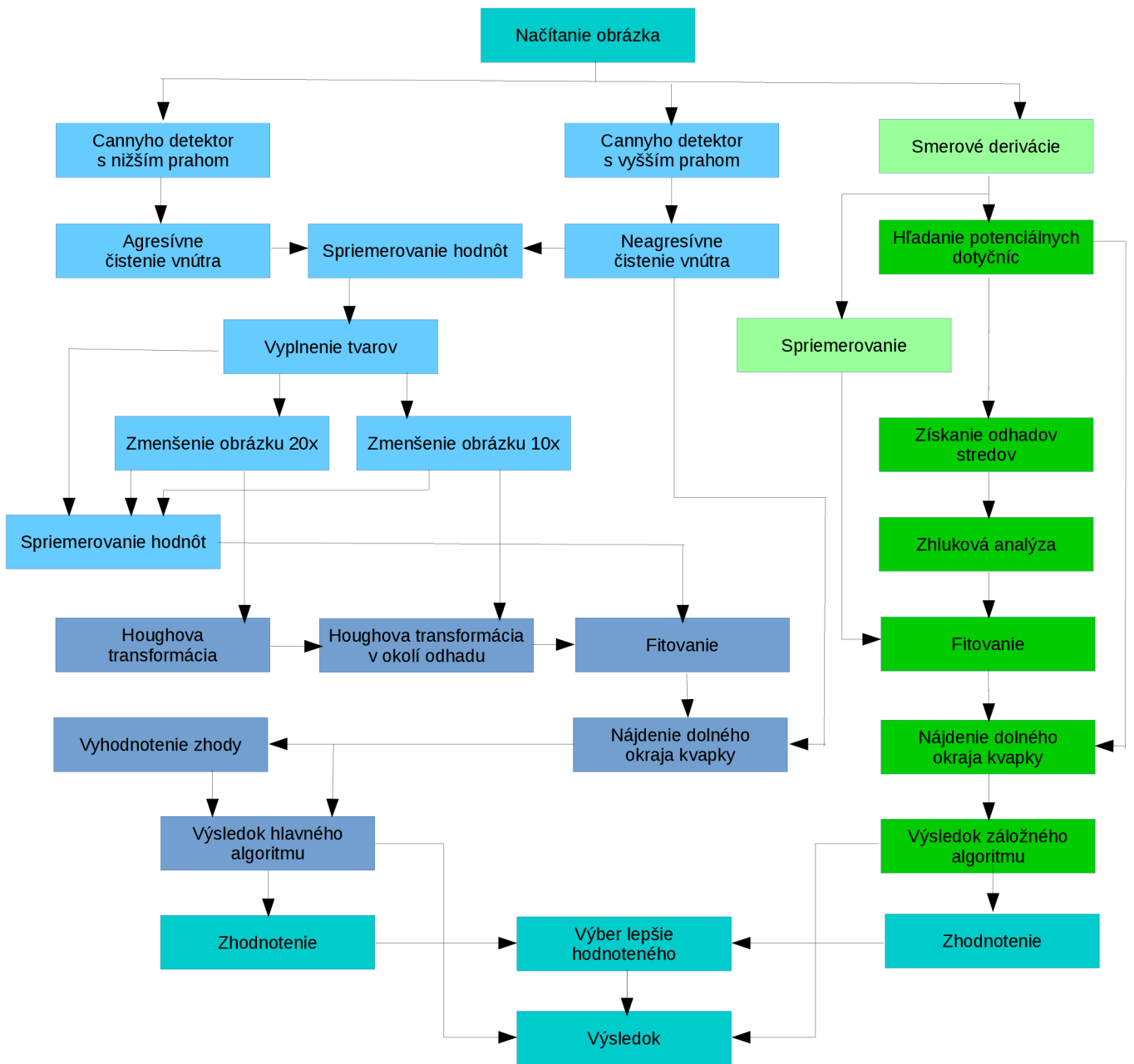


Obr. 4.10: Obrázok po agresívnom čistení vnútra kvapky

Používa sa fotografia po aplikácii Cannyho detektora, vyplnenia oblastí vnútri spojitých častí a agresívnej formy filtra na čistenie vnútra kvapky. Algoritmus najprv obrázok zmenší dvadsaťnásobne a na takto zmenšenom obrázku vykoná klasickú Houghovu transformáciu (obrázok 4.12). Body, ktorých hodnota je aspoň 90% maxima a sú lokálnymi maximami, sú vybrané pre ďalší výpočet.

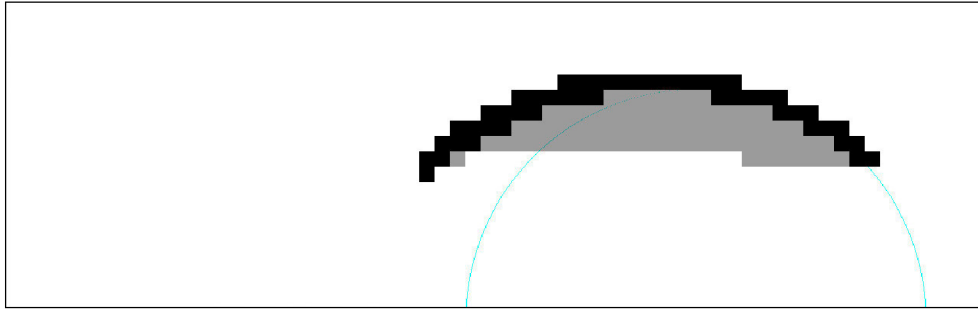
Potom sa nezmenšený obrázok zmenší desaťnásobne a na takto zmenšenom obrázku sa vykoná Houghova transformácia v okolí odhadov, ktoré boli získané v predchádzajúcom kroku. Z týchto hodnôt sú opäť vybrané body, ktorých hodnota je 90% maxima (patričného okolia odhadu, nie celkového) a sú lokálnymi maximami (obrázok 4.13). Táto časť je možné paralelizovať tak, že nevyžaduje komunikáciu medzi úlohami, preto sa každý odhad spracováva na samostatnom vlákne.

Potom je vytvorený ďalší obrázok spriemerovaním hodnôt na nezmenšenom obrázku a na obidvoch zmenšených obrázkoch z predchádzajúcich krokov. Na takto získané odhady sa použije algoritmus podobný metóde najmenších štvorcov, ktorý maximalizuje súčet hodnôt v pixeloch, ktorými útvar prechádza. Aby sa znížila pravdepodobnosť uviaznutia v lokálnom minime, skúšajú sa kombinácie posunov v každej súradnici o -8, -4, -2, -1, 0, 1, 2, 4 a 8 pixelov (obrázok 4.14). Tento algoritmus bude ďalej nazývaný *fitovanie*. Aj táto časť sa jednoduchej paralelizovateľnosti

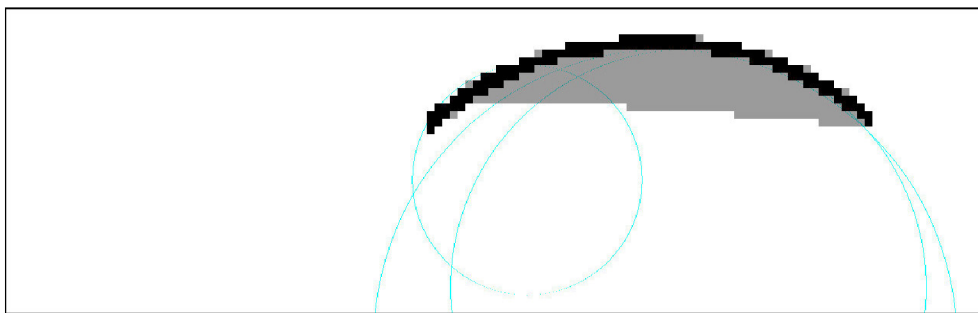


Obr. 4.11: Diagram krokov zvoleného riešenia





Obr. 4.12: Maximum klasickej Houghovej transformácie na dvadsaťkrát zmenšenom obrázku (červená kružnica)



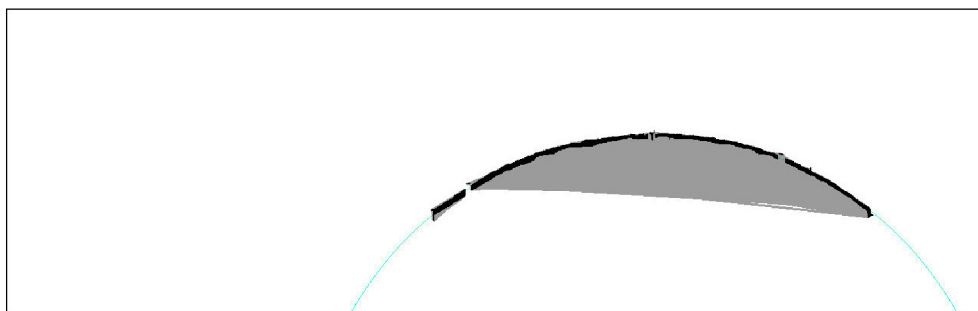
Obr. 4.13: Maximá klasickej Houghovej transformácie na desaťkrát zmenšenom obrázku v okolí výsledkov predchádzajúceho kroku (obrázok 4.12)

rozdeľuje na viac vlákien, jedno vlákno pre každý počiatkový odhad.

Odhady parametrov kružnice často výjdu rovnaké pri rôznych počiatkových odhadoch, preto sú podobné odhady zlúčené. Ak v nejakom kroku vyjde viac ako 10 odhadov, ďalej sa použije iba 10 najlepších. Algoritmus nedôjde k jednému výsledku, ale k viacerým možnostiam, z ktorých je vybraná správna podľa kvality výsledku.

Výpočet odmocniny je pomalý algoritmus, preto bol spočiatku používaný aproximačný algoritmus *fast inverse square root*, ktorý má nepresnosť približne 1%, avšak neobsahuje žiadne iterácie. Po vytvorení automatizovaných testov bola úspešnosť a rýchlosť postupu využívajúceho tento algoritmus porovnaná s postupom používajúcim iné algoritmy, konkrétne odmocninu založenú na bitovom posune a implementáciu odmocniny v štandardných knižniciach C (tabuľka 4.1).

Štandardná implementácia odmocniny sa ukázala ako najúspešnejšia. Bitový posun bol síce rýchlejší, ale úspešnosť postupu bola príliš nízka.



Obr. 4.14: Maximá po fitovaní v okolí výsledkov predchádzajúceho kroku (obrázok 4.13)

typ algoritmu	úspešnosť	čas výpočtu
bitový posun	27%	85%
fast inverse square root	67%	102%
štandardná implementácia	76%	100%

Tab. 4.1: Rýchlosť výpočtu rôznych implementácií odmocniny (100% je rýchlosť štandardnej implementácie) a úspešnosť celého algoritmu pri použití tejto implementácie

### 4.3.1 Záložné riešenie

Tento algoritmus mal výrazne odlišné fungovanie voči bežným metódam hľadania kružnice. Mal veľkú pravdepodobnosť úspechu, ale nie najvyššiu. Avšak bol vďaka svojej odlišnosti schopný nájsť správne riešenie približne v 20% prípadov, kde hlavný algoritmus používajúci Houghovu transformáciu zlyhal. Rozhodovanie, kedy použiť výsledok tohoto algoritmu je popísané v kapitole 5.

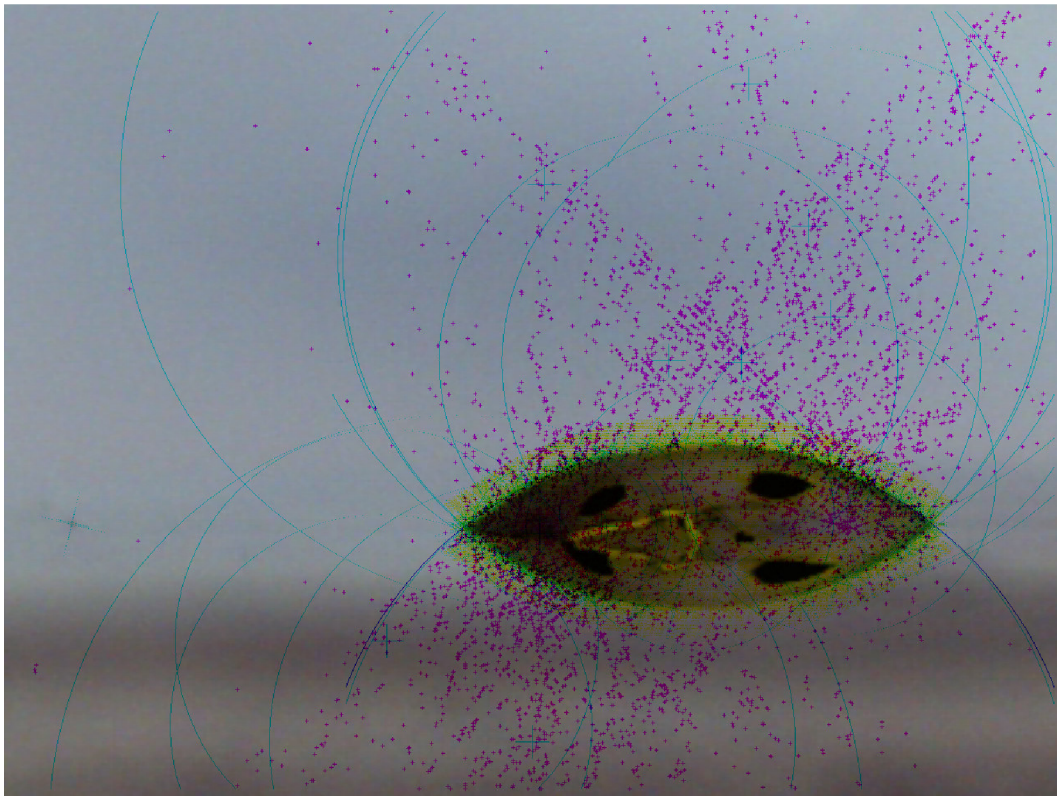
Na začiatku boli vypočítané smerové derivácie obrázka v mnohých smeroch (po skúšaní rôznych počtov sa ako najlepší počet smerov z pohľadu prenosi a rýchlosti ukázal počet 16). V týchto deriváciách boli nájdené body, ktoré mali vyššiu hodnotu ako prahová hodnota a bol určený smer, v ktorom bola táto derivácia najvyššia. Tým boli získané dotyčnice kružnice a body, kde sa dotýkali, avšak presnosť nebola dostatočná.

V druhom kroku bol náhodne vybraný určitý počet dvojíc dotyčníc (používal sa počet 1000, vyššie počty už nevedli k zlepšeniu presnosti) a pre každú bol nájdený bod, kde sa pretínali kolmice na ne vedené z bodu dotyku. S týmto bodom boli uložené aj jeho vzdialenosti od bodov dotyku. Z týchto bodov boli vylúčené tie, pri ktorých bol príliš veľký rozdiel ich vzdialeností od bodov dotyku. Na zostávajúce body bola použitá zhluková analýza. Stredy zhlukov boli prvotnými odhadmi polohy

stred a priemerné vzdialenosti od bodov dotyku boli použité ako odhad polomeru 4.15.

Na tieto odhady bolo použité fitovanie ako pri hlavnom algoritme. Aby sa zabránilo uviaznutiu v lokálnych minimách, boli vyskúšané aj body vo väčších vzdialenostiach od bodu odhadu podobne ako v hlavnom algoritme.

Podobne ako vyplňanie oblastí popísané v kapitole 4.2.5, aj tento algoritmus používal náhodnosť a preto na niektorých vstupoch niekedy uspel a niekedy zlyhal.



Obr. 4.15: Použitie metódy dotyčníc - nájdené kolmice na kružnicu sú vyznačené fialovou farbou, prieniky dvojíc týchto kolmíc ako zelené body, potenciálne kružnice červenou a vybraná kružnica žltou (pre lepšiu viditeľnosť sú farby invertované). Prvotné odhady kružníc sú zelené, pravdepodobná kružnica je fialová.

## 4.4 Nájdenie dolného okraja kvapky

Keď už je známe, kde sa nachádza kružnica, treba ešte zistiť, akú časť oblúka vidno na fotografii.

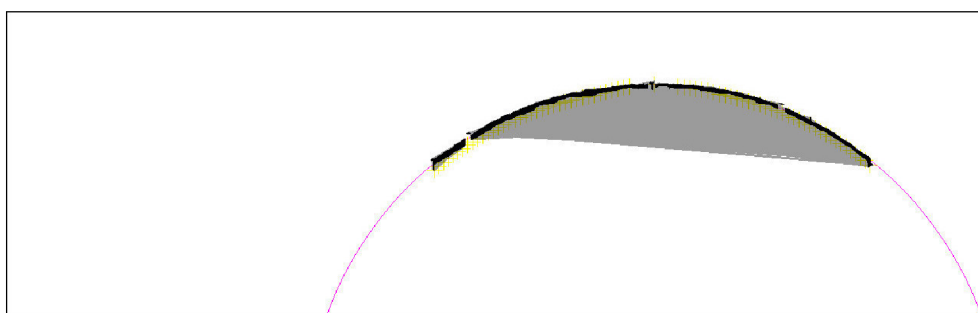
V ideálnom prípade by stačilo nájsť okraj vzorky pomocou Houghovej transformácie na priamky a nájsť body, kde sa pretne s kružnicou kvapky. V praxi však

zväčša nebolo možné nájsť túto hranu ani pre človeka, pretože predný okraj vzorky bol bližšie ku kamere ako kvapka, v dôsledku čoho bol výrazne rozmazaný. Ak sa na kvapku pozeral mierne zvrchu, žiaden okraj sa na fotografii ani nenachádzal.

Preto bolo potrebné sledovať oblúk kružnice a hľadať miesto, kde sa končí. Úlohu bolo možné riešiť tak, že sa sledovala kružnica od vrchu a našiel sa bod, kde sa už hrana nevyskytovala. Tento prístup fungoval, avšak v zlyhával v prípadoch keď:

- kvapka nemala tvar dokonalého oblúka, zvlášť pri dolnom okraji.
- oblúk kvapky nebol úplný
- v blízkosti spodného okraja kvapky sa nachádzala hrana substrátu
- kvapka sa odrážala od substrátu a kontaktný uhol bol blízky  $90^\circ$

Vytvorený algoritmus teda sleduje oblúk iterovaním cez všetky uhly v polárnych súradniciach (od najvyššieho bodu na kvapke). Pri každom kroku skúša okrem polomeru nájdenej kvapky aj niekoľko hodnôt podobných tomuto polomeru, ktoré, ak sa zhodujú s oblúkom v grafe lepšie ako polomer, tak sú použité ako nový polomer (obrázok 4.16). Takýto algoritmus dokáže nájsť oblúk, tolerujúc mierne sa meniaci polomer. Tolerancia však nesmie byť príliš vysoká, pretože inak by mohol nájsť okraj substrátu a sledovať ten.



Obr. 4.16: Nájdenie dolného okraja oblúku kvapky

Časti, kde sa oblúk kvapky stratí, sa od konca neodlišujú. Avšak po nájdení potenciálneho konca sa pokračuje v prehľadávaní oblúka, či sa nájde ďalšia súvislá časť. Očakáva sa prítomnosť hrán na oboch stranách. Ak je očakávaná prítomnosť hrán s takým polomerom, s akým sa naposledy pracovalo, tak sa často stávalo, že algoritmus sledoval nesprávny oblúk, napríklad odraz od kvapky, ktorý nebol odstránený práve kvôli trhline v oblúku, a potom nenašiel pokračovanie správneho oblúka. Ak očakávaná prítomnosť hrán s rovnakým polomerom ako vyšiel z Houghovej transformácie, tak si algoritmus nedokáže poradiť s kvapkami s kontaktným uhlom blízko  $90^\circ$  a zrkadlovým obrazom od vzorky, čo je však zriedkavý prípad.

V tomto čase už bolo možné priamo získať výsledok, tri body definujúce oblúk s okrajmi. Body, kde sa oblúk končil, boli použité ako body pretnutia oblúku kvapky

s rovinou vzorky. Ako tretí bod, ktorý bol potrebný na jednoznačný popis kružnice, bol použitý najvyšší bod na kvapke.

Ako najužitočnejší vstup sa ukázala fotografia vzniknutá spriemerovaním hodnôt fotografie po aplikácii Cannyho detektora a fotografie po aplikácii čistenia vnútra.

Celková úspešnosť na sade 161 fotografií z rôznych experimentov bola približne 76%, ak bolo za správne riešenie považované riešenie s odchýlkou pod  $8^\circ$  pri kontaktných uhloch nad  $20^\circ$  a pod  $2^\circ$  pri kontaktných uhloch pod  $20^\circ$ .

#### 4.4.1 Záložný postup

Tento postup bol používaný spolu so záložným algoritmom spočívajúcim v hľadaní dotyčníc. Na nájdenie koncov oblúka boli použité dotyčnice, v časti s oblúkom bola vzdialenosť dotyčníc od stredu približne rovná polomeru, v časti bez oblúka bola zväčša náhodná (nesprávne zaradené dotyčnice). Nájdením tejto zmeny bolo možné relatívne presne určiť, kde sa nachádza dolný okraj kvapky.

### 4.5 Neperspektívne prístupy

Okrem nakoniec použitého postupu boli vyskúšané rôzne prístupy, ktoré čiastočne fungovali, avšak nedosahovali uspokojujúce výsledky.

#### 4.5.1 Bežná Houghova transformácia

Bežná Houghova transformácia vyskúša pre všetky prípustné polohy stredu a všetky prípustné polomery súčet hodnôt pixelov v kružnici. Tento spôsob výpočtu bol vysoko spoľahlivý, avšak kvôli zložitosti algoritmu ho nebolo možné reálne použiť.

Zložitosť algoritmu je  $\mathcal{O}(h \cdot w \cdot (r_{max} - r_{min}))$ , kde  $h$  a  $w$  sú počty prípustných polôh stredu a  $r_{min}$  a  $r_{max}$  sú minimálny a maximálny polomer. Vzdialenosť stredu od ľavého okraja fotografie sa nepohybovala v príliš veľkom intervale, avšak výška stredu bola pri nízkych kontaktných uhloch často aj 3000 pixelov pod spodným okrajom a polomer mohol mať hodnotu väčšiu ako 3000. Pre zrýchlenie bol na výpočet súradníc bodov na kružnici použila namiesto sínusu alebo cachovania rovnica kruhu:

$$f(r) = \sqrt{1 - r^2}. \quad (4.3)$$

Vykonanie Houghovej transformácie však stále trvalo niekoľko minút, čo bolo príliš dlho. Pri zmenšení Houghovho priestoru tak, aby čas vykonávania algoritmu bol dostatočne krátky (30x70 prípustných polôh stredu, 70 prípustných polomerov), presnosť sa zhoršila natolko, že metóda uspela iba výnimočne.

Pretože Houghov priestor je pri lokalizácii kružnice trojrozmerný, tento postup nie je ilustrovaný obrázkami.

## 4.5.2 Gradientná Houghova transformácia

Výrazne zmenšený Houghov priestor umožňuje nájsť oblasti, ktoré sú potenciálne blízko parametrov kružníc a použiť ich ako prvotné odhady. Okolo každého odhadu je možné vypočítať hodnoty v nezmenšenom Houghovom priestore a posunúť sa do bodu, ktorý je bližšie k parametrom kružnice a použiť ho ako nový odhad a toto opakovať dovtedy, kým sa nájde maximum.

Týmto postupom sa síce vo väčšine prípadov prišlo na správne riešenie, avšak bola príliš veľká pravdepodobnosť, že v zmenšenom Houghovom priestore sa nenachádzalo nijaké výrazné zvýšenie hodnoty v blízkosti správneho výsledku a potom sa pri postupe po gradiente nenašiel správny oblúk. Postupné zlepšovanie odhadu pre maximalizáciu súčtu hodnôt pixelov bolo nakoniec použité, avšak prvotné odhady boli získané inak.

## 4.5.3 Hľadanie roviny podložky pomocou Houghovej transformácie

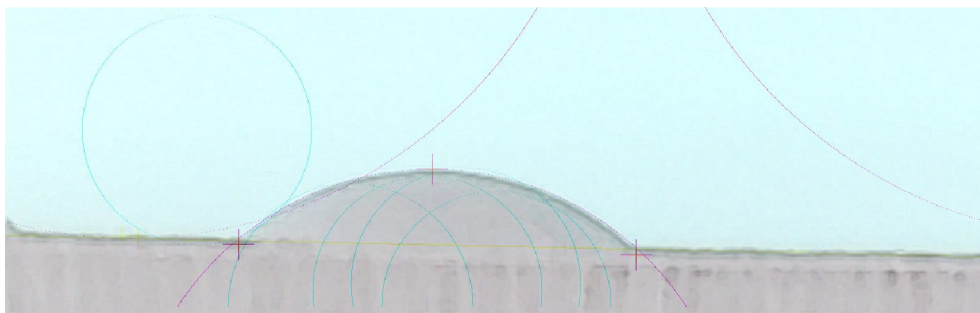
Pri tomto algoritme išlo o nájdenie miesta, kde sa oblúk kvapky končí. Vychádzal z predpokladu, že sa oblúk končí na podložke, na ktorú sa pozeráme v smere jej roviny. Okraj podložky je na fotografii viditeľný ako výrazný prechod medzi farbou podložky a farbou pozadia. Preto sa ako vhodné riešenie javilo použitie Houghovej transformácie na čiary.

Na výpočet bola použitá fotografia po aplikácii Gaussovho a následne Sobelovho operátora. Na ňu bola použitá Houghova transformácia na lokalizáciu priamok a v nej bolo nájdené maximum (obrázok 4.17). Pretože podložka mala byť približne vodorovná, stačil malý rozsah možných sklonov (rozdiel výšok začiatku a konca do 30 pixelov).

Algoritmus v približne 30% prípadov korektne našiel miesto, kde sa končí oblúk kvapky (napríklad obrázok 4.18). Avšak v približne 60% prípadov bol okraj podložky nedostatočne ostrý a maximum Houghovej transformácie bola úplne iná čiara (napríklad obrázok 4.19). V približne 10% prípadov kamera nesnímala z roviny podložky a koniec oblúku kvapky neležal na jej okraji (napríklad obrázok 4.20). Aj v prípade vyriešenia problému s neostrými okrajmi podložky by bola úspešnosť tejto metódy príliš nízka.



Obr. 4.17: Houghova transformácia na lokalizáciu priamok. Maximum je jasne viditeľné.



Obr. 4.18: Prípád, keď Houghova transformácia korektne našla dolný okraj oblúka kružnice (pre lepšiu viditeľnosť sú farby invertované)

#### 4.5.4 Hľadanie okrajov v obraze transformovanom do polárnych súradníc

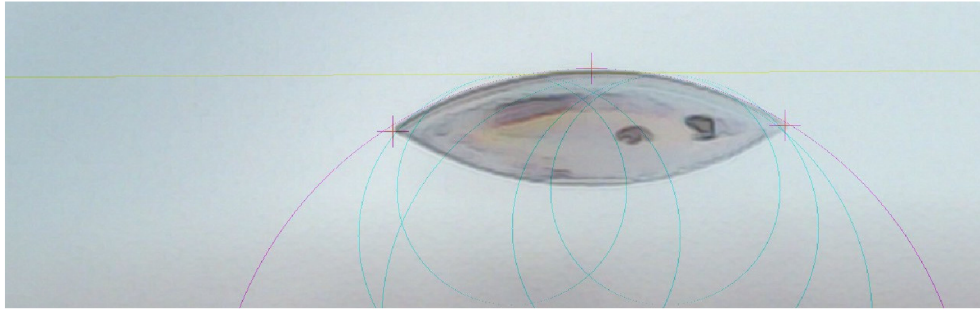
Tento algoritmus bol založený na myšlienke, že neúplná kružnica je v polárnych súradniciach úsečkou a konce úsečky sa hľadajú podstatne ľahšie ako konce kružnice (obrázok 4.21). Stred kružnice bol získaný z viacstupňovej Houghovej transformácie.

Ako vstup do algoritmu bol použitý výstup zo Sobelovho operátora alebo kombinácia výstupu zo Sobelovho operátora a Cannyho detektora. Potenciálne priamky, na ktorých táto úsečka ležala, bolo možné nájsť pomocou obyčajnej Houghovej transformácie na priamky alebo odvodiť z parametrov predpokladanej kružnice.

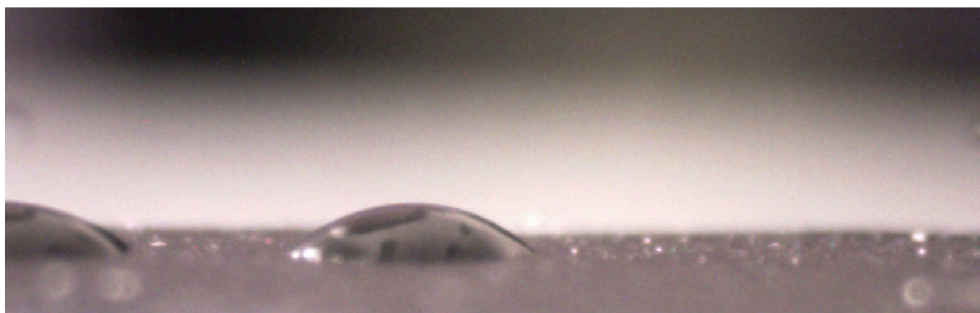
Pozdĺž týchto priamok boli skúšané kombinácie začiatočného bodu a koncového bodu a bola počítaná ich hodnotiaca funkcia. Funkcia bola súčtom veličiny  $\alpha$  každom v bode úseku:

$$\alpha(r, \theta) = \int_{-d}^d v(r, x + \theta) \cdot (1 - |x - d| \cdot 2) dx, \quad (4.4)$$

kde  $r$  a  $\theta$  boli súradnice bodu,  $v$  bola hodnota transformovaného obrázka v tomto bode a  $d$  bol vhodne určený parameter. Táto veličina bola súčtom hodnôt v bodoch



Obr. 4.19: Prípád, keď bola podložka príliš neostrá na nájdenie roviny podložky (pre lepšiu viditeľnosť sú farby invertované)



Obr. 4.20: Prípád, keď sa oblúk kružnice nekončí na hrane podložky (pre lepšiu viditeľnosť sú farby invertované)

na krátkej kolmici na priamku vedenej cez tento bod násobeným koeficientom, ktorý klesal so vzdialenosťou lineárne tak, že od určitej vzdialenosti bol záporný.

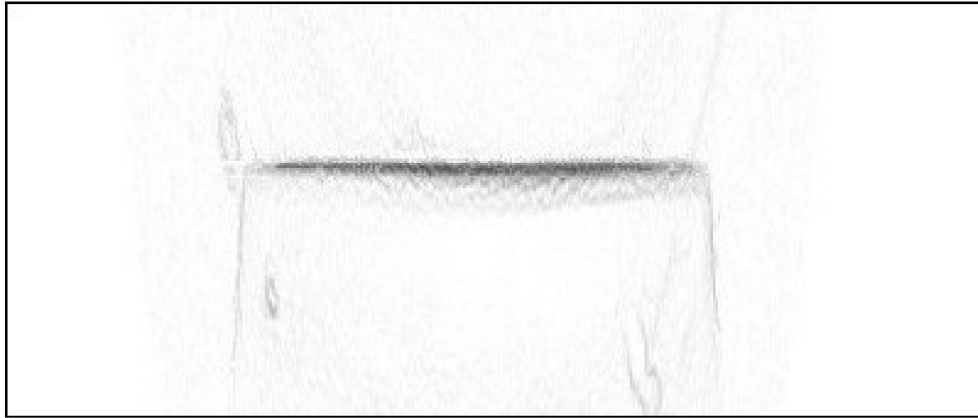
Táto veličina hodnotila body na priamke podľa toho, ako sú na nej v tomto bode sústredené body s vysokou hodnotou. Body, pri ktorých bola vysoká hodnota na priamke a nízka ďalej od priamky mali pozitívne hodnotenie. Body, pri ktorých bola nízka hodnota na priamke a vysoká niekde v blízkosti priamky mali negatívne hodnotenie.

Ukážka hodnota priamky v závislosti od začiatku a konca priamky je na obrázku 4.22.

Tento spôsob umožňoval nájsť oblasti pozdĺž úsečiek, ktoré mali veľké hodnoty v blízkosti úsečiek a nízke hodnoty ďalej od úsečiek, pričom bolo možné preklenúť nevyhovujúce oblasti, ak za nimi úsečka pokračovala. Výsledky, kde bola úsečka výrazne asymetrická (teda kvapka by vľavo končila v úplne inej výške ako vpravo), boli znevýhodnené. Úspešnosť tohoto postupu s použitím viacškálovej Houghovej transformácie bola približne 66%.

V praxi sa totiž hrúbky úsečky rôznili a stávalo sa, že úsečka vôbec nebola naj-



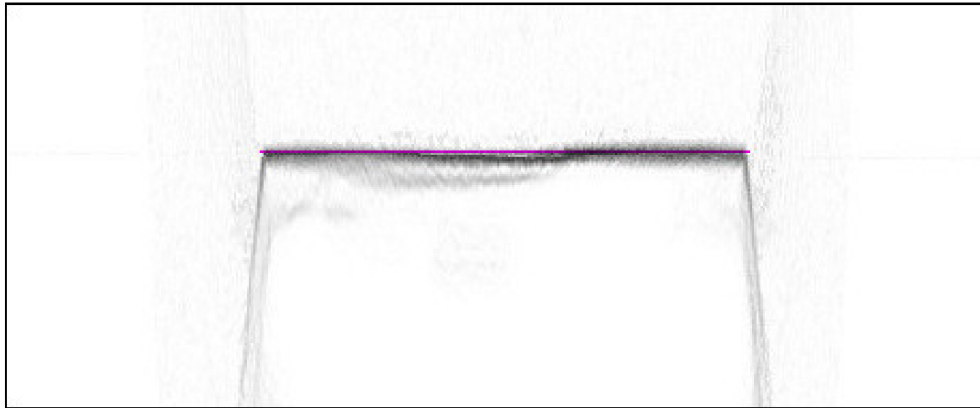


Obr. 4.21: Polárna transformácia výsledku detekcie hrán

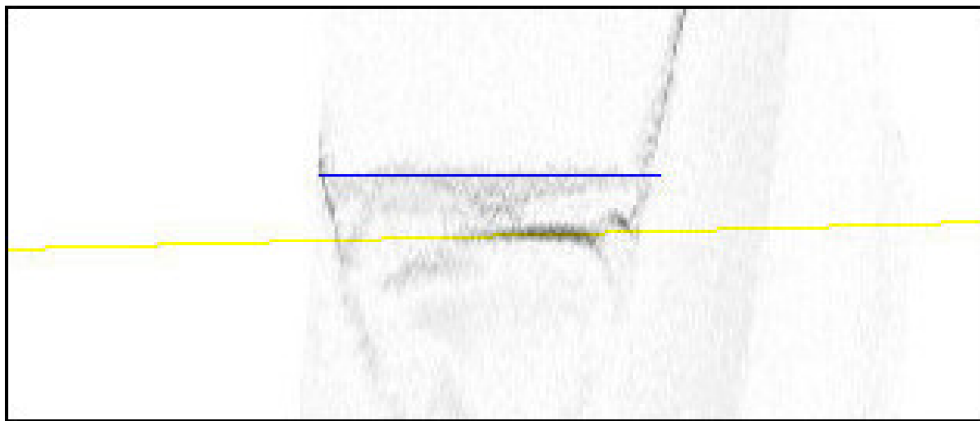


Obr. 4.22: Kvalita úsečky v závislosti od polohy začiatku a konca

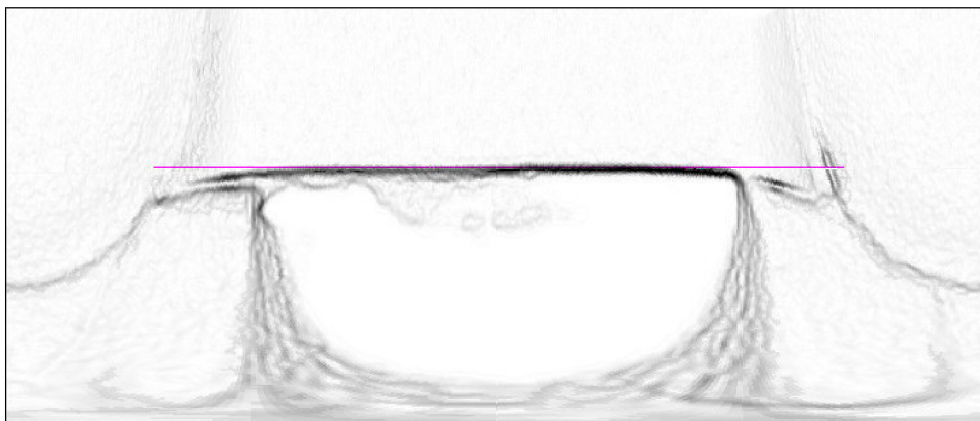
dená. Okrem toho sa stávalo, že úsečka bola krivá, pretože kvapka bola mala mierne eliptický tvar (obrázok 4.24). Preto boli pre sklon lineárnej závislosti parametra od vzdialenosti od priamky skúšané rôzne hodnoty a ako výsledok bol použitý ten s najlepším výsledkom. Problémom tejto metódy však bol nezanedbateľný vplyv na čas výpočtu napriek cachovaniu medzivýsledkov a hlavne tendencia považovať za úsečku aj rôzne iné štruktúry, napríklad odraz kvapky od povrchu substrátu (obrázok 4.25). Úspešnosť tohoto postupu s použitím viacškálovej Houghovej transformácie bola 66%, rovnako ako bez tejto úpravy.



Obr. 4.23: Nájdenie koncov kružnice pomocou polárnej transformácie



Obr. 4.24: Problémy spôsobené mierne eliptickým tvarom kvapky



Obr. 4.25: Problémy spôsobené odrazom kvapky od substrátu

### 4.5.5 Hľadanie elíps

Tento postup mal riešiť problém, keď tvar kvapky nebol dokonale kruhový. Prispôbenie algoritmu hľadania koncov oblúka na tento problém však spôsobovalo, že okraj podložky ovplyvnil hľadanie konca oblúka a zhoršil presnosť.

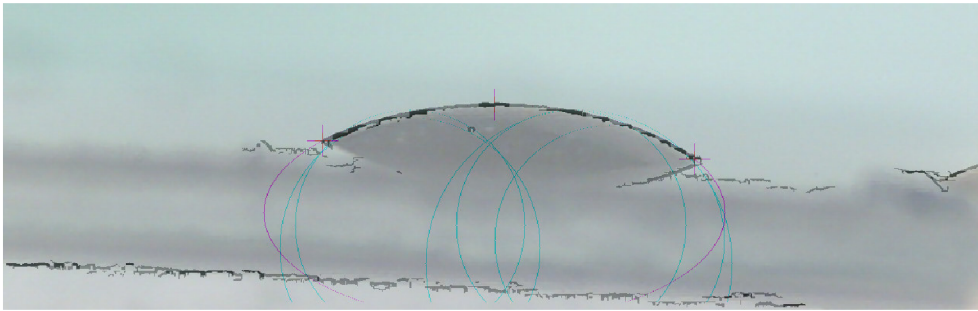
V tomto prístupe bola po odhade polohy kružnice získanom z Houghovej transformácie zmenšeného obrázka pomocou fitovania hľadaná elipsa namiesto kružnice.. Ako počiatočný odhad oboch poloosí bol použitý polomer odhadu kružnice. Predpokladala sa elipsa s uhlom náklonu  $0^\circ$ , pretože sily deformujúce tvar kružnice pôsobili symetricky. Boli teda hľadané štyri parametre namiesto troch. Pre výpočet bodov elipsy boli použité vzorce:

$$y = b\sqrt{1 - \frac{x^2}{a^2}}, \quad (4.5)$$

$$x = \frac{ab \cos \theta}{\sqrt{b^2 \cos^2 \theta + a^2 \sin^2 \theta}} \quad \text{a} \quad y = \frac{ab \sin \theta}{\sqrt{b^2 \cos^2 \theta + a^2 \sin^2 \theta}}, \quad (4.6)$$

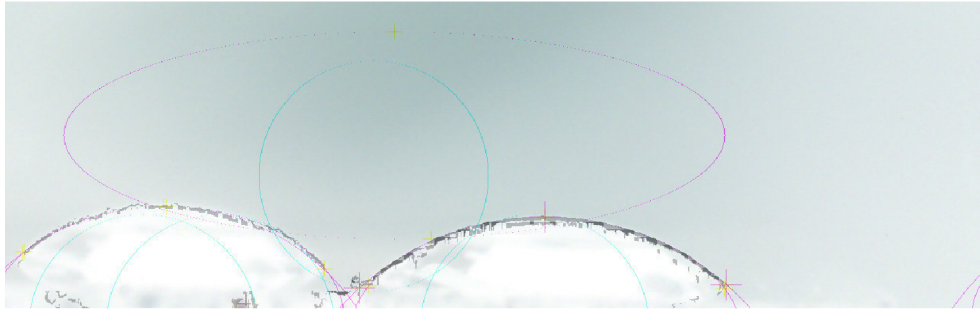
kde  $x$  a  $y$  boli súradnice bodu a  $a$  a  $b$  boli dĺžky poloosí ( $a$  v smere  $x$ ,  $b$  v smere  $y$ ). Vzorce 4.6 slúžili na určenie súradníc bodov z parametra  $\theta$ .

Tento postup bol pomerne úspešný. Bol veľmi úspešný pri hľadaní malých kontaktných uhlov (obrázok 4.26), pravdepodobne kvôli tomu, že elipsa s vysokou excentricitou nemala stred príliš ďaleko od súradníc objektu na fotografii a odchýlka vrchnej časti jej oblúka od oblúka kružnice nebola príliš veľká.

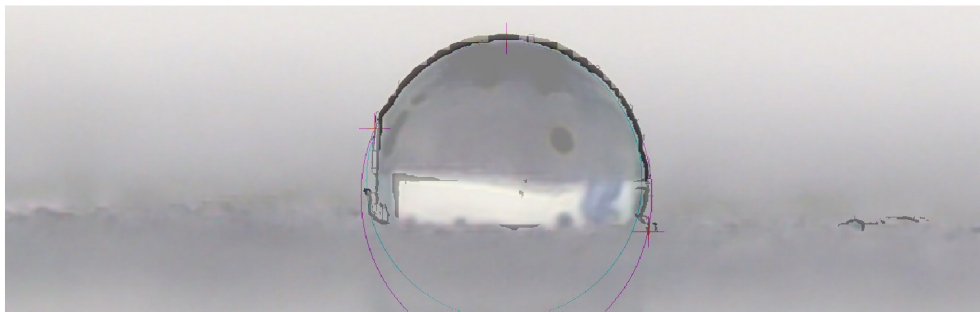


Obr. 4.26: Úspech pri nájdení malého kontaktného uhlu pomocou hľadania elíps (pre lepšiu viditeľnosť sú farby invertované)

Napriek tomu bol tento postup menej úspešný ako hľadanie kružníc. Kvôli vyššiemu počtu stupňov voľnosti mal výpočet väčšiu pravdepodobnosť na uviaznutie v lokálnom minime (obrázok 4.27). Okrem toho sa často stávalo, že nájdená elipsa čiastočne obsahovala rôzne odrazy a povrch substrátu namiesto celého oblúku kvapky (obrázok 4.28). Elipsa tiež v mnohých prípadoch obsahovala aj spodnú hranu kvapky



Obr. 4.27: Prípád, kde vyšší počet voľných parametrov elipsy spôsobil, že fitovanie skončilo v lokálnom minime a hľadanie kružnice uspelo (pre lepšiu viditeľnosť sú farby invertované)



Obr. 4.28: Zablúdenie hľadania elíps fitovaním v nesúvisiacich štruktúrach (pre lepšiu viditeľnosť sú farby invertované)

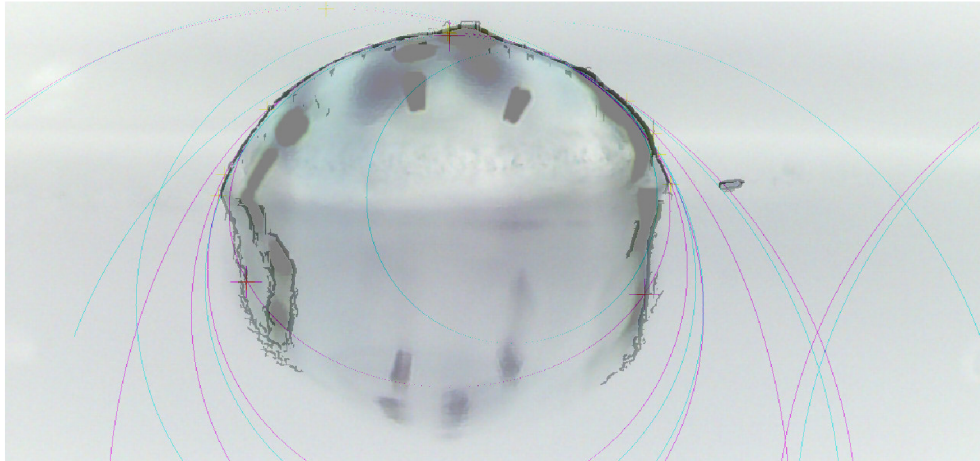
alebo jej odraz od substrátu, čo znižovalo presnosť nájdenia hornej časti oblúka (obrázok 4.29).

#### 4.5.6 Hľadanie kružníc aj elíps

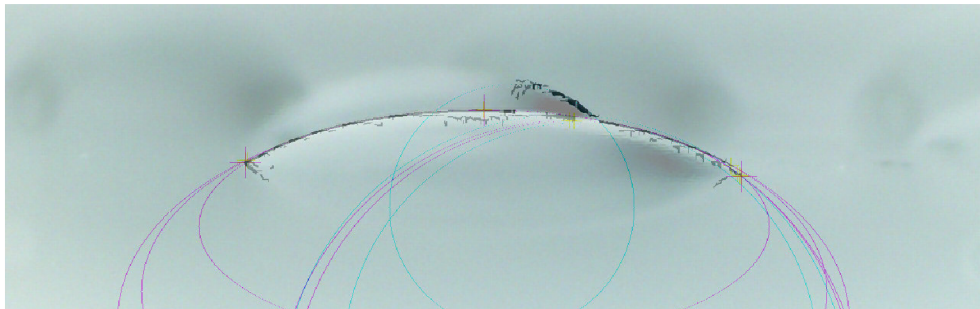
Kvôli pomerne vysokej úspešnosti pri použití fitovania elíps sa zdalo, že kombinácia hľadania kružníc a hľadania elíps by mohla mať väčšiu pravdepodobnosť na úspech ako tieto metódy samostatne.

Preto bol vyskúšaný postup, kde bola každý počiatočný odhad fitovaná aj elipsa aj kružnica. Následne boli nájdené okraje, vypočítané hodnotenie výsledku a bolo vybrané to riešenie, kde bolo hodnotiace kritérium najvyššie. Vo väčšine prípadov bol výsledok oboch výpočtov rovnaký (4.30).

Vyskytol sa však problém s hodnotiacim kritériom. Nepodarilo sa nájsť také, ktoré by našlo najlepšie z riešení s dostatočnou spoľahlivosťou na to, aby počet vybraných nesprávne nájdených elíps (obrázok 4.31) prevyšoval počet vybraných



Obr. 4.29: Nechcené hľadanie elíps aj s odrazom kvapky od substrátu (pre lepšiu viditeľnosť sú farby invertované)



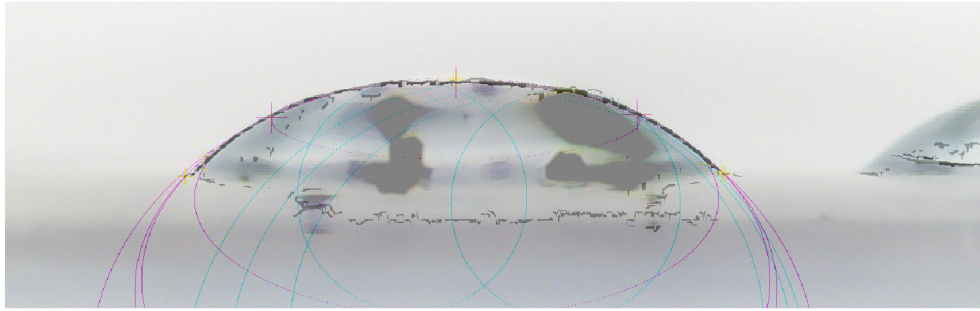
Obr. 4.30: Veľmi častý prípad, kde hľadanie elipsy viedlo na rovnaký výsledok ako hľadanie kružnice (pre lepšiu viditeľnosť sú farby invertované)

správne nájdených elíps k nesprávne nájdeným kružniciam (obrázok 4.32). Kvôli tomu bola úspešnosť algoritmu približne o 3% nižšia ako bez hľadania elíps.

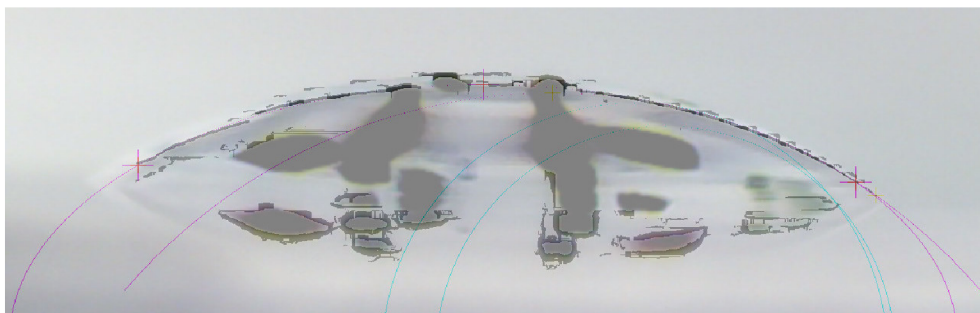
Pretože pri hľadaní elipsy bol tvar elipsy často skreslený podložkou, elipsa bola často širšia ako oblúk kvapky a nájdený kontaktný uhol bol nižší. Tento problém znemožňoval použitie elipsy iba v prípadoch, kde bol uhol malý.

## 4.6 Automatické testovanie a doladovanie parametrov

Testovať úspešnosť výpočtu manuálnou kontrolou výsledkov bolo nepraktické, preto bol napísaný algoritmus, ktorý prečítal súbor, v ktorom boli názvy fotografií a ich manuálne určené kontaktné uhly pre overenie správnosti, vypočítal kontaktné uhly



Obr. 4.31: Nežiadané preferovanie lepšie vyhovujúcej ale neúplnej elipsy, keď bola kružnica nájdená správne (pre lepšiu viditeľnosť sú farby invertované)



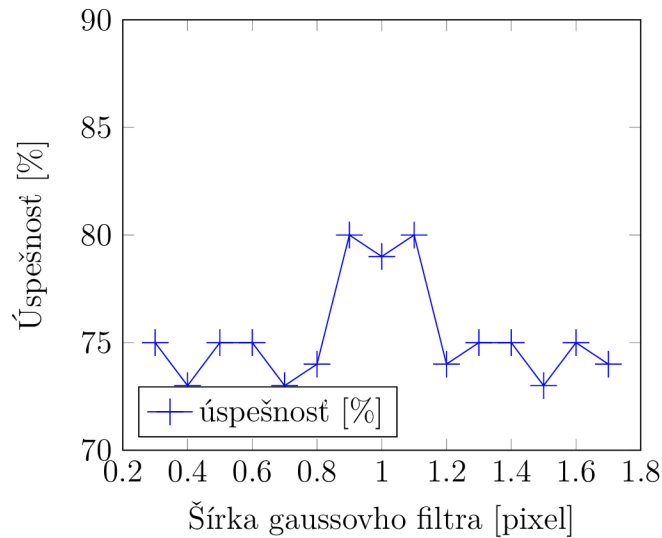
Obr. 4.32: Prípád, kde hľadanie elipsy viedlo na lepší výsledok ako hľadanie kružnice (pre lepšiu viditeľnosť sú farby invertované)

z týchto fotografií a vypočítal úspešnosť postupu. Tieto fotografie boli vybrané tak, aby sa medzi sebou líšili a pochádzali z veľkého množstva experimentov.

Tým sa umožnilo aj riešiť problémy podobné nasledujúcemu: Pri nízkom dolnom prahu Cannyho detektora sú pripojené aj menej výrazné hrany, čo zlepšuje úspešnosť nájdenia málo výrazných okrajov kvapiek, avšak zvýši množstvo nesprávne vyznačených útvarov nesúvisiacich s experimentom. Vysoká hodnota naopak zníži množstvo vyznačených parazitných útvarov, avšak zhorší úspešnosť nájdenia málo výrazných okrajov. Bolo treba nájsť kompromis, kde je úspešnosť najlepšia. Vďaka možnosti merať úspešnosť automaticky bolo možné vytvoriť algoritmus, ktorý doladzoval parametre výpočtu (napríklad prahy Cannyho detektora) tak, aby bola maximalizovaná úspešnosť. Toto umožnilo zvýšiť úspešnosť výpočtu o niekoľko percent.

Pretože niektoré časti algoritmu používali náhodnosť (vypĺňanie oblastí popísané v kapitole 4.2.5 a záložné riešenie popísané v kapitole 4.3.1), úspešnosť pre tie isté parametre na vzorke 161 fotografií mala rozptyl približne 2%. To bránilo preučeniu parametrov na šum v jednej skupine výsledkov, avšak často viedlo k dojmu, že nejaký parameter zlepšuje úspešnosť, aj keď bola za tým náhoda. Preto bolo potrebné

Ukážka nápadnej závislosti úspešnosti na parametri



Obr. 4.33: Ukážka závislosti úspešnosti na parametri, kde existuje nápadná optimálna hodnota

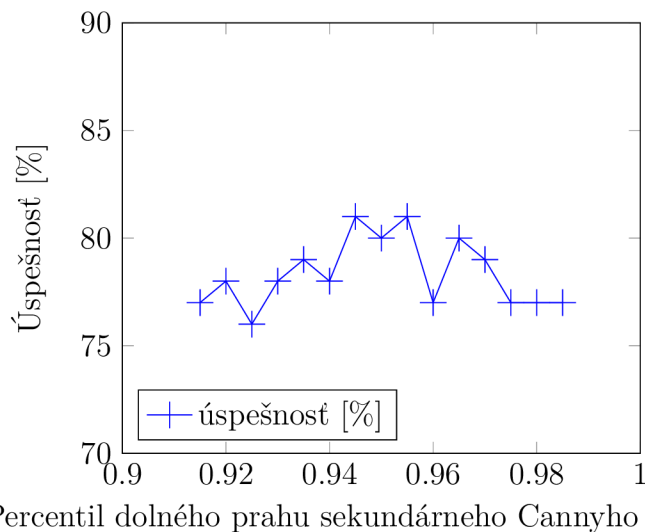
zostrojíte dlhšiu krivku závislosti na danom parametri.

Úspešnosť celého algoritmu výrazne závisela na nastavení niektorých parametrov. Príkladom takéhoto parametra bola šírka Gaussovho filtra (obrázok 4.33). Na mnohých parametroch bola úspešnosť menej závislá, stačilo, aby boli v nejakom širšom intervale. Takýmto parametrom bol napríklad dolný prah Cannyho detektora (obrázok 4.34). Na konkrétnych hodnotách mnohých parametrov úspešnosť závisela veľmi málo a v meranej závislosti bol iba šum (obrázok 4.35). Väčšina parametrov mala na úspešnosť malý vplyv, počiatočnú hodnotu určenú ručne nebolo treba ďalej doladovať. To nasvedčovalo tomu, že nedochádzalo k špecializácii parametrov na skúšanú vzorku. Jednotlivé parametre sú vypísané v tabuľke 4.2.

## 4.7 Zhrnutie

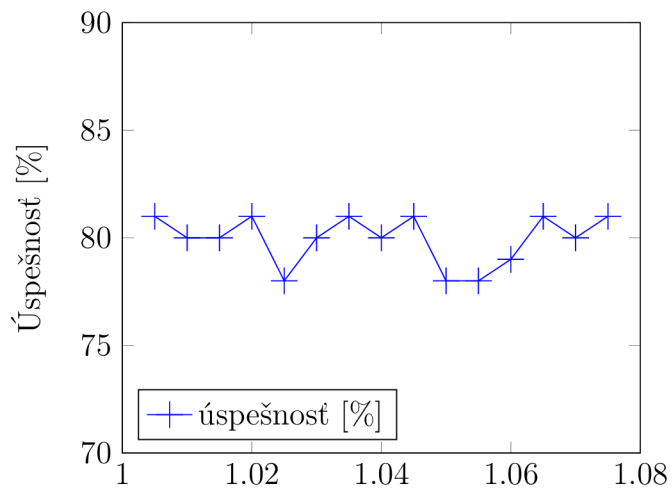
Bežne používané metódy s ľahko dostupnou implementáciou dokázali nájsť na obrázkoch z testovacej sady kvapku iba výnimočne. Kvôli tomuto boli vyskúšané rôzne algoritmy a z nich boli vybrané dva. Jeden z nich mal najvyššiu úspešnosť. Druhý dokázal vyriešiť nezanedbateľnú časť úloh, ktoré prvý algoritmus vyriešiť nedokázal. Parametre algoritmov boli automaticky doladené tak, aby sa maximalizovala úspešnosť na testovacej sade.

Ukážka od šumu odlišiteľnej závislosti úspešnosti na parametri



Obr. 4.34: Ukážka závislosti úspešnosti na parametri, kde je optimálna hodnota viditeľne v nejakom intervale

Ukážka od šumu neodlišiteľnej závislosti úspešnosti na parametri



Obr. 4.35: Ukážka závislosti úspešnosti na parametri, keď má konkrétna hodnota parametra minimálny vplyv na úspešnosť



parameter	popis	citlivosť
	hlavný algoritmus	
dif1upper	horný prah prvého Cannyho detektora	stačil dosť vysoký
dif2upper	horný prah druhého Cannyho detektora	stačil dosť vysoký
dif1lower	dolný prah prvého Cannyho detektora	mierna závislosť
dif2lower	dolný prah druhého Cannyho detektora	mierna závislosť
blurSize	šírka Gaussovho filtra	silná závislosť
shapeTheshold	prah spojenia bodov pred vyplňaním útvarov	stredná závislosť
searchSpread	šírka oblasti hľadania maxima pri hľadaní konca	mierna závislosť
minSearchSpread	minimálna šírka oblasti hľadania maxima	stačila dosť malá
spreadIncrease	rozširovanie oblasti pri veľkom rozptyle	mierna závislosť
spreadDecrease	zuzovanie oblasti pri malom rozptyle	mierna závislosť
maxBad	minimálne skóre pre prítomnosť oblúka	stredná závislosť
minGood	maximálne skóre pre začatie novej časti oblúka	stredná závislosť
unbalancePenalty	penalizácia za nerovnakú výšku konca oblúka na koncoch	stredná závislosť
	záložný algoritmus	
fallbackIntersWidthMax	rozdelenie rozsahu možných stredov podľa šírky	stačil dosť vysoký
fallbackIntersHeightMax	rozdelenie rozsahu možných stredov podľa výšky	stačil dosť vysoký
fallbackIntersRadiusMax	rozdelenie rozsahu možných stredov podľa polomera	stačil dosť vysoký
fallbackAngleEnd	povolená odchýlka pre uznanie dotyčnice	mierna závislosť
fallbackAngleMin	dĺžka koncovej časti oblúka	mierna závislosť
fallbackAngles	počet uhlov, na ktoré bol oblúk rozdelený	stačil dosť vysoký

Tab. 4.2: Zoznam doladovaných parametrov

## 5 Automatické zhodnotenie kvality výsledku

Pre rozsiahlejšie testovanie programu bolo nutné, aby program dokázal sám posúdiť, či kontaktný uhol určil správne. Táto funkcionality bola užitočná aj pre použitie náhradného algoritmu v prípade zlyhania. Tiež by to umožnilo používateľovi nastaviť, aby sa nesprávne výsledky ani nezobrazovali.

Pri výbere výsledku z viacerých možných stačilo jednoduché kritérium založené na súvislosti a súmernosti nájdeného oblúka, pretože zvyčajne bolo jedno riešenie správne a ostatne obsahovali závažné nedostatky.

### 5.1 Kritériá na overenie správnosti výsledku

Niektoré nesprávne výsledky bolo možné identifikovať pomerne jednoducho. Príkladom takéhoto výsledku sú prípady, keď niektorý z bodov leží mimo obrázku alebo sú nejaké dva body príliš blízko seba (obrázok 5.1).

Bolo však príliš veľa prípadov, kde bola kvapka nájdená, ale koniec nebol nájdený správne alebo bol namiesto kvapky nájdený útvar podobný kvapke. V takých prípadoch exaktné kritérium neexistovalo a bolo nutné skúšať rôzne kritériá a o výsledku hlasovať (obrázky 5.2 a 5.3).

Každé kritérium vypočítalo nejakú numerickú hodnotu, ktorej prah bol nastavený tak, aby bola čo najvyššia úspešnosť určenia správnosti výsledku pomocou tohoto kritéria samotného.

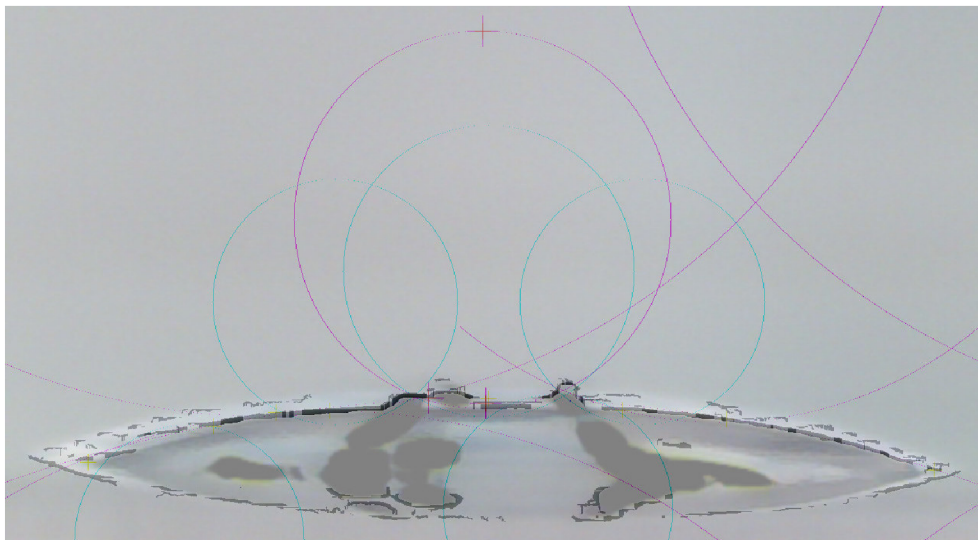
Kritériá mali rôznu úspešnosť (konkrétne úspešnosti sú v tabuľke 5.1). Uvedené sú od najúspešnejšieho k najmenej úspešnému.

#### 5.1.1 Kritérium príslušnosti

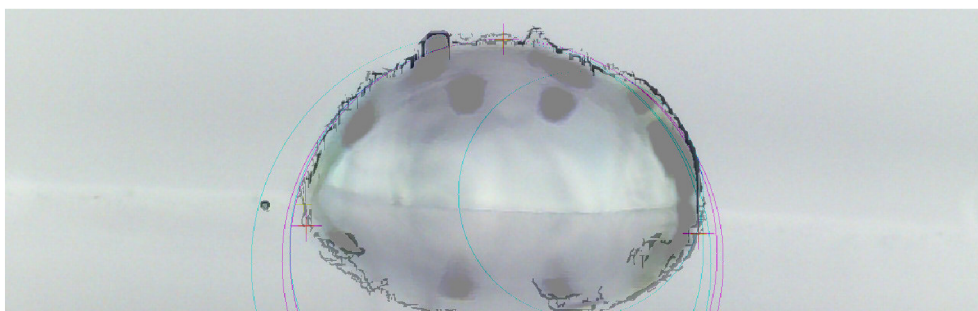
Ukázalo sa, že kritérium príslušnosti bolo najlepšie kritérium určenia správnosti nájdenia kružnice. Kritérium bolo založené na transformácii z kapitoly 4.5.4.

Bol počítaný počet bodov, kde bol v bodoch v blízkosti kružnice vyšší gradient ako v bodoch ďalej od nej (do istej vzdialenosti). Potom bol vypočítaný pomer počtu takýchto bodov v časti patriacej oblúku kvapky k počtu bodov v časti nepatriacej oblúku kvapky. Obidve časti boli normalizované dĺžkou týchto oblúkov.

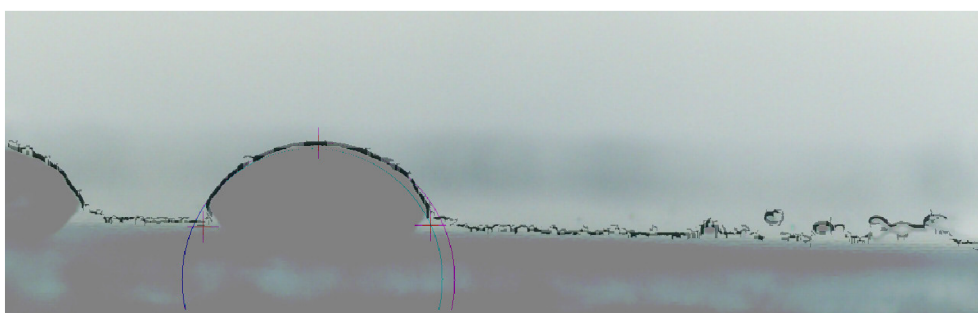
Toto kritérium hodnotilo sústredenosť zmeny jasú na obode kružnice v časti označenej za oblúk a porovnávalo ho so sústredenosťou zmeny jasú v časti, kde sa kružnica nachádzať nemala. Výsledú hodnotu znižovala aj prítomnosť kružnice v oblasti, kde sa podľa riešenia nenachádza, aj málo výrazné sústredenie zmeny jasú v časti, kde sa kružnica podľa riešenia nachádza.



Obr. 5.1: Lahko odhaliteľná chyba (pre lepšiu viditeľnosť sú farby invertované)



Obr. 5.2: Ťažko odhaliteľná chyba 1 (pre lepšiu viditeľnosť sú farby invertované)



Obr. 5.3: Ťažko odhaliteľná chyba 2 (pre lepšiu viditeľnosť sú farby invertované)

Ak bol pomer aspoň 10 (hodnota nájdená empiricky), tak riešenie bolo označené za správne.

Bol použitý aj variant, v ktorom sa nenormalizovalo dĺžkou oblúkov, pretože sa takéto body nemôžu nachádzať v častiach kružnice, ktoré ležia mimo obrázka. V tomto prípade mal byť pomer aspoň 5. Úspešnosť tejto variácie bola podstatne nižšia.

### **5.1.2 Kritérium približného odhadu kontaktného uhla**

Toto kritérium bolo založené na hľadaní okrajov v obraze transformovanom do polárnych súradníc, čo bol jeden z menej úspešných algoritmov hľadania koncov oblúku kvapky (kapitola 4.5.4). Pri tomto kritériu bol odhadnutý kontaktný uhol tak, že podobne ako v kapitole 4.5.4 bol nájdený taký interval bodov na kružnici, aby bol maximalizovaný rozdiel súčtu hodnôt zmeny jasu v blízkosti týchto bodov a súčtu vzdialenejších bodov. Ak bol takto nájdený kontaktný uhol dostatočne blízky nameranému, tak bolo riešenie označené za správne.

### **5.1.3 Kritérium šumu**

Toto kritérium bolo založené na myšlienke, že v časti, kde je prítomný oblúk, sa body s najvyššou hodnotou rozdielu farieb nachádzajú blízko seba po obvode kružnice a v častiach, kde je oblúk neprítomný, sa nachádzajú náhodne. Preto algoritmus prechádzal kružnicu po obvode a boli počítané vzdialenosti maxím v blízkosti kružnice od stredu a rozdiely medzi dvomi vedľa seba sa ukladali samostatne pre časť na nájdenom oblúku a pre časť mimo nájdený oblúk. Ich súčty boli normalizované dĺžkami častí oblúkov. Ak bol pomer rozdielov mimo kružnice a rozdielov na kružnici väčší ako 5, tak bolo riešenie označené za správne.

### **5.1.4 Kritérium sústredenia**

Podobne ako v kritériu príslušnosti a kritériu približného odhadu kontaktného uhla bol počítaný rozdiel hodnôt rozdielu jasu v blízkosti kružnice a ďalej od kružnice, do istej vzdialenosti. V tomto kritériu boli sčítané hodnoty týchto rozdielov v časti patriacej oblúku a v časti nepatriacej oblúku. Súčet v časti nepatriacej oblúku bol vynásobený konštantou. Ak bol rozdiel týchto hodnôt pre tieto časti po normalizácii dĺžkou častí väčší ako istá minimálna hodnota, tak bolo riešenie označené za správne.

Bol použitý aj variant, kde bol počítaný pomer týchto súčtov namiesto rozdielu.

### 5.1.5 Kritérium súvislosti

V tejto časti, podobne ako pri kritériu šumu algoritmus prechádzal po obvode kružnice a hľadal maximá v jej blízkosti. Prechádzal sa iba po časti, ktorá bola podľa výsledku súčasťou oblúka. Ak tieto maximá boli dostatočne blízko kružnice, zvyšoval skóre, ak boli ďalej, prudko ho znižoval. Ak bolo skóre vyššie ako hranica, tak bolo riešenie označené za správne.

### 5.1.6 Kritérium znečistenia

Pri tomto kritériu sa vychádzalo z predpokladu, že okolo kvapky by sa nemali nachádzať nijaké podstatné útvary. Preto boli sčítané hodnoty rozdielov jasu v oblastiach okolo kvapky do istej vzdialenosti a ak bol tento súčet po normalizácii nižší ako prahová hodnota, riešenie bolo označené za správne. Úspešnosť tohoto kritéria bola nízka.

## 5.2 Hlasovanie medzi kritériami

Na zlúčenie výsledkov jednotlivých kritérií bolo použité hlasovanie. Ako váhy hlasov jednotlivých kritérií nebolo možné použiť ich úspešnosť, pretože medzi niektorými bola silná korelácia. Namiesto toho boli nájdené tak, že boli pri počítaní výsledkov zo skúšobných dát vypisované výsledky všetkých kritérií zhromaždené z niekoľkých behov, vložené to tabuľkového procesora s implementovaným výpočtom výsledku hlasovania aj správnosti a konkrétne váhy boli nájdené ručne. Úspešnosti jednotlivých sád parametrov sú v tabuľke 5.1.

Boli spracovávané iba prípady, ktoré neboli evidentne nesprávne. Okrem toho boli za úspešné považované aj prípady, kde riešenie bolo úplne nesprávne, ale výsledok bol náhodou blízky správnemu. Preto skutočná úspešnosť bola vyššia ako hodnoty uvedené v tabuľke, ktoré slúžia iba na porovávanie.

Sada parametrov A používala iba normalizované kritérium príslušnosti. Úspešnosť bola vysoká, avšak nesprávne riešenia boli vyhodnocované ako správne (true negative) oveľa častejšie ako boli správne riešenia vyhodnocované ako nesprávne (false positive), čo znemožňovalo merať štatistiku úspešnosti a tiež bránilo použitiu záložnej metódy v prípade neúspechu. Sada parametrov B bola odhadnutá pri implementácii kritérií, správne riešenia boli vyhodnocované ako nesprávne podobne často ako boli nesprávne riešenia vyhodnocované ako správne. Sada parametrov C mala najvyššiu úspešnosť, ale nesprávne riešenia boli vyhodnocované ako správne príliš často. Sada parametrov D mala iba mierne nižšiu úspešnosť ako sada C, avšak pravdepodobnosti nesprávneho zaradenia medzi správne riešenia a medzi nesprávne riešenia boli podobné.

		Sady váh			
kritérium	samotné kritérium	A	B	C	D
kritérium príslušnosti	87%	1	4	3	3
kritérium príslušnosti nenorm.	86%	0	0	2	0
kritérium odhadu kontakt. uhla	79%	0	3	3	3
kritérium šumu	77%	0	3	2	2
kritérium sústredenia (rozdiel)	72%	0	1	0	0
kritérium sústredenia (pomer)	75%	0	1	0	0
kritérium súvislosti	63%	0	2	2	2
kritérium znečistenia	56%	0	1	0	0
minimálny súčet		1	6	4	4
úspešnosť		88%	87%	89%	89%

Tab. 5.1: Úspešnosti jednotlivých kritérií a ich váhy pri hlasovaní

V 60% prípadov bol súčet aspoň 8 pri sade parametrov D, a vtedy sa stávalo iba výnimočne, že bol výsledok napriek tomu nesprávny (vždy sa jednalo o prípady s nejasným koncom oblúka, kde výsledok nebol príliš vzdialený od správneho, napríklad na obrázkoch 5.2 a 5.3).

### 5.3 Zvolené riešenie

Ako riešenie bola nakoniec použiť sada parametrov D.

Pretože napriek sporným prípadom vo väčšine prípadov bolo možné takmer isto riešenie označiť za správne alebo za nesprávne, program mohol s riešením vrátiť štyri hodnotenia - *isto nesprávne*, *pravdepodobne nesprávne*, *pravdepodobne správne* a *správne*. Rozsahy pre jednotlivé hodnotenia sú uvedené v tabuľke 5.2.

súčet	hodnotenie
0	isto nesprávne
1-3	pravdepodobne nesprávne
4-7	pravdepodobne správne
8+	správne

Tab. 5.2: Hodnotenie výsledku podľa súčtu hlasov s váhami zo sady D

## 5.4 Využitie v algoritme

Schopnosť algoritmu zhodnotiť kvalitu svojho riešenia umožňovala vybrať správnejšie riešenie spomedzi výsledkov viacerých algoritmov. Kvôli nezanedbateľnému množstvu prípadov, kde bol nesprávny výsledok označený za správny, bolo možné takto vybrať iba z algoritmov, ktoré sa dopĺňali, a preto bol použitý iba jeden záložný algoritmus, metóda dotyčníc.

Stávalo sa však, že nesprávne riešenie malo lepšie hodnotenie ako správne. Preto bol otestovaný aj variant, pri ktorom výsledok záložného, menej úspešného postupu, mohol nahradiť výsledok hlavného, úspešnejšieho, iba ak mal výsledok menej spoľahlivého postupu výrazne lepšie hodnotenie ako výsledok spoľahlivejšieho postupu, t.j. buď mal menej spoľahlivý postup riešenie ohodnotenú ako *správne* a spoľahlivejší postup riešenie ohodnotenú ako *pravdepodobne nesprávne* alebo mal spoľahlivejší postup riešenie ohodnotenú ako *isto nesprávne* a menej spoľahlivý mal lepšie hodnotenie. Avšak tento prístup mal horšiu úspešnosť ako variant, kde bol použitý výsledok záložného postupu ak mal lepšie hodnotenie ako výsledok hlavného postupu.

Tento postup tiež spôsobil, že sa stávalo častejšie, že algoritmus označil nesprávne riešenie za správne (tabuľka 5.3). Algoritmus má 81% pravdepodobnosť správne určiť kontaktný uhol, 6% riešení označuje za nesprávne, aj keď sú správne a 7% riešení označuje za správne, aj keď sú nesprávne (takže 9% z označených za správne sú nesprávne a 33% z označených za nesprávne sú správne).

vyriešené	ohodnotených ako úspešné	ohodnotených ako neúspešné
úspešne	75%	6%
neúspešne	7%	12%

Tab. 5.3: Hodnotenie výsledkov pri úspešnom spracovaní a pri neúspešnom spracovaní, v percentách z celku

## 5.5 Zhrnutie

Aby program dokázal určiť, či sa podarilo nájsť kvapku a kontaktný uhol, bol vytvorený algoritmus na automatickú kontrolu výsledkov. Algoritmus testoval výsledok pomocou rôznych kritérií a o správnosti potom rozhodol pomocou hlasovania. Pretože algoritmus nebol úplne spoľahlivý, výsledky mohli byť okrem *správne* a *nesprávne* aj *pravdepodobne správne* a *pravdepodobne nesprávne*. Toto kritérium bolo využité aj pri výbere výsledku z dvoch algoritmov.

## 6 Automatizovaný test na veľkej vzorke snímok

V rámci tejto časti bola vytvorená sada 11221 fotografií z rôznych experimentov konaných rôznymi ľuďmi z rôznych tímov. Fotografie neboli nijak triedené, aby zodpovedali reálnemu rozdeleniu fotografií vykonávaných na zariadení. Vzorové riešenia neboli zadané a úspešnosť sa posudzovala iba podľa hodnotiacich kritérií. Na tejto zbierke bola v 81% prípadoch hlásená úspešnosť *správne* alebo *pravdepodobne správne*. Ak predpokladáme, že 9% riešení označených za *správne* alebo *pravdepodobne správne* sú nesprávne a 33% riešení označených za *isto nesprávne* alebo *pravdepodobne nesprávne* sú správne (úspešnosť automatického hodnotenia na testovacej sade), tak reálna úspešnosť vychádza 80%.

Čas spracovania jednej fotografie bol priemerne 512 ms na štvorjadrovom procesore Intel s mikroarchitektúrou Kaby Lake s funkciou hyper-threading a frekvenciou 3.7 GHz (fotografie čítané z SSD) a 1025 ms na dvojjadrovom procesore Intel s mikroarchitektúrou Kaby Lake s funkciou hyper-threading a frekvenciou 2.5 GHz (fotografie čítané z HDD). Algoritmus používal menej ako 100 MiB pamäte RAM, takže kapacita RAM nemala vplyv na čas výpočtu.

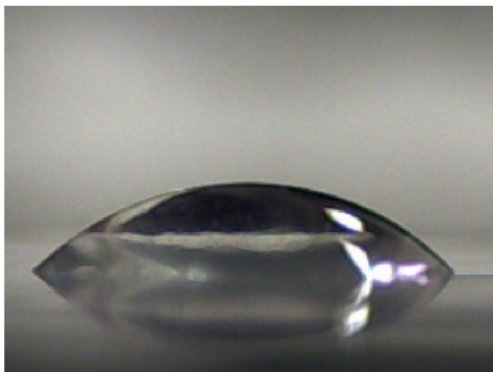
Hlásená úspešnosť algoritmu závisela značne od experimentu, z ktorého pochádzala fotografia. Pri fotografiách z niektorých experimentov bola 100% (ako napríklad obrázok 6.1a). Na výsledkoch niektorých experimentov program pravidelne zlyhával. Zvyčajne za tým bol jedna z troch príčin:

- Kontaktný uhol je extrémne nízky, typicky pod  $10^\circ$  (obrázok 6.1b). V takýchto prípadoch je obtiažne kontaktný uhol určiť aj ručne, pretože tvar kvapky sa výrazne odchyľuje od oblúka kružnice kvôli väčšiemu vplyvu tiaže a povrchových nerovností.
- Koniec oblúka kvapky je nejasný a je ho problém dostatočne presne určiť aj manuálne (obrázok 6.1c). Jedná sa o chybu experimentátora, keď podložka nie je vodorovná.
- Tvar kvapky nie je kruhový kvôli drsnosti materiálu (obrázok 6.1d) alebo zlej manipulácii. V takýchto prípadoch je možné určiť kontaktný uhol manuálne, avšak nie jednoduchým označením bodov konca oblúka a ďalšieho bodu.

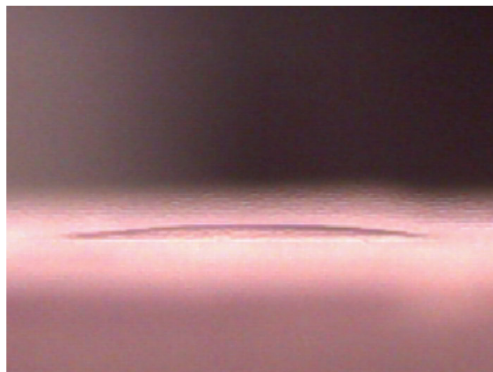
Pravdepodobným dôvodom rozdielu medzi najčastejšími problémami na testovacej sade obrázkov a najčastejšími problémami na reálnej sade obrázkov je, že testovacia sada bola vytvorená tak, aby bola čo najväčšia rozmanitosť fotografií a reálna sada vystihuje pravdepodobnosti výskytu jednotlivých typov fotografií.

Príčinou dlhšieho času vykonávania algoritmu mohla byť odlišnosť vo fotografiách alebo čítanie fotografií z disku (v prípade testovacej sady boli všetky načítané v

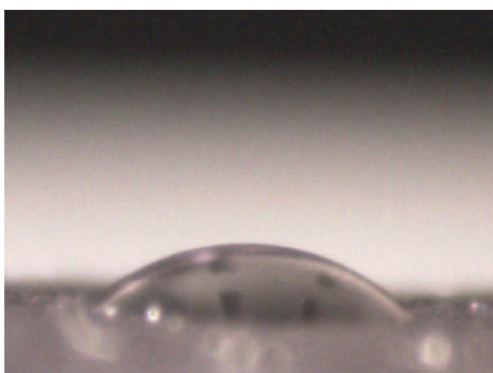




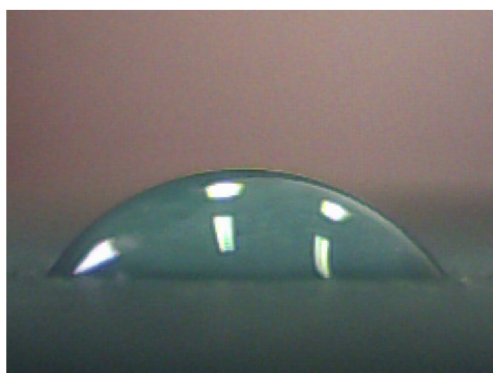
(a) Fotografia z experimentu zo skupiny s takmer 100% úspešnosťou



(b) Fotografia z experimentu zo skupiny s takmer 100% neúspešnosťou kvôli extrémne nízkemu kontaktnému uhlu



(c) Fotografia z experimentu zo skupiny s nízkou úspešnosťou kvôli nejasnému koncu oblúka



(d) Fotografia z experimentu zo skupiny s nízkou úspešnosťou kvôli anomálnemu tvaru kvapky

Obr. 6.1: Ukážky fotografií z testovania na veľkej vzorke (pre lepšiu viditeľnosť sú kvapky orezané)

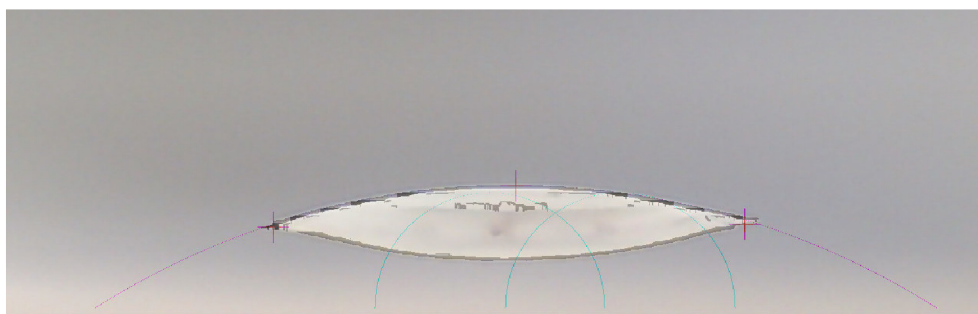
RAM). Pri spracovávaní fotografií z pamäťového zariadenia pripojeného cez USB port bol tento rozdiel väčší.

## 7 Konečné zhodnotenie algoritmu

Na testovacej sade 161 snímok mal algoritmus 81% úspešnosť. Stredná odchýlka od ručne určených výsledkov bola  $1.8^\circ$ . Stredná kvadratická odchýlka od manuálne určených hodnôt bola  $3.5^\circ$ . Stredná kvadratická odchýlka merania s tou istou kvalitou na tom istom substráte s ručným určovaním kontaktného uhla je približne  $4^\circ$  (pri malých kontaktných uhloch je menšia).

Priemerný čas spracovania fotografie bol 434 ms na štvorjadrovom procesore Intel s architektúrou Kaby Lake s funkciou hyper-threading a frekvenciou 3.7 GHz. Na dvojjadrovom procesore Intel s architektúrou Kaby Lake s funkciou hyper-threading a frekvenciou 2.5 GHz bol priemerný čas spracovania fotografie 750 ms. Algoritmus používal menej ako 100 MiB pamäte RAM, takže kapacita RAM nemala vplyv na čas výpočtu.

Malé kontaktné uhly nespôsobovali problémy vo výpočte, pretože v týchto prípadoch bol oblúk čiastočne prekrytý výsledkom Houghovej transformácie a výsledok bol následne nájdený fitovaním (obrázok 7.1).



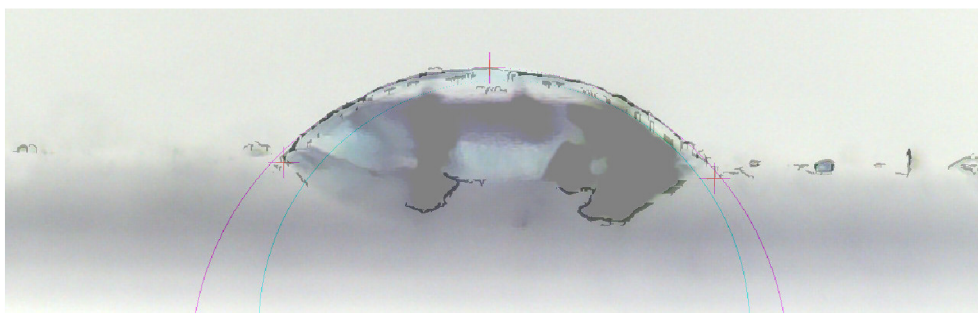
Obr. 7.1: Úspešne vypočítaný kontaktný uhol na fotografii kvapky s malým kontaktným uhlom a neviditeľným okrajom podložky (pre lepšiu viditeľnosť sú farby invertované)

Dobre sú zvládané fotografie, kde sa svetelné zdroje alebo substrát odrážali od povrchu kvapky (obrázok 7.2) vďaka filtru odstraňujúcemu objekty vnútri štruktúr.

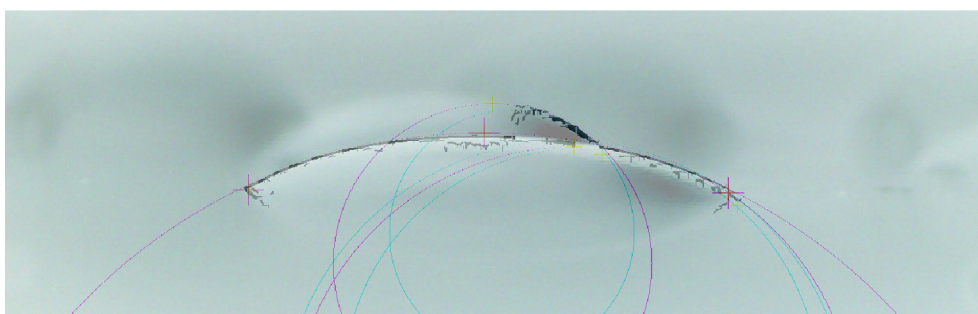
Výpočet prebiehal správne aj na fotografiách, na ktorých bolo viac kvapiek, pretože fotografovaná kvapka bola zvyčajne jediná ostrá a ostatné boli často neúplné, pretože sa nachádzali na okraji fotografie alebo za meranou kvapkou (obrázok 7.3).

Ak aj kvapka na fotografii nemala celkom rovný povrch, zvyčajne bola nájdená správne, avšak kontaktný uhol bol určený nepresne.

Najväčšie problémy pri výpočte boli spôsobené neostrými fotografiami kvapiek, pri ktorých neboli hrany vyznačené Cannyho detektorom (obrázok 7.5). Takmer polovica zlyhaní bola spôsobená práve neostrými fotografiami. Úspešnosť na neostrých



Obr. 7.2: Úspešne vypočítaný kontaktný uhol na fotografii kvapky s množstvom odrazov okolitých svetiel (pre lepšiu viditeľnosť sú farby invertované)



Obr. 7.3: Úspešne vypočítaný kontaktný uhol na fotografii s viacerými kvapkami (pre lepšiu viditeľnosť sú farby invertované)

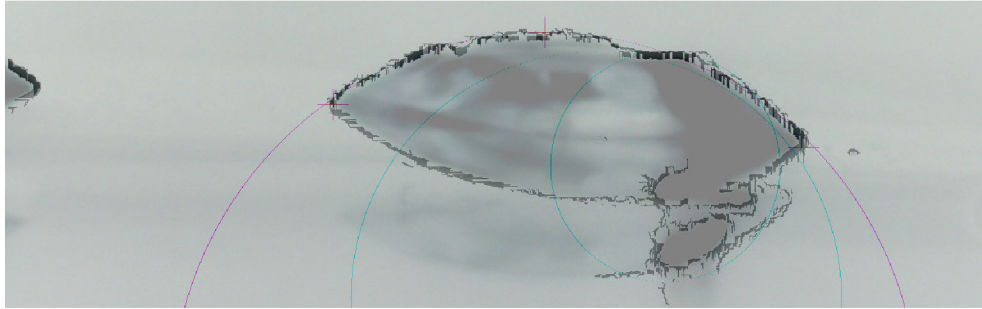
fotografiách bola približne polovičná, v mnohých prípadoch riešenie našiel záložný algoritmus, pretože nepoužíval Cannyho detektor (obrázok 7.6).

Približne 20% zlyhaní bolo spôsobených ostrým okrajom podložky nachádzajúcim sa nad okrajom kvapky, čo spôsobilo, že filter odstraňujúci objekty vnútri štruktúr na kvapke nekorektne zmazal pokračovanie oblúka kvapky. Toto bolo dôsledkom kompromisu, vynechanie tohoto filtra by spôsobilo chybnú interpretáciu rôznych štruktúr na podložke alebo odrazu kvapky ako pokračovanie oblúka. Vo väčšine prípadov tento okraj nebol dostatočne ostrý na to, aby bol zachytený Cannyho detektorom. Povrch podložky by teoreticky mal byť ostrý iba v rovnakej vzdialenosti od kamery ako kvapka.

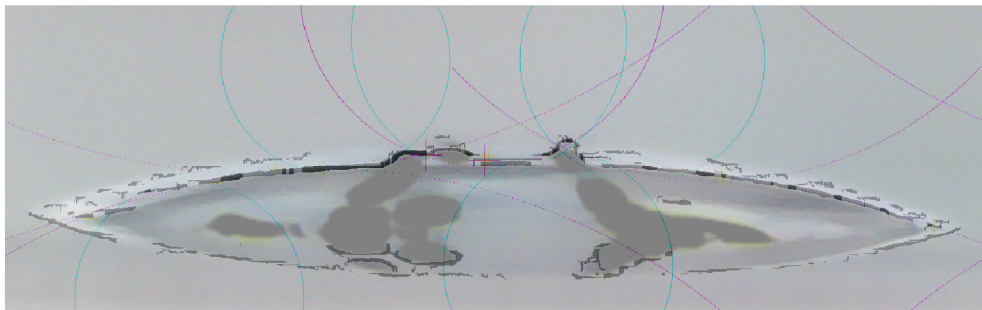
Približne 15% zlyhaní bolo spôsobených odrazom kvapky od substrátu. Tento problém nastával iba ak bol kontaktný uhol blízky  $90^\circ$  a povrch substrátu bol nedostatočne ostrý.

Zvyšné zlyhania boli spôsobené kombináciou rôznych faktorov, ktoré sami o sebe zlyhávanie nespôsobovali.

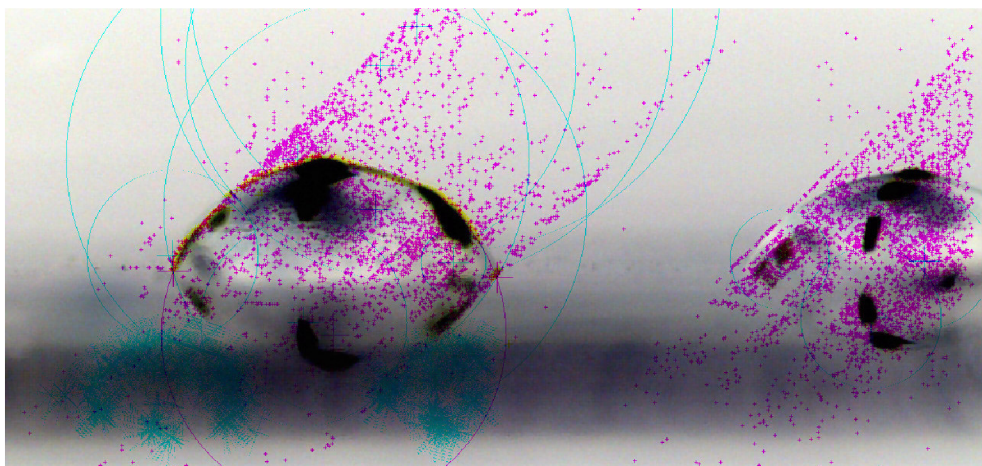
Úspešnosť pri niektorých fotografiách závisela na náhode, pretože závady boli na



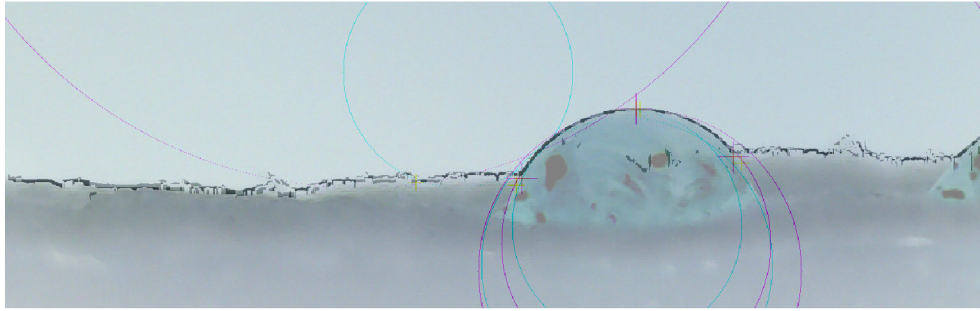
Obr. 7.4: Úspešne ale nepresne vypočítaný kontaktný uhol na fotografii s kvapkou nie celkom kruhového tvaru (pre lepšiu viditeľnosť sú farby invertované)



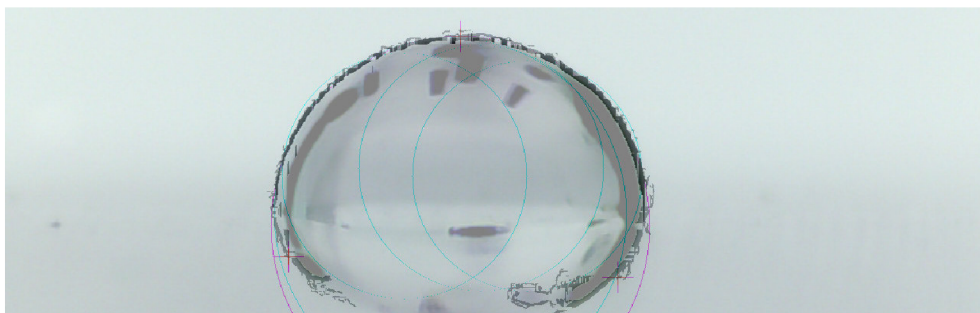
Obr. 7.5: Neúspešné spracovanie neostrej fotografie (pre lepšiu viditeľnosť sú farby invertované)



Obr. 7.6: Neostrá fotografia, ktorej kontaktný uhol bol vypočítaný vďaka záložnému algoritmu (pre lepšiu viditeľnosť sú farby invertované)



Obr. 7.7: Neúspešné spracovanie fotografie kvapky, pri na ktorej sa našla hrana nad koncom oblúka (pre lepšiu viditeľnosť sú farby invertované)

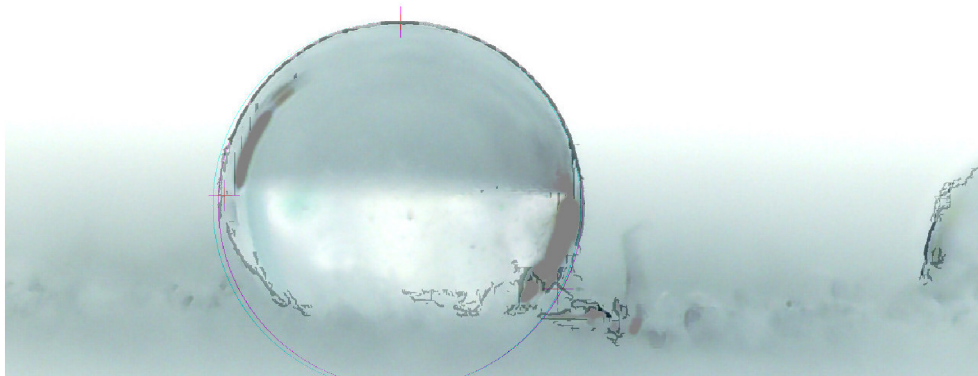


Obr. 7.8: Neúspešné spracovanie fotografie kvapky, ktorá sa odrážala od substrátu (pre lepšiu viditeľnosť sú farby invertované)

hranici zlyhania (7.9).

Testovanie na rozsiahlej reálnej sade fotografií malo rovnakú úspešnosť, zlyhania však nastávali zväčša z iných príčin. Týmito príčinami bolo hlavne nesplnenie predpokladov teórie, čo spôsobilo, že kvapka nemala kruhový tvar a chyby experimentátora. Takéto prípady spôsobili približne 70% zlyhaní. V testovacej sade sa takéto fotografie nenachádzali, pretože bol problém určiť kontaktný uhol manuálne.

Program možno ďalej zlepšovať vo viacerých oblastiach. Fotografie kvapiek s extrémne nízkym kontaktným uhlom sa vôbec nenachádzali v testovacej sade a preto program nebol tvorený tak, aby takéto prípady riešil, kvôli čomu často zlyhával na reálnej sade fotografií. Fitovanie kružníc aj elíps by mohlo takéto prípady riešiť. Hodnotenie úspechu algoritmu z hodnôt kritérií je zaťažené resubstitučnou chybou a mohlo by byť lepšie, ak by bolo využité strojové učenie. Nájdenie ďalších kritérií pre odhad správnosti výsledku by mohlo umožniť výber z výsledkov väčšieho množstva algoritmov, prípadne toho istého algoritmu s výrazne inými parametrami. Väčšinu času výpočtu zaberá fitovanie, optimalizáciou tejto časti by bolo možné algoritmus výrazne zrýchliť.



Obr. 7.9: Príklad fotografie kvapky, ktorej kontaktný uhol sa podarí vypočítať iba s istou pravdepodobnosťou (pre lepšiu viditeľnosť sú farby invertované)

## 8 Záver

Dôležitou vlastnosťou mnohých látok je ich povrchové napätie. Povrchové napätie je zvyčajne určované meraním kontaktných uhlov kvapiek kvapalín s dobre známymi vlastnosťami na ich povrchoch. Kontaktný uhol sa určuje z fotografie kvapky, buď manuálne alebo v prípade dobrých svetelných podmienok automatizovane. Dosiahnutie vhodných svetelných podmienok však značne zvyšuje výrobnú cenu zariadenia. Predmetom tejto diplomovej práce bolo štúdium metód použiteľných na automatické nájdenie kvapky na fotografii za menej vyhovujúcich svetelných podmienok a návrh a test metódy pre výpočet kontaktného uhla.

Prvá časť práce sa venuje problematike merania povrchového napätia a významu kontaktného uhla pri jeho meraní. Okrem toho popisuje zariadenie na meranie kontaktného uhla SeeSystem od firmy Advex Instruments, pre ktoré bol softvér vytvorený. SeeSystem kvôli minimalizácii ceny nemá nijaké prostriedky na kontrolu svetelných podmienok, preto je automatizované hľadanie kontaktných uhlov náročné.

V druhej časti práce sú vysvetlené rôzne metódy užitočné pri určení kontaktného uhla kvapky na fotografii, ako je Cannyho detektor, Houghova transformácia, metóda najmenších štvorcov alebo zhluková analýza.

Tretia časť popisuje tvorbu sady 161 rozmanitých referenčných fotografií kontaktných uhlov s manuálne určenými výsledkami na overenie, či výpočet došiel k správnejmu výsledku. V sade bola viac ako polovica fotografií s rôznymi vadami, ktoré komplikovali určenie kontaktného uhla, ako neostrosť fotografií, odrazy svetelných zdrojov od kvapiek, neostré kvapky v pozadí a podobne.

Štvrtá časť sa venuje návrhu a implementácii samotného algoritmu na výpočet kontaktných uhlov. Na začiatku bol vyskúšaný bežný algoritmus na hľadanie kružníc alebo elíps implementovaný vo voľne dostupných knižniciach OpenCV a bolo zistené, že na testovacej sade fotografií algoritmus uspel iba výnimočne. Kvôli tomu bolo nutné vytvoriť vlastné algoritmy.

Fotografia je spočiatku filtrovaná pomocou Cannyho detektora a rôznych ďalších filtrov pre čistenie vnútra kvapiek, vyplňanie súvislých oblastí a smerové rozdiely jas.

Hlavný algoritmus na fotografiu najprv aplikuje Cannyho detektor. Po ňom odstráni odrazy na kvapke pomocou upraveného algoritmu na hľadanie ciest. Na výsledok aplikuje algoritmus na vyplňanie dutých objektov. Následne fotografiu výrazne zmenší a na takúto zmenšenú fotografiu použije Houghovu transformáciu s prehľadávaním celého priestoru a nájde potenciálne polohy kružníc. Potom hľadá maximá Houghovej transformácie v okolí týchto polôh a získa upresnené odhady polôh kružníc. V prípade, že je týchto odhadov príliš veľa, zníži ich počet pomocou zhlukovej analýzy. Potom získa presné polohy kružníc pomocou algoritmu, ktorý maximali-

zuje súčet hodnôt v pixeloch, cez ktoré prechádza kružnica, postupnými úpravami polomeru a súradníc stredu. Konce oblúkov nájde pomocou algoritmu založeného na sledovaní čiary, čím tiež zistí, ktorá z nájdených kružníc je skutočná kvapka. Úspešnosť algoritmu na testovacej sade bola 76%. Tvorba tohoto algoritmu trvala približne pol roka.

K dispozícii je aj záložný algoritmus, ktorý je od hlavného algoritmu veľmi odlišný. Na fotografii vypočíta smerové detekcie hrán a nájde na nich lokálne maximá. Predpokladá sa, že sa jedná hlavne o miesta, kde oblúk kvapky pretínajú dotyčnice. Ďalej je využitý poznatok, že kolmice na dotyčnice v bodoch dotyku sa pretínajú v strede kružnice. Tým sú získané odhady stredu kružnice. Odhady sú spresnené tiež pomocou algoritmu maximalizujúceho súčet hodnôt v pixeloch na kružnici zmenami súradníc stredu a polomeru. Za koniec oblúka je považovaná časť, kde sa prestanú vyskytovať na oblúku lokálne maximá. Algoritmus má síce menšiu úspešnosť ako hlavný algoritmus, avšak dokáže správne nájsť riešenie v mnohých prípadoch, kde hlavný algoritmus zlyháva. Tvorba tohoto algoritmu trvala približne tri mesiace.

Okrem týchto algoritmov boli vyskúšané rôzne iné algoritmy, ktoré však mali nižšiu úspešnosť na všetkých typoch fotografií, napríklad gradientná Houghova transformácia, hľadanie roviny podložky pomocou Houghovej transformácie, hľadanie úsečiek po transformácii kružnice do polárnych súradníc alebo hľadanie elíps. Tvorba týchto algoritmov trvala približne štyri mesiace.

Rôzne numerické parametre použité v algoritmoch boli doladované automatizovanými výpočtami závislostí úspešnosti na týchto parametroch.

Piata časť popisuje algoritmus pre hodnotenie správnosti určenia kontaktného uhlu. Algoritmus hodnotí výsledok pomocou niekoľkých kritérií, z ktorých sa konečné rozhodnutie určí hlasovaním. Tento výsledok umožňuje rozhodnúť, kedy použiť výsledok záložného algoritmu namiesto hlavného, čo zvýšilo úspešnosť na testovacej sade na 81%. Hodnotenie je okrem toho použité aj pre automatizované testovanie na väčších vzorkách a tiež umožňuje používateľovi nastaviť, aby sa úplne zlé výsledky ani nezobrazovali. Hodnotenie výsledku nie je vždy správne, 6% riešení označuje za nesprávne, aj keď sú správne (false positive) a 7% riešení označuje za správne, aj keď sú nesprávne (true negative). Väčšinu riešení však dokáže s istotou označiť za správne alebo nesprávne.

Šiesta časť hovorí o rozsiahlom teste metódy na sade 11221 fotografií z rôznych experimentov. Tieto fotografie mali približne také pravdepodobnostné rozdelenie ako budú mať fotografie spracovávané týmto algoritmom. Správnosť riešení bola určená algoritmom z predchádzajúcej kapitoly. Úspešnosť bola 80%. Algoritmus zlyhával hlavne pri fotografiách kvapiek, ktoré nemali celkom kruhový tvar kvôli porušeniu predpokladov teórie a je obtiažne ich spracovať aj manuálne (preto takéto prípady neboli použité v testovacej sade). Pri fotografiách z takýchto experimentov bola



úspešnosť často blízka 0%. Naopak, pri fotografiách z niektorých experimentov, kde mali fotografie vyhovujúce vlastnosti, bola úspešnosť takmer 100%.

Siedma časť hodnotí úspešnosť, presnosť a rýchlosť algoritmu a analyzuje, v ktorých prípadoch algoritmus funguje správne a v ktorých má ťažkosti. Na testovacej sade mal algoritmus úspešnosť 81% a strednú kvadratickú odchýlku od vzorových riešení  $3.5^\circ$ . Opakované meranie kontaktného uhla za rovnakých podmienok s manuálnym určením kontaktného uhla malo strednú kvadratickú odchýlku  $4^\circ$ . Výpočet jedného kontaktného uhla trval priemerne 450 ms na modernom desktopovom procesore a priemerne 750 ms na modernom úspornom laptopovom procesore.

Algoritmus dokázal s veľkou úspešnosťou spracovať fotografie, na ktorých bolo viac kvapiek, fotografie kvapiek s nerovným povrchom, fotografie s odrazmi zdrojov svetla na povrchoch kvapiek, fotografie kvapiek s nízkymi kontaktnými uhlami (pokiaľ neboli také nízke, že vplyv tiažovej sily na tvar kvapky nebol zanedbateľný, čo spôsobovalo, že tvar kvapky nebol kruhový) alebo fotografie, na ktorých povrch pod kvapkou nebolo vidieť. Neostré kvapky dokázal často vyriešiť záložný algoritmus, ak hlavný zlyhal, pretože Cannyho detektor nenašiel správne obrysy kvapky. Tento problém sa však zriedkakedy vyskytoval na reálnej sade fotografií. Ďalšou časťou príčinou zlyhania bolo nesprávne nájdenie konca oblúka, keď sa nesúvisiace objekty alebo odrazy kvapky od povrchu ležiace na kružnici mylne považovali za pokračovanie kruhu. Program zlyhával aj kvôli kombináciám faktorov, ktoré sami o sebe zlyhanie nespôsobovali.

Program bol tiež skompilovaný do dynamickej knižnice .dll, vďaka čomu je možné ho použiť iným programom asynchrónnym volaním s adresami, kam zapísať výsledok. Softvér zariadenia SeeSystem bol modifikovaný tak, že využíva túto knižnicu na výpočet kontaktného uhla. Tým sa odstraňuje ručné spracovávanie výsledkov, ktoré zaberá približne polovicu času stráveného meraním.

Riešenie je možné ďalej zlepšovať. Väčšinu času strávenú výpočtom zaberá algoritmus spresňujúci polohu stredu a polomer a akákoľvek optimalizácia tejto časti by sa prejavila na dobe výpočtu. Nájdenie ďalších kritérií a aplikácia strojového učenia na výsledky týchto kritérií by mohla zlepšiť schopnosť algoritmu rozhodnúť o správnosti riešenia, čo by umožnilo použitie niektorých z neperspektívnych postupov ako ďalšie záložné riešenia. Testovanie na väčšej vzorke s manuálne určenými výsledkami by umožnilo nájsť presnejšie hodnoty číselných konštánt použitých v algoritme. Tieto kroky sú však časovo náročné a úspešnosť výpočtu by sa pravdepodobne zvýšila iba mierne.

# Literatúra

- [1] *Contact angle measurement and contact angle interpretation*, D. Y. Kwok, A. W. Neumann. DOI: 10.1016/S0001-8686(98)00087-6. ISSN 0001-8686.
- [2] *Medical Image Processing, Reconstruction and Restoration: Concepts and Methods*, J. Jan. ISBN 0-8247-5849-8
- [3] *Surface Tension and Water*, stránky The USGS Water Science School [online]. [cit. 23.12.2017]. Dostupné z URL: <<https://water.usgs.gov/edu/surface-tension.html>>.
- [4] *Contact angle*, stránky firmy Cruss Scientific [online]. [cit. 22.12.2017]. Dostupné z URL: <<https://www.kruss-scientific.com/services/education-theory/glossary/contact-angle/>>.
- [5] *Advex Instruments - SeeSystem*, oficiálne stránky výrobcu [online]. [cit. 6.10.2017]. Dostupné z URL: <<http://www.advex-instruments.cz>>.
- [6] *Studium volné povrchové energie plazmaticky opracovaných povrchů materiálu*, Mgr. Ing. Jiří Bilík, bakalárska práca [online]. [cit. 5.1.2018]. Dostupné z URL: <[https://is.muni.cz/th/35486/prif\\_b/Bakalarka-1\\_stat.pdf](https://is.muni.cz/th/35486/prif_b/Bakalarka-1_stat.pdf)>.
- [7] *Method according to Zisman*, stránky firmy Cruss Scientific [online]. [cit. 9.5.2018]. Dostupné z URL: <<https://www.kruss-scientific.com/services/education-theory/glossary/method-according-to-zisman/>>.
- [8] *Equation of state*, stránky firmy Cruss Scientific [online]. [cit. 9.5.2018]. Dostupné z URL: <<https://www.kruss-scientific.com/services/education-theory/glossary/equation-of-state/>>.
- [9] *Owens, Wendt, Rabel and Kaelble (OWRK) method*, stránky firmy Cruss Scientific [online]. [cit. 9.5.2018]. Dostupné z URL: <<https://www.kruss-scientific.com/services/education-theory/glossary/owens-wendt-rabel-and-kaelble-owrk-method/>>.
- [10] *Surface Tension Components and Molecular Weight of Selected Liquids*, stránka firmy Accu Dyne Test [online]. [cit. 5.1.2018]. Dostupné z URL: <[https://www.accudynetest.com/surface\\_tension\\_table.html](https://www.accudynetest.com/surface_tension_table.html)>.
- [11] *Image Filtering*, stránka profesora Rangachara Kasturiho, University of South Florida [online]. [cit. 9.5.2018]. Dostupné z URL: <[http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision\\_Chapter4.pdf](http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter4.pdf)>.

- [12] *Gaussian Filtering*, text k prednáške na University of Auckland [online]. [cit. 22. 10. 2017]. Dostupné z URL: <[https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering\\_1up.pdf](https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf)>.
- [13] *Prewitt Operator*, návod na používanie na stránke tutorialspoint [online]. [cit. 22. 10. 2017]. Dostupné z URL: <[https://www.tutorialspoint.com/dip/prewitt\\_operator.htm](https://www.tutorialspoint.com/dip/prewitt_operator.htm)>.
- [14] *Sobel Edge Detector*, text k prednáške na University of Edinburgh [online]. [cit. 22. 10. 2017]. Dostupné z URL: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>>.
- [15] *Canny Edge Detection*, text k prednáške na Harvey Mudd College [online]. [cit. 2. 10. 2017]. Dostupné z URL: <<http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html>>.
- [16] C. Y. Wong et al *A survey on ellipse detection methods*, 2012 IEEE International Symposium on Industrial Electronics [cit 27. 3. 2018]
- [17] W Gander, G. H. Golub, R. Strelbel *Least-Squares Fitting of Circles and Ellipses*, BIT Numerical Mathematics, 1994 [online]. [cit 4. 5. 2018] Dostupné z URL: <<https://www.emis.de/journals/BBMS/Bulletin/sup962/gander.pdf>>.
- [18] *Hough Transform*, stránky o programe HIPR2 [online]. [cit. 23. 12. 2017]. Dostupné z URL: <<https://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>>.
- [19] Danko Antolovic *Review of the Hough Transform Method, With an Implementation of the Fast Hough Variant for Line Detection* [online]. [cit 4. 5. 2018] Dostupné z URL: <<https://www.cs.indiana.edu/ftp/techreports/TR663.pdf>>.
- [20] L.V. Bijuraj *Clustering and its Applications*, Proceedings of National Conference on New Horizons in IT - NCNHIT 2013 [online]. [cit. 23. 12. 2017]. Dostupné z URL: <<http://www.met.edu/Institutes/ICS/NCNHIT/papers/39.pdf>>.
- [21] *UPGMA*, stránky firmy GelQuest [online]. [cit. 23. 12. 2017]. Dostupné z URL: <[http://www.sequentix.de/gelquest/help/upgma\\_method.htm](http://www.sequentix.de/gelquest/help/upgma_method.htm)>.
- [22] *OpenCV library*, oficiálne stránky [online]. [cit. 09. 05. 2018]. Dostupné z URL: <<https://opencv.org>>.

- [23] *Fastest Gaussian Blur (in linear time)*, Ivan Kutsir [online]. [cit. 03. 04. 2018]. Dostupné z URL: <http://blog.ivank.net/fastest-gaussian-blur.html>.

# Zoznam príloh

A Obsah priloženého DVD

74

# A Obsah priloženého DVD

Diplomova praca .....	Koreňový adresár
├─ diplomova_praca.pdf.....	Elektronická verzia diplomovej práce
├─ program_linux.....	program pripravený na použitie na OS Linux
│   └─ findPoints .....	skompilovaný program
│   └─ findPoints.cpp.....	zdrojový kód programu
│   └─ example.bmp.....	obrázok na overenie funkcie programu
├─ program_windows .....	program pripravený na použitie na OS Windows
│   └─ findPoints.exe.....	skompilovaný program
│   └─ findPoints.cpp.....	zdrojový kód programu
│   └─ findPoints.dll...	program skompilovaný pre použitie iným programom
│   └─ findPointsDLL.h.....	záhlavie pre kompiláciu s iným programom
│   └─ example.bmp.....	obrázok na overenie funkcie programu