



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**ZPRACOVÁNÍ OBRAZU ZA ÚČELEM DETEKCE  
LASEROVÉHO PAPRSKU V OTEVŘENÉM PROSTORU**

IMAGE PROCESSING FOR LASER BEAM DETECTION IN OPEN SPACE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**LENKA ŠOKOVÁ**

**VEDOUcí PRÁCE**

SUPERVISOR

**doc. Dr. Ing. DUŠAN KOLÁŘ**

BRNO 2023

## Zadání bakalářské práce



148377

Ústav: Ústav inteligentních systémů (UITS)  
Studentka: **Šoková Lenka**  
Program: Informační technologie  
Specializace: Informační technologie  
Název: **Zpracování obrazu za účelem detekce laserového paprsku v otevřeném prostoru**  
Kategorie: Zpracování obrazu  
Akademický rok: 2022/23

### Zadání:

1. Prostudujte literaturu související se snímáním laserových paprsků v otevřeném prostoru. Dále prostudujte možnosti zpracování obrazu za účelem zvýraznění a detekce čar v obrazu.
2. Seznamte se s možnostmi detekce zdroje laserového paprsku v obraze. Provedte snímání a měření za účelem vytvoření reálných dat.
3. Navrhněte způsob zpracování obrazu za účelem detekce laserového paprsku a jeho následné extrakce a definování jeho trajektorie pomocí vektoru.
4. Implementujte program/y na základě předchozího bodu zadání, včetně experimentálního ověření jeho/jejich funkčnosti.
5. Dosažené výsledky shrňte a diskutujte možná rozšíření.

### Literatura:

- TOMLINSON, Erin, Richard WESTHOFF, Tom REYNOLDS a Brian SAAR. *Aircraft Laser Strike Geolocation System*. In: *17<sup>th</sup> AIAA Aviation Technology, Integration, and Operations Conference* [online]. Denver, Colorado, 2007, s. 2284-2292. ISBN 978-1-5108-4368-4. <https://doi.org/10.2514/6.2017-4389>.
- HLAVÁČ, Václav a Radim ŠÁRA. *Computer Analysis of Images and Patterns*. 6<sup>th</sup> International Conference, CAIP'95. Prague, Czech Republic: Springer, 1995. ISBN 3-540-60268-2.

Při obhajobě semestrální části projektu je požadováno:  
Splnění bodů 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Kolář Dušan, doc. Dr. Ing.**  
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.  
Datum zadání: 1.11.2022  
Termín pro odevzdání: 10.5.2023  
Datum schválení: 3.11.2022

## Abstrakt

Táto bakalárska práca sa zaoberá analýzou obrazu za účelom detekcie laserového lúča v otvorenom priestore. Popisuje základné princípy funkčnosti lasera, jeho vlastnosti a metódy detekcie objektov v obraze. Zároveň implementuje systém, ktorého vstupom je snímka laserového lúča, ktorý detekuje a definuje jeho trajektóriu pomocou vektora.

## Abstract

This Bachelor's thesis deals with image processing in order to detect the laser beam in open space. It describes the basic principles of laser operation, its behaviour and methods of detecting objects in the image. Also, it implements a system whose input is an image of a laser beam, which detects and then calculates its trajectory using a vector.

## Klíčové slová

laser, spracovanie obrazu, hranová detekcia, triangulácia, Houghova transformácia

## Keywords

laser, image processing, edge detection, triangulation, Hough transform

## Citácia

ŠOKOVÁ, Lenka. *Zpracování obrazu za účelem detekce laserového paprsku v otevřeném prostoru*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Dr. Ing. Dušan Kolář

# Zpracování obrazu za účelem detekce laserového paprsku v otevřeném prostoru

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením pána doc. Dr. Ing. Dušana Koláča. Uviedla som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpala.

.....  
Lenka Šoková  
9. mája 2023

## Podakovanie

Rada by som sa poďakovala pánovi doc. Ing. Dušanovi Koláčovi za vedenie mojej bakalárskej práce a za poskytnutie odbornej konzultácie pri jej tvorbe. Ďalej by som sa chcela touto cestou poďakovať všetkým spolupracovníkom, teda celému výskumnému tímu STRaDe a všetkým účastníkom projektu APALER (Aviation protection against low-energy lasers), ktorí so mnou spolupracovali na projekte a nezanedbateľne mi pomáhali počas celého procesu písania práce.

V neposlednom rade, by som chcela vyjadriť veľkú vďaku svojej rodine a kamarátom za ich neoceniteľnú podporu počas celého procesu tvorby a špeciálne by som chcela poďakovať Mgr. Ide Matulíkovej za opravu a finálnu korektúru textu a kamarátke Terke za pomoc so štylistikou a neustále povzbudzovanie.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Lasery</b>	<b>4</b>
2.1	Svetlo . . . . .	4
2.1.1	Fotón . . . . .	4
2.1.2	Spektrálna citlivosť ľudského oka . . . . .	5
2.1.3	Zdroje svetla . . . . .	6
2.2	Princíp laserov . . . . .	6
2.2.1	Laserové komponenty . . . . .	7
2.2.2	Základné procesy v lasery . . . . .	8
2.2.3	Charakteristika laserov . . . . .	9
2.3	Typy laserov a ich využitie . . . . .	10
2.4	Bezpečnostné riziká laserov . . . . .	11
<b>3</b>	<b>Predspracovanie obrazu</b>	<b>13</b>
3.1	Vyhľadanie obrazu . . . . .	13
3.1.1	Lineárne filtrovanie . . . . .	13
3.1.2	Nelineárne filtre vyhladzovania . . . . .	15
<b>4</b>	<b>Detekcia čiar a zdroja v obraze</b>	<b>17</b>
4.1	Detekcia hrán . . . . .	17
4.1.1	Gradient obrazovej funkcie (metódy založené na prvej derivácii jasovej funkcie) . . . . .	18
4.1.2	Laplacián obrazovej funkcie (Metódy založené na druhej derivácii) . . . . .	20
4.1.3	Optimálne hranové detektory . . . . .	20
4.1.4	Výhody a nevýhody jednotlivých hranových detektorov . . . . .	21
4.2	Houghova transformácia . . . . .	22
4.3	Triangulácia . . . . .	23
<b>5</b>	<b>Návrh a implementácia systému</b>	<b>24</b>
5.1	Vytvorenie dátovej sady . . . . .	24
5.1.1	Popis merania 1 . . . . .	24
5.1.2	Popis merania 2 . . . . .	25
5.1.3	Popis merania 3 . . . . .	26
5.2	Návrh systému . . . . .	27
5.2.1	Predspracovanie obrazu . . . . .	28
5.2.2	Detekcia laserového lúča a výpočet trajektórie . . . . .	31
5.3	Implementácia systému a testovanie . . . . .	33

5.3.1	Použité technológie . . . . .	33
5.3.2	Vývojový proces . . . . .	33
5.3.3	Paralelizácia systému . . . . .	37
5.4	Popis výslednej implementácie . . . . .	42
<b>6</b>	<b>Testovanie systému a zhrnutie dosiahnutých výsledkov</b>	<b>43</b>
6.1	Testovanie systému . . . . .	43
6.1.1	Vplyv rozmeru veľkosti obrazu na detekciu laserového lúča . . . . .	43
6.1.2	Porovnanie rýchlosti verzií programu na rôznych systémoch . . . . .	44
6.2	Úspešnosť detekcie laserového lúča . . . . .	47
6.3	Vyhodnotenie . . . . .	48
<b>7</b>	<b>Záver</b>	<b>50</b>
	<b>Literatúra</b>	<b>52</b>
<b>A</b>	<b>Počítačová zostava pre testovanie</b>	<b>54</b>
A.1	Zostava 1 . . . . .	54
A.2	Zostava 2 . . . . .	54
A.3	Zostava 3 . . . . .	54
A.4	Zostava 4 . . . . .	54
<b>B</b>	<b>Obsah pamäťového média</b>	<b>55</b>

# Kapitola 1

## Úvod

Letecká doprava je v súčasnosti, na základe štatistiky, považovaná za jeden z najbezpečnejších druhov prepravy. Toto miesto získala vďaka tomu, že jej bezpečnosti je venovaná nemalá pozornosť, pretože narozdiel od iných druhov transportu, následky nehôd majú zväčša katastrofický charakter a často sa končia úmrtím posádky aj cestujúcich. Najlepším spôsobom, ako týmto katastrofám predísť, je práve prevencia a minimalizácia ohrozujúcich podnetov.

V posledných rokoch je jednou z najčastejších hrozieb leteckej dopravy ožarovanie lietadla laserovými lúčmi. Tento problém sa neustále zvyšuje a výrazne narúša bezpečnosť prepravy. Zásah kabíny laserom môže na čas oslepiť pilota, dezorientovať a pri priamom zásahu do oka aj nenávratne poškodiť jeho zrak. Vďaka týmto aktivitám tak útočník ohrozí nielen pilota, ale aj celú posádku lietadla. Tieto útoky vznikajú najmä v kritických bodoch letu, ako vzlietnutie a pristávanie, kedy pozornosť pilota je vyžadovaná najviac.

Kvôli katastrofickým dopadom ožarovania leteckého personálu laserom je v Českej republike táto aktivita klasifikovaná ako trestný čin obecného ohrozenia. Podľa trestného zákona môže páchateľ dostať tri až osem rokov väzenia.

Vzhľadom na závažnosť situácie, sa hľadajú nové spôsoby a technológie, ako zabrániť týmto útokom a ako rýchlo lokalizovať páchateľa. Jedným zo spôsobov riešenia tohto problému je vytvorenie systému, ktorý pomocou umiestnených kamier v blízkosti pristávacích a vzletových zón zhotoví snímky potenciálneho laserového lúča, ktorý detekuje a následne lokalizuje jeho zdroj.

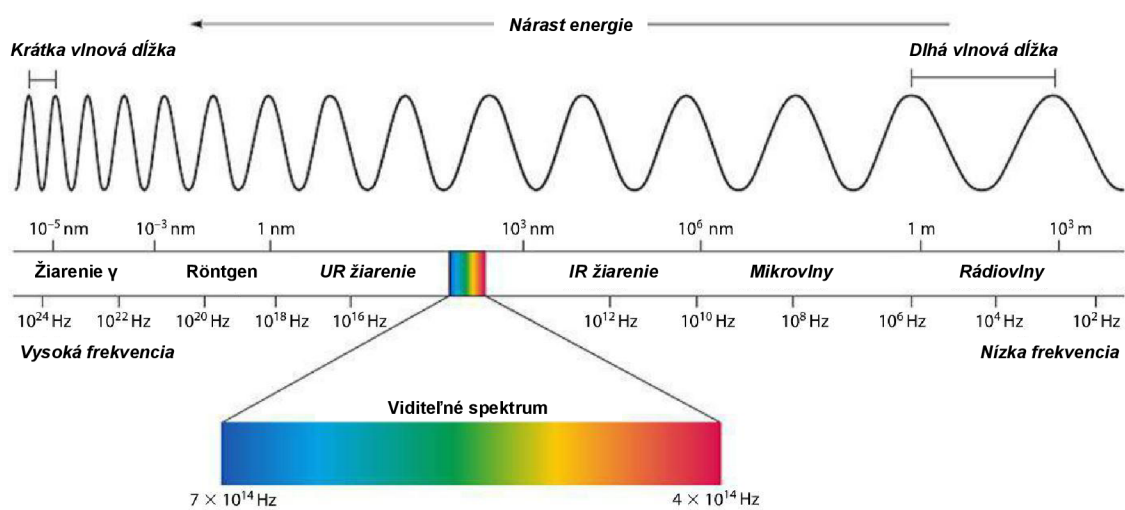
Mojim cieľom je vytvoriť snímky lasera, ktoré by sa čo najviac podobali reálnym dátam. Na základe týchto snímok zabezpečiť implementáciu, ktorá spoľahlivo a rýchlo detekuje laserový lúč v obraze a definuje jeho trajektóriu pomocou vektoru, čo výrazne dopomôže k jeho následnej lokalizácii páchateľa.

# Kapitola 2

## Lasery

### 2.1 Svetlo

Svetlo je viditeľnou časťou elektromagnetického žiarenia, ktoré vzniká pohybom elektricky nabitých častíc, čím vytvára vlny. Oblasti elektromagnetického žiarenia sa líšia svojou vlnovou dĺžkou, prejavmi a účinkami. Jeho rozpätie siaha od dlhých vlnových dĺžok, ako sú rádiové vlny a mikrovlny, až po krátke vlnové dĺžky ako gama lúče. Podľa vlnových dĺžok vo viditeľnom spektre mení svetlo svoju farbu od fialovej po červenú [10]. Elektromagnetické spektrum, ktoré zobrazuje rozpätie jednotlivých častí elektromagnetického žiarenia, je zobrazené na obrázku 2.1.



Obr. 2.1: Elektromagnetické spektrum. *Prevzaté a upravené z [9].*

#### 2.1.1 Fotón

Základnou jednotkou elektromagnetického žiarenia je fotón. Jeho energia je priamo úmerná k frekvencii a nepriamo úmerná vlnovej dĺžke. To znamená, že fotóny s vyššou frekvenciou majú vyššiu energiu a kratšiu vlnovú dĺžku, zatiaľ čo fotóny s nižšou frekvenciou majú nižšiu energiu a dlhšiu vlnovú dĺžku. Príkladom môže byť gama žiarenie, ktoré má vyššiu

frekvenciu a kratšiu vlnovú dĺžku v porovnaní s rádiovými vlnami. Svetlo je tiež charakterizované svojou intenzitou, ktorá závisí od množstva fotónov dopadajúcich na plochu za jednu sekundu. Z toho vyplýva, že dva zdroje svetla s rovnakou vlnovou dĺžkou môžu mať rôznu intenzitu. Táto intenzita svetla klesá v závislosti od vzdialenosti svetelného zdroja [20].

### 2.1.2 Spektrálna citlivosť ľudského oka

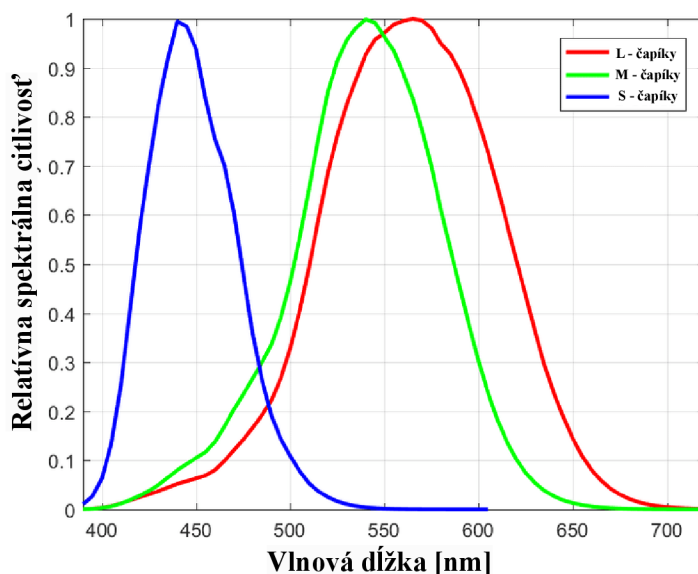
Farebné videnie začína svetlom, ktoré vstupuje do ľudského oka. Láme sa na rohovke a následne prechádza cez šošovku, ktorá pomocou zakrivenia zabezpečuje premietanie zaostreného obrazu na sietnicu v zadnej časti oka.

Sietnica obsahuje dva druhy svetlocitlivých buniek - tyčinky (angl. *rods*) a čapíky (angl. *cones*). Čapíky zabezpečujú farebné videnie a delia sa na tri typy, pričom každý jeden typ dokáže absorbovať iba určitú časť rozsahu farebného spektra.

Týmito typmi sú:

- L (z angl. *long*) typ - absorbuje dlhé vlnové dĺžky, najvyššie hodnoty farebného spektra
- M (z angl. *medium*) typ - absorbuje stredné vlnové dĺžky, stredné hodnoty farebného spektra
- S (z angl. *small*) ty absorbuje krátke vlnové dĺžky, najnižšie hodnoty farebného spektra

Rozdelenie čapíkov v ľudských očiach nie je rovnomerné. Pomer týchto ich typov L:M:S sa pohybuje okolo 60:30:5. Toto rozdelenie sa môže výrazne líšiť od jednotlivca k jednotlivcovi. Z tohto dôvodu ľudské oko nevníma všetky vlnové dĺžky rovnomerne a túto nerovnomernosť popisuje spektrálna citlivosť oka [11].



Obr. 2.2: Relatívna citlivosť oka k elektromagnetickým vlnám viditeľného spektra rôznych vlnových dĺžok. Prevzaté a upravené z [8].

Na obrázku 2.2 je zobrazená spektrálna citlivosť oka na elektromagnetické vlny rôznych dĺžok. Graf ukazuje, ako sa citlivosť jednotlivých typov čapíkov líši pri rôznych vlnových dĺžkach svetla. V strede tohto viditeľného spektra je oblasť, ktorú ľudské oko vníma najcitlivejšie. Pre túto oblasť je vlnová dĺžka okolo 555 nm, ktorá je vnímaná ľudským okom ako zelená farba. Táto vlastnosť spôsobuje, že zdroje zeleného svetla (napr. lasery) s nižším výkonom sú potom viditeľnejšie ako zdroje iných farieb s vyšším výkonom. Hranice tohto viditeľného spektra ľudského oka sa ťažko definujú, pretože citlivosť oka asymptoticky klesá k nule pri dlhých aj krátkych vlnových dĺžkach. Všeobecne sa však uvádza, že ľudské oko vníma farby v rozsahu vlnových dĺžok od okolo 430 nm do 690 nm, pričom citlivosť oka v týchto hraničných hodnotách klesá na 1% svojej maximálnej hodnoty. Toto rozpätie, ale závisí od intenzity svetla, a tak pri dostatočnom osvetlení môže oko vnímať vlnové dĺžky aj za tieto určené hranice [13].

### 2.1.3 Zdroje svetla

Zdrojom svetla je každý žiariaci bod, ktorý vysiela svetelné lúče. Tieto zdroje sa dajú kategorizovať podľa procesu vyžarovania svetla na:

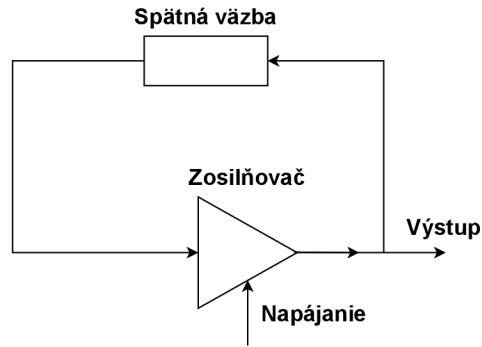
- Prírodné zdroje svetla - slnko, hviezdy, hmyz, a podobne.
- Umelé zdroje svetla
  - Tepelné zdroje - predmety sa zahrievajú na určitú teplotu, pri ktorej začnú vyžarovať svetlo (napríklad žiarovka, sviečka, a podobne).
  - Luminiscencia - svetlo vzniká urýchlením zmien v predmete (vibráciou elektrónov) pomocou prúdu. Príkladmi sú obrazovky, lasery, a podobne.

Charakteristickým znakom každého svetelného zdroja je jeho spektrum, teda graf energie žiarenia, ktorý môže byť rôznej šírky [5].

## 2.2 Princíp laserov

Laser (z angl. Light Amplification by Stimulated Emission of Radiation) je optický oscilátor, ktorý generuje koherentné svetlo prostredníctvom optického zosilnenia. Jeho princíp bol prvýkrát popísaný Albertom Einsteinom v roku 1907 a prvý funkčný laser bol zostrojený v roku 1960 Theodorom H. Mainmanom v Hughes Research Laboratories v Kalifornii. Tento laser mal vlnovú dĺžku 649,3 nm a jeho základným prvkom boli chromové ióny a rubínový kryštál [12].

Optický oscilátor, ktorým je laser, sa skladá zo zosilňovača s mechanizmom saturácie zosilnenia a systému spätnej väzby, ako je znázornené na obrázku 2.3. Jeho princípom je, že výstupný signál sa vracia sfázovaný znovu do vstupu, ktorý sa následne zosilní. Proces zosilnenia sa opakuje, pokiaľ sa nevytvorí dostatočne veľký výstupný signál [27].



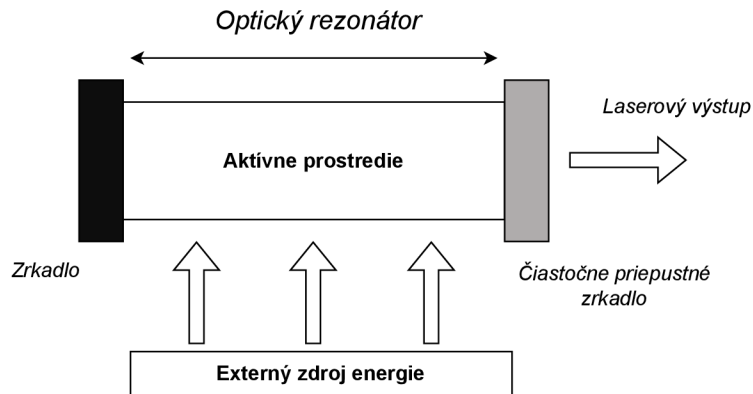
Obr. 2.3: Zostavenie optického zosilňovača.

Lasery sa využívajú v mnohých vedeckých a technických aplikáciách. Existuje mnoho typov laserov ako plynových laserov, vláknových laserov, pevnolátkových laserov, farbivových laserov, diódových laserov a excimerových laserov. Všetky tieto typy laserov zdieľajú základnú sadu komponentov [29].

### 2.2.1 Laserové komponenty

Laser sa skladá z nasledujúcich komponentov (popis podľa obrázka 2.4):

1. **Aktívne prostredie** - Aktívne prostredie predstavuje zosilovač v optickom oscilátore, ktorým je laser. Toto prostredie sa skladá zo zoskupenia atómov, molekúl, iónov alebo iných častíc, ktoré sú prvkami polovodičov, organických farbív, rôznych plynov (ako hélium a neón) alebo kryštálov. Názvy laserov sa často odvíjajú od typu prvkov aktívneho prostredia, ktoré obsahujú. V aktívnom prostredí sa častice excitujú do vyšších energetických stavov pomocou externého zdroja energie. Svetlo potom vzniká emisiou týchto častíc do nižších vrstiev.
2. **Externý zdroj energie** - Dodáva externú energiu na excitovanie atómov a iných častíc do vyšších energetických vrstiev.
3. **Optický rezonátor** - Optický rezonátor sa skladá z dvoch zrkadiel umiestnených oproti sebe. Aby sa dosiahla spätná väzba, aktívne prostredie musí byť umiestnené medzi týmito zrkadlami. Tento proces umožňuje vznik koherentného laserového svetla prostredníctvom viacerých odrazov fotónov. Vyvedenie laserového lúča je zabezpečené tým, že jedno zrkadlo je polopriepustné. Týmto spôsobom sú filtrované fotóny, ktoré prešli pozdĺž osi a ostatné fotóny sú odrazené späť do aktívneho prostredia [27].



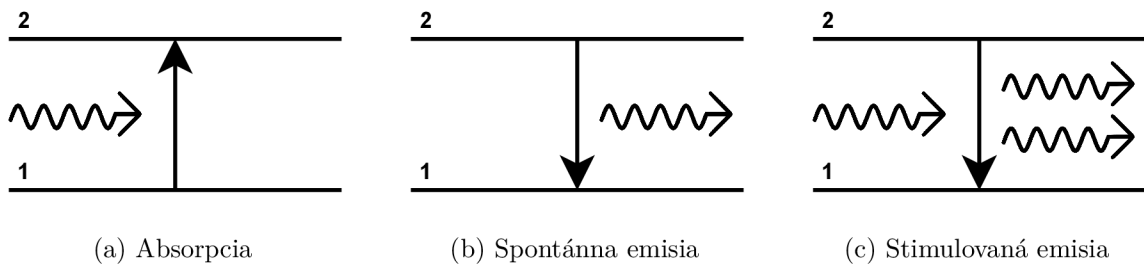
Obr. 2.4: Popis zostavy lasera.

### 2.2.2 Základné procesy v lasery

Základné procesy laserov zahŕňajú absorpciu, spontánnu emisiu a stimulovanú emisiu. V atómoch sú elektróny usporiadané na rôznych energetických úrovniach, nazývaných orbitály. Čím je elektrón na vyššom orbitále, tým je vzdialenejší od jadra a má vyššiu energiu.

Absorpcia (viď. 2.5a) nastáva vtedy, keď elektrón absorbuje energiu a presunie sa na vyššiu energetickú úroveň. Vtedy sa atóm dostáva do excitovaného stavu. Táto úroveň však nie je stabilná a tak sa elektrón po určitom čase vracia naspäť do nižšej energetickej úrovne a uvoľní absorbovanú energiu vo forme fotónu s rovnakou energiou ako rozdiel energií medzi týmito stavmi. Tento proces sa nazýva spontánnu emisiu (viď. 2.5b).

Pri stimulovanej emisii (viď. 2.5c) fotón interaguje s excitovaným atómom, čím sa elektrón excitovaného atómu vráti na nižšiu energetickú úroveň. Zároveň tak uvoľní svoju energiu vo forme fotónu. Tým vzniknú dva fotóny s rovnakou energiou, fázou a smerom ako pôvodný fotón [12].



Obr. 2.5: Znázornenie základných procesov k vzniku laserového lúča. Úroveň (1) značí nižší energetický stupeň atómu a úroveň (2) vyšší energetický stupeň atómu.

Laser je zariadenie, ktoré generuje laserový lúč pomocou riadeného uvoľňovania fotónov. K tomu je potrebné dodávať dostatočné množstvo energie externým zdrojom. Atómy v aktívnom prostredí absorbujú túto energiu a prechádzajú do excitovaného stavu. Je potrebné zabezpečiť dostatočné množstvo atómov v excitovanom stave, aby sa zvýšila populačná inverzia - pomer počtu atómov v excitovanom stave oproti počtu atómov v základnom stave.

Pomocou spontánnej emisie niektoré excitované atómy opäť prechádzajú do základného stavu a uvoľnia energiu vo forme fotónu. Tento fotón môže interagovať s ďalšími excito-



vanými atómami pomocou procesu stimulovanej emisie. Vytvára sa tak reťazová reakcia, kde sa uvoľňujú ďalšie fotóny, ktoré môžu interagovať s ďalšími excitovanými atómami. Týmto spôsobom vzniká svetelný lúč, ktorý narastá exponenciálne. Aby sa fotóny udržali v aktívnom prostredí lasera, sú odrážané pomocou zrkadiel.

Laserové svetlo je monochromatické vďaka pevnej vzdialenosti medzi energetickými úrovňami v atómoch a riadeným uvoľňovaním fotónov. Elektrón prechádza z nižšej úrovne na vyššiu úroveň po dodaní presného množstva energie danému rozdielu energetických úrovní. Pri návrate elektrónu do základného stavu sa uvoľní presné množstvo energie vo forme fotónu svetla určitej vlnovej dĺžky. Elektróny tak produkujú kaskádu fotónov - identických v frekvencii a vlnovej dĺžke [12].

### 2.2.3 Charakteristika laserov

Medzi najvýznamnejšie charakteristiky lasera patria:

#### Smerovosť laseru

Laserový lúč sa na rozdiel od bežných zdrojov svetla (elektrická žiarovka, slnko) pohybuje len jedným svetlom a vytvára malé uhlové rozpätie. Umožní tak sústrediť laserový lúč na jedno malé miesto. Vďaka tejto vlastnosti je užitočný napríklad pri diaľkovom prieskume alebo geodézii [24]. Vzdialenosť  $z$  od laserového zdroja po koniec rovnobežných lúčov je daná (2.1).

$$z = \frac{d^2}{\lambda} \quad (2.1)$$

kde  $d$  je priemer laserového lúča a  $\lambda$  je vlnová dĺžka svetla [12].

#### Intenzita

Intenzita svetelného toku  $I$  je definovaná ako množstvo energie za jednotku času na jednotku plochy (2.2).

$$I = \frac{E}{t} A = \frac{P}{A} [ W/m^2 ] \quad (2.2)$$

kde  $E$  značí energiu,  $t$  čas,  $P$  výkon a  $A$  obsah plochy.

V laseri sa svetlo šíri v úzkom lúči a svetelná energia je sústredená v malej oblasti. Laserové svetlo má teda väčšiu intenzitu v porovnaní s bežným svetlom. Veľkosť intenzity, napríklad plynového lasera, je približne  $10^{12} W/m^2$ . Laser, vďaka tejto vysokej intenzite, je široko používaný na rezanie a zváranie kovov a zliatín [12].

#### Monochromatickosť

Svetlo vyžarované zdrojom sa šíri v určitom frekvenčom rozsahu  $\Delta v$  (2.3) okolo strednej frekvencie  $\nu_0$ .

$$\Delta v = \nu_1 - \nu_2 \quad (2.3)$$

Pomer  $\Delta v/\nu_0$  meria stupeň monochromatickosti svetla  $\epsilon$  (2.4).

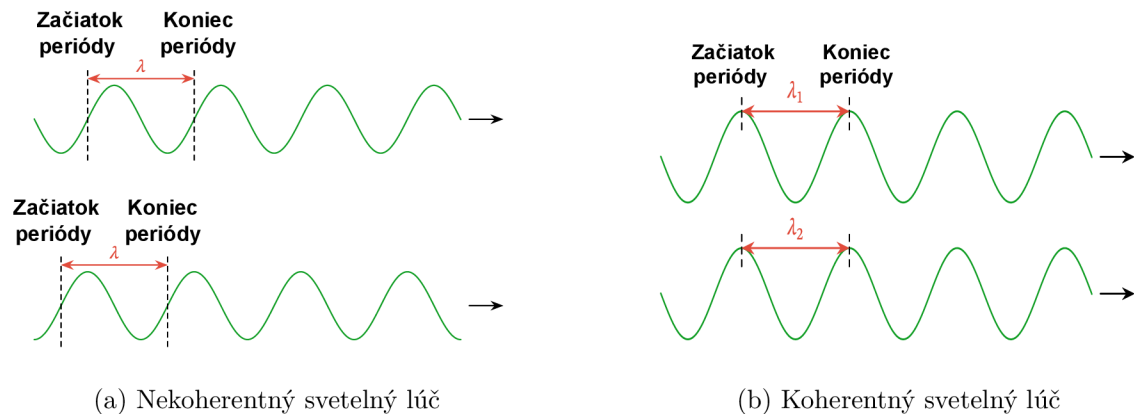
$$\epsilon = \frac{\Delta v}{v_0} \quad (2.4)$$

kde  $\Delta v$  je šírka pásma a  $v_0$  stredná frekvencia.

Svetelný lúč nadobúda monochromatický charakter pri veľmi nízkom rozsahu frekvencie  $\Delta v$ , ktorého typickým príkladom sú lasery. Fotóny, vyžarované z bežných zdrojov, majú naopak rôznu energiu, frekvenciu a široký rozsah vlnovej dĺžky s približnou šírkou pásma  $10^{10}$  Hz. Šírka pásma kvalitného lasera je približne 500 Hz [12].

## Koherentnosť

Koherentnosť je presnosť alebo súhlas vlnového tvaru svetelného signálu, kde fotóny v emitovanom svetle kmitajú v súlade (vo fáze) a súhlasia v priestore aj v čase (viď. 2.6b). Laserový lúč sa touto vlastnosťou odlišuje od ostatných zdrojov svetla, ktoré vyžarujú svetelné vlny šíriace sa všetkými smermi a v rozdielnych fázach (viď. 2.6a). Vďaka koherentnosti môže laserový lúč byť silne zúžený, sústredený a presne usmernený [24].



Obr. 2.6: Zobrazenie nekoherentného a koherentného svetelného lúča. *Prevzaté a upravené z [18].*

## 2.3 Typy laserov a ich využitie

Lasery sa dajú kategorizovať podľa použitého typu aktívneho prostredia, ktoré je potrebné na vytvorenie svetelného lúča. Medzi tieto druhy laserov patria:

### Pevnolátkové lasery

Ako aktívne prostredie v pevnolátkových laseroch sa používajú pevné látky, rubíny alebo kryštály, zmiešané so vzácnymi prvkami. Medzi typické prvky patria neodým, chróm, erbium alebo ytterbium. Medzi najznámejšie pevnolátkové lasery patria rubínové lasery, keďže ide o prvý typ lasera, ktorý bol skonštruovaný. Pevnolátkové lasery sa využívajú napríklad, v technológiach na mapovanie prostredia alebo v medicíne pri oftalmologických a onkologických procedúrach [24].

## **Plynové lasery**

V plynových laseroch sa ako aktívne prostredie používa plyn, ktorým prechádza elektrický prúd na generovanie svetla. Tieto lasery patria medzi najrozšírenejšie typy a využívajú sa v širokej škále aplikácií, vrátane holografie, spektroskopie, skenovania čiarových kódov, merania znečistenia ovzdušia, spracovania materiálov a laserovej chirurgie. Najčastejšie sa však používajú na laserové značenie, rezanie a zváranie. Medzi príklady plynových laserov patria lasery na báze oxidu uhličitého (CO<sub>2</sub>), hélium-neónové lasery, argónové lasery, kryptónové lasery a excimerové lasery [12].

## **Kvapalinové (farbivové) lasery**

Ako aktívne prostredie v kvapalinových laseroch sa používa roztok organického farbiva vo vode alebo alkohole. Tieto lasery sú významné pre svoju schopnosť generovať široké spektrum vlnových dĺžok. Príkladom môže byť Xanthenová farba, ktorá umožňuje vytváranie lúčov v rozmedzí 500 až 700 nm viditeľného spektra alebo scintilačná farba, ktorá generuje lúče v ultrafialovej oblasti spektra. Kvapalinové lasery sa často používajú v medicínskom výskume a pri separácii izotopov [24].

## **Polovodičové lasery**

Polovodičový laser, tiež nazývaný polovodičový injekčný laser či laserová dióda, má štruktúru podobnú svetelnej LED dióde. Oba typy majú zdroj energie v podobe elektrického prúdu, ktorý je injektovaný do P-N prechodu. Avšak pri LED dióde je energia generovaná pomocou spontánnej emisie, zatiaľ čo v polovodičových laseroch sa využíva stimulovaná emisia. Aktívnym prostredím v polovodičových laseroch je okolie PN-prechodu a kryštál polovodiča plní súčasne funkciu optického rezonátora, teda zrkadiel. Ich spektrálna šírka je typicky väčšia než u ostatných laserov.

Medzi hlavné výhody polovodičových laserov patria ich malé rozmery, vysoká účinnosť, integrovateľnosť s elektronickými súčastami, ľahké čerpanie a modulácia pomocou injekčného elektrického prúdu. Využívajú sa najmä v čítačkách čiarových kódov, laserových ukazovátkach, laserových tlačiarňach, laserových skeneroch a v niekoľkých ďalších aplikáciách. [27].

## **Plazmové röntgenové lasery**

Vytvoriť funkčný röntgenový laser sa podarilo až v posledných rokoch. Dôvodom boli vysoké nároky na zdroje a technológiu, vďaka krátkej vlnovej dĺžke a vysokej energii röntgenových vln. Okrem toho bolo potrebné použiť účinné zrkadlá, ktoré dokážu odrážať röntgenové lúče a udržať ich v aktívnom prostredí. Röntgenové lasery sa využívajú najmä v mikrolitografii pri výrobe polovodičových čipov a v holografii jednotlivých bunkových štruktúr v biologických systémoch. [27].

## **2.4 Bezpečnostné riziká laserov**

Lasery sa klasifikujú z bezpečnostných dôvodov do tried, na základe ich bezpečnostných rizík. Väčšina laserových produktov preto musí byť označená štítkom s uvedením ich triedy. Potenciál lasera spôsobiť poškodenie sa meria hlavne jeho výkonom v miliwattoch vo vzťahu k vlnovej dĺžke a času vystavenia.

Pre spotrebitelské lasery s viditeľným lúčom existujú štyri hlavné triedy a ich podtriedy.

### **Trieda 1**

Lasery 1. triedy nepredstavujú riziko ohrozenia, pokiaľ laser nie je rozobraný. Typickým príkladom sú prehrávače CD-ROM.

- **Trieda 1M**

Podskupina triedy 1 s rovnakými kritériami, pričom lúč je nebezpečný pri sledovaní so zväčšením.

### **Trieda 2**

Laser 2. triedy je bezpečný laser, pokiaľ mu nie je oko vystavené dlhšie ako 0,25 sekundy (odhadovaný čas na žmurknutie alebo odvrátenie pohľadu) a nepresahujúci priemerný vyžarovací výkon 1 mW. Z tohto dôvodu musí byť označený štítkom s varovaním, napr. *Ne-  
pozerajte sa do lúča* (angl. Do not stare into beam). Príkladom môže byť skener tovaru v supermarkete.

- **Trieda 2M**

Podskupina triedy 2 s rovnakými kritériami, pričom lúč je nebezpečný pri sledovaní so zväčšením.

### **Trieda 3**

Tretia trieda sa rozdeľuje na dve podskupiny:

- **Trieda 3R**

Tento typ lasera predstavuje nebezpečenstvo len vtedy, ak sa zaostrí do oka. Väčšina laserových ukazovadiel sú 3R lasery a ich výkonnosť nepresahuje 5 mW.

- **Trieda 3B**

Predstavuje nebezpečenstvo pri pohľade na priamy alebo odrazený lúč. Ich výkonnosť je v rozhraní 25mW - 250mW. Používajú sa na čítanie CD a DVD.

### **Trieda 4**

Lasery 4. triedy patria medzi najvýkonnejšie lasery s najväčším potenciálom spôsobiť poškodenie aj pri nepriamom pozorovaní. Majú schopnosť popáliť pokožku a predstavujú riziko požiaru. Ich výkonnosť sa klasifikuje od 500mW [4].

## Kapitola 3

# Predspracovanie obrazu

### 3.1 Vyhladenie obrazu

Vyhladenie, alebo tiež rozmazanie obrazu, je jednoduchý a často využívaný proces predspracovania obrazu. Jeho cieľom je odstrániť alebo redukovať šum a iné nežiadúce prvky v obraze [6].

Na vyhladenie obrazu sa používajú rôzne druhy filtrov, ktoré menia hodnoty jasů pixelov vstupného obrazu na iné hodnoty vo výstupnom obraze. Filtre sa aplikujú buď na celý obraz alebo len na určitú časť obrazu. Pri lokálnom filtrovaní sa používajú priestorové filtre, ktoré sú charakterizované pomocou konvolučných masiek alebo filtračných jadier. Tieto filtre prechádzajú obraz systematicky a každý bod sa upraví na základe malého okolia okolo neho. V závislosti od postupu sa druhy filtrovania rozlišujú na lineárne a nelineárne [17].

#### 3.1.1 Lineárne filtrovanie

Nová hodnota reprezentatívneho bodu sa pri lineárnom filtrovaní počíta ako kombinácia hodnôt pixelov z jeho okolia. Rozdiely medzi jednotlivými lineárnymi filtrami sa prejavujú vo veľkosti váh, ktoré určuje konvolučná maska  $H$ . Medzi bežné lineárne filtre patria, napríklad, priemerovanie a Gaussov filter [17].

#### Filtrácia priemerovaním

Priemerovanie je základná metóda vyhladzovania obrazu, ktorá využíva priemernú hodnotu pixelov v jeho určitom okolí. Tento postup spočíva v nahradení hodnoty každého pixelu priemernou hodnotou pixelov v jeho okolí, čo vedie k redukcii šumu a rozmazaniu obrazu.

Majme vstupnú masku okolia určitého pixela  $f(x, y)$ , ktorá obsahuje  $m$  počet pixelov so súradnicami  $(i, j)$ . Potom sa môže výpočet upraveného okolia  $g(x, y)$  vyjadriť pomocou rovnice (3.1).

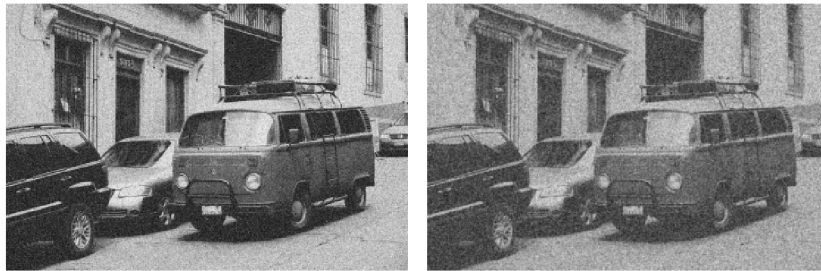
$$g(x, y) = \frac{1}{m} \sum_{(i,j) \in s} f(i, j) \quad (3.1)$$

Je dôležité zohľadniť veľkosť okolia, pretože zväčšenie okolia môže spôsobiť intenzívnejšie rozmazanie v okrajoch a detailoch obrázka. V takom prípade sa používa modifikovaná metóda, nazývaná vážené priemerovanie, pri ktorej sa stredovému bodu alebo jeho susedom priradí vyššia váha. Táto metóda je zlepšenou verziou priemerovania a umožňuje lepšie zachovanie detailov a hrán v obraze [19].

Na popis týchto metód sa používajú konvolučné masky, kde väčšie masky sa vytvárajú analogicky. Nasledujúce rovnice popisujú konvolučné masky pre priemerovanie (3.2):

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.2)$$

Nová hodnota pixelu sa vypočíta ako súčin hodnôt v konvolučnej maske a odpovedajúcich hodnôt pixelov v obraze. Tieto súčiny sa následne sčítajú a výsledok sa vydelením celkovou váhou masky, ktorá predstavuje súčet všetkých jej hodnôt [17]. Vplyv filtrovania priemerovaním je vidieť na obrázku 3.1.



Obr. 3.1: Aplikácia filtra priemerovania (obr. vpravo) na pôvodný obraz (obr. vľavo).

### Gaussovo rozmazanie

Gaussovo rozmazanie (Gaussov filter) je jednou z najužitočnejších filtračných metód v oblasti počítačového videnia. Tento proces sa vykonáva konvolúciou každého pixelu vstupného obrazu s Gaussovým jadrom [6].

Gaussové jadro je dvojrozmerná matica, ktorej hodnoty sú definované vzorcom (3.3).

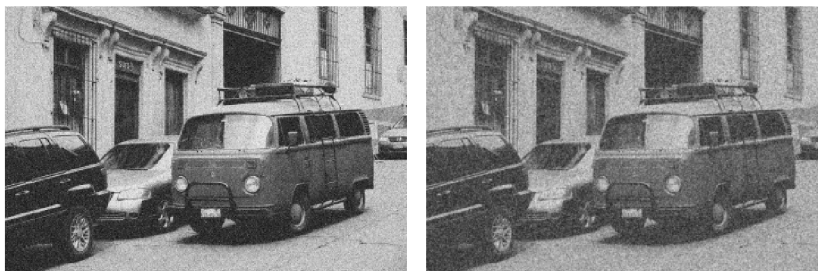
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.3)$$

kde  $\sigma$  je štandardná odchýlka, ktorá určuje šírku filtra a vplýva na výsledný vyhladený obraz. Čím väčšia je hodnota  $\sigma$ , tým širšie frekvenčné pásmo má Gaussov filter. Jadro Gaussovho rozloženia potom môže nadobudnúť hodnoty vyjadrené maticou (3.4):

$$\frac{1}{57} \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 3 & 5 & 3 & 1 \\ 2 & 5 & 9 & 5 & 2 \\ 1 & 3 & 5 & 3 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix} \quad (3.4)$$

Gaussovo rozmazanie sa často používa ako predspracovanie v oblasti počítačového videnia, pretože dokáže zlepšiť kvalitu obrazov s rôznymi rozmermi. Táto metóda kombinuje obraz s normálnym rozdelením a využíva sa pre nízkofrekvenčné časti obrázkov, aby sa potlačil vysokofrekvenčný šum. Avšak, pri niektorých situáciách môže dochádzať k rozmazaniu hrán alebo detailov v obraze, preto je dôležité správne zvoliť veľkosť parametrov Gaussovho filtra [19]. Jeho vplyv ukazuje obrázok 3.2.





Obr. 3.2: Gaussov filter (obr. vpravo) aplikovaný na pôvodný obraz (obr. vľavo).

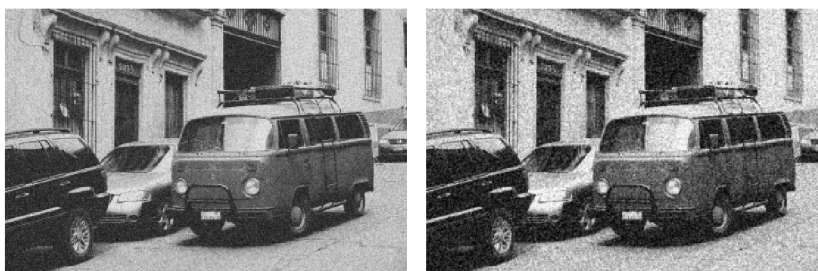
### 3.1.2 Nelineárne filtre vyhladzovania

Nelineárne filtre sa zameriavajú na zachovanie hrán. Nepoužívajú pri tom výpočet intenzity pixelov, ale vyberajú vhodnú hodnotu z okolia, ktorú aplikujú na upravovaný bod. Ich cieľom je eliminovať odchylené body a nahradiť ich nejakou typickou hodnotou. Medzi najčastejšie používané nelineárne filtre patria mediánová filtrácia a bilaterálny filter [17].

#### Filtrácia mediánom

Filtrácia mediánom je typ filtra pre spracovanie obrazu, ktorý nahrádza každý pixel mediánovou hodnotou jeho okolia. Tento typ filtra je účinný pri odstraňovaní šumu z obrázkov a zároveň obnovuje hrany a ostatné detaily v obraze. Taktiež je výhodný v tom, že do obrazu nepridáva nové hodnoty intenzity.

Pri výpočte mediánu sa z hodnôt susediacich pixelov vytvorí usporiadaná postupnosť a stredný prvok tejto postupnosti sa vyberie ako mediánová hodnota. Ak je počet prvkov v postupnosti párny, stredný prvok nie je jednoznačne určený. Preto sa obvykle používajú usporiadané postupnosti s nepárnym počtom prvkov, aby sa zabezpečilo jednoznačné určenie mediánu [19]. Vplyv mediánovej filtrácie je ukázaný na obrázku 3.3.



Obr. 3.3: Aplikovanie mediánového filtrovania (obr. vpravo) na pôvodný obraz (obr. vľavo).

#### Bilaterálne filtrovanie

Bilaterálny filter bol navrhnutý Tomášom Manduchim v roku 1998. Od ostatných filtrov sa odlišuje tým, že jeho základným princípom nie je len redukcia šumu, ale aj zohľadnenie hodnôt pixelov s cieľom zachovať hrany obrazu [6]. Postup spočíva v kombinácii dvoch filtrov - priestorového a rozsahového.

Princíp priestorového filtra je v tom, že každá nová hodnota pixelu  $I(u, v)$  sa vypočíta ako vážený priemer pôvodných pixelov  $I$  v jeho okolí. Váhy jednotlivých pixelov sú

určené filtračným jadrom  $H_p$ , pričom váha každého pixelu závisí od jeho polohy vzhľadom na stredový pixel  $(u, v)$  (3.5).

$$I(u, v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} I(u+m, v+n) \cdot H_p(m, n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot H_p(i-u, j-v) \quad (3.5)$$

Rozsahový filter zvažuje len rozdiel hodnôt pixelov v okolí, pričom poloha spracovávaného pixela sa neberie do úvahy. Závisí iba na rozdieli medzi hodnotou v aktuálnom stredovom pixeli  $I(u, v)$  a jeho vlastnou hodnotou  $I(i, j)$ . Vplyv každého pixelu je určený jadrom  $H_r$  a tým pádom nenabudne rozmazanie ani zvýraznenie obrazu (3.6).

$$I(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot H_r(I(i, j) - I(u, v)) \quad (3.6)$$

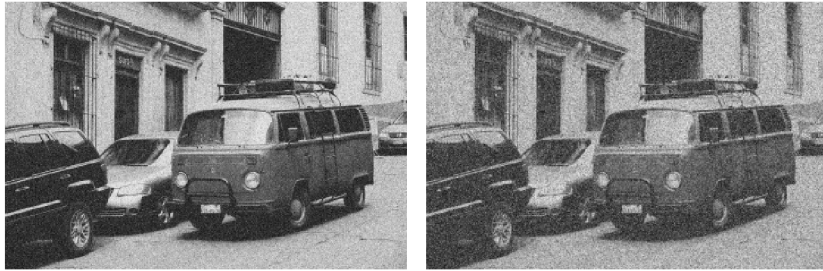
Bilaterálny filter spája tieto dva filtre do formy (3.7).

$$I(u, v) = \frac{1}{W_{u,v}} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot H_p(i-u, j-v) \cdot H_r(I(i, j) - I(u, v)) \quad (3.7)$$

kde  $H_p$  a  $H_r$  sú filtračné jadrá priestoru a rozsahu.  $W_{u,v}$  je zloženie jednotlivých váh  $w_{i,j}$  (3.8), ktoré sú použité na normalizáciu filtračných jadier.

$$W_{u,v} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} w_{i,j} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} H_p(i-u, j-v) \cdot H_r(I(i, j) - I(u, v)) \quad (3.8)$$

Rozsah filtrovaného okolia je tak obmedzené priestorovým jadrom  $H_p$ . V okolí stredového pixela  $(u, v)$  je každému pixelu  $(i, j)$  priradená váha  $w_{i,j}$ , ktorá závisí od jeho priestorej polohy vzhľadom na  $(u, v)$  a od podobnosti hodnoty so stredovým pixelom  $(u, v)$ .



Obr. 3.4: Aplikovanie bilaterálneho filtrovania (obr. vpravo) na pôvodný obraz (obr. vľavo).

Vďaka kombinácii týchto dvoch filtrov v plochých oblastiach, kde sú hodnoty okolitých pixelov podobné ako hodnota stredového pixelu, pôsobí bilaterálny filter ako bežný vyhladzovací filter, ktorý využíva len priestorové jadro  $H_p$ . Avšak v oblastiach blízko hrán sa do procesu vyhladzovania zahrnú iba pixely s podobnými hodnotami ako stredový pixel, čím sa predchádza rozmazaniu okrajov. Medzi bežne používané filtračné jadrá pri bilaterálnom filtrovaní patrí Gaussovo jadro [7]. Vplyv bilaterálneho filtra demonštruje obrázok 3.4.



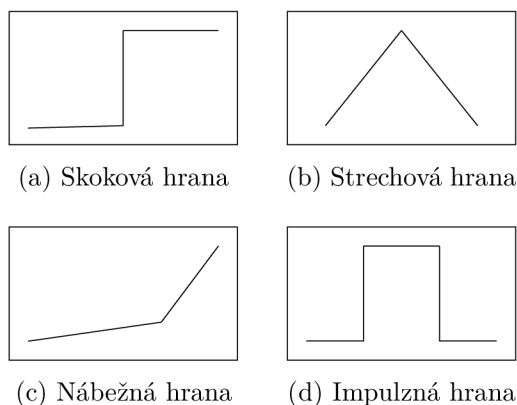
## Kapitola 4

# Detekcia čiar a zdroja v obraze

### 4.1 Detekcia hrán

Detekcia hrán patrí medzi jednu z najviac používaných operácií v počítačovom videní, ktorá umožňuje zachytiť významné vlastnosti objektov v obraze. Medzi tieto vlastnosti patria hlavne rozdiely vo fotometrických, geometrických a fyzikálnych javoch v predmetoch, čo je spôsobené v zmenách v úrovni šedej farby (jas, odraz) [22].

Tieto oblasti s výraznou zmenou intenzity a kontrastu sa nazývajú hrany, vďaka ktorým môžeme získať obrys hľadaného objektu. Medzi najbežnejšie typy používaných hrán patria skokové (step), impulzné (line), nábežné (ramp) a strechové (roof) hrany (obr.: 4.1).



Obr. 4.1: Príklady typov hrán.

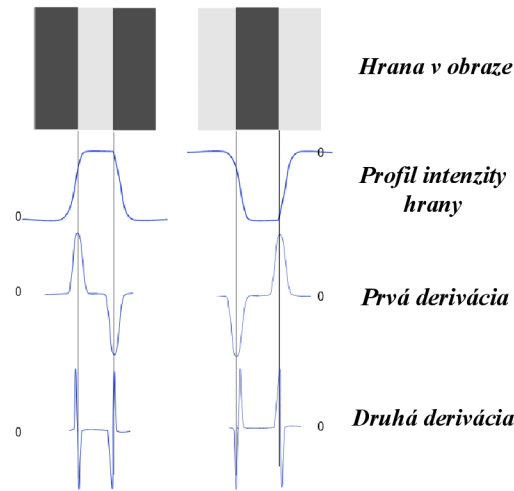
Skokové hrany značia náhlu zmenu hodnôt z jednej intenzity jasu na opačnú. Impulzné hrany taktiež indikujú prudkú zmenu hodnôt, ale po istej vzdialenosti sa opäť vrátia na pôvodnú hodnotu. Tieto spomenuté hrany patria medzi ideálne, v reálnom svete sa však viac stretávame s nábežnými a strechovými hranami. Nábežné hrany majú vlastnosti podobné so skokovými, ale ich zmeny hodnôt nie sú okamžité, ale postupné. Podobný spôsob správania majú aj strechové voči impulzívnym hranám [21].

Účelom detekcie hrán je ich lokalizácia a identifikácia javov, ktoré ich vytvárajú. Vďaka tomuto procesu sa výrazne zníži množstvo dát, vyfiltrujú sa nepotrebné informácie a zároveň sa zachovávajú iba dôležité štrukturálne vlastnosti obrazu. Pri detekcii hrán postupujeme po štyroch základných krokoch: (1) vyhladenie, (2) zlepšenie kvality obrazu, (3) detekcia a (4) hranová lokalizácia.

Cieľom pri vyhladení obrázka je zredukovať čo najviac šumu bez odstránenia hrán a tak zvýšiť produktivitu hranového detektora. Vylepšenie obrazu zvýši jeho kvalitu a tým sa vytvorí obraz vhodnejší na ďalšiu analýzu. Detekcia hrán extrahuje všetky potenciálne hranové body a hranová lokalizácia zahŕňa iba selekciu tých bodov, ktoré sú reálnymi prvkami hrán.

Hranové body sa v hranových detektoroch zisťujú deriváciami, pomocou ktorých vieme identifikovať zmenu v obraze. Algoritmy detekcie podľa toho klasifikujeme na metódu prvej derivácie (Gradient), metódu druhej derivácie (Laplacian) a na optimálne hranové detektory [21].

Vplyv prvej a druhej derivácie na krajné body hrán popisuje obrázok 4.2, pomocou čoho sa detekujú výsledné hrany na základe zmeny intenzity hodnôt obrazu.



Obr. 4.2: Vplyv prvej a druhej derivácie na hranu objektu v obraze. *Prevzaté a upravené z [2].*

#### 4.1.1 Gradient obrazovej funkcie (metódy založené na prvej derivácii jasovej funkcie)

Gradientné metódy detekujú hrany na základe maximálnej a minimálnej hodnoty v prvej derivácii obrazu. Gradient je dvojdimensionálny ekvivalent prvej derivácie definovaný ako vektor (4.1).

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4.1)$$

Gradient je spojený s dvoma dôležitými vlastnosťami: (1) vektor  $G[f(x, y)]$  ukazuje smer maximálnej rýchlosti nárastu funkcie  $f(x, y)$  a (2) veľkosť gradientu (4.2) predstavuje túto rýchlosť.

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} \quad (4.2)$$

Veľkosť gradientu (4.2) je teda totožná s maximálnou rýchlosťou nárastu  $f(x, y)$  na jednotkovú vzdialenosť v smere gradientu. Gradient môže byť tiež približne vyjadrený absolútnymi hodnotami (4.3).

$$G[f(x, y)] \approx |G_x| + |G_y| \quad (4.3)$$

Z vektorovej analýzy je smer gradientu definovaný ako (4.4)

$$a(x, y) = \tan^{-1} \left( \frac{G_x}{G_y} \right) \quad (4.4)$$

kde uhol  $a$  je určený vzhľadom k  $x$ -ovej osi.

Gradient obrazovej funkcie je citlivý na šum a môže produkovať hrubé hrany. Niektoré z príkladov metód využívajúcich prvú deriváciu sú Robertsonov, Prewittov a Sobelov operátor na detekciu hrán [21].

### Robertsonov operátor

Robertsonov operátor vykonáva jednoduchý a rýchly výpočet vďaka 2-D gradientnému meraniu v snímke. Poskytuje jednoduchú aproximáciu gradientnej veľkosti danú rovnicou (4.5).

$$G[f[i, j]] = |f[i, j] - f[i + 1, j + 1]| + |f[i + 1, j] - f[i, j + 1]| \quad (4.5)$$

Pri použití konvolučnej masky sa zmení na tvar (4.6)

$$G[f(x, y)] = |G_x| + |G_y| \quad (4.6)$$

pričom maska Robertsonovho operátora je daná konvolučnými maskami (4.7)

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (4.7)$$

kde  $G$  značí gradient,  $x$  a  $y$  sú horizontálne a vertikálne masky os [21].

### Sobelov operátor

Sobelov operátor patrí medzi jeden z najviac používaných hranových detektorov. Jeho funkcia je na základe nachádzania približnej absolútnej veľkosti gradientu v každom bode šedotónového obrazu.

Celkovo ide o veľkosť gradientu  $\|\nabla f\|$ , ktorá sa vypočíta podľa rovnice (4.8).

$$\|\nabla f\| = \sqrt{S_x^2 + S_y^2} \quad (4.8)$$

Hodnoty  $S_x$  a  $S_y$  sa získavajú pomocou konvolučných matíc (4.9), kde  $S_x$  určuje gradient v smere  $x$  (stĺpce) a  $S_y$  určuje gradient v smere  $y$  (riadky) [21].

$$S_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad S_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4.9)$$

## Prewittov operátor

Prewittov operátor je podobný Sobelovmu (4.1.1). Využíva sa na detekciu zvislých a vodorovných hrán a adekvátnym spôsobom odhaduje ich veľkosť a orientáciu. Je predovšetkým vhodný na kontrastné obrázky s nízkym šumom. Na nájdenie najväčšieho gradientu využíva konvolučné masky veľkosti 3x3 (4.10) [21].

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (4.10)$$

### 4.1.2 Laplacián obrazovej funkcie (Metódy založené na druhej derivácii)

Pri metódach založených na druhých deriváciách, hranové body sú detekované hľadaním nulového priechodu s druhou deriváciou funkcie. Využívajú Laplaceov operátor  $\nabla^2$ , ktorého výpočet (4.11) je podobný výpočtu veľkosti gradientu.

$$\nabla \times \nabla = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \times \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4.11)$$

Medzi jeho vlastnosti patrí vysoká citlivosť na šum a nevedomosť o smere hrany, kvôli použitiu druhej derivácie. Typickým príkladom operátora hranových detektorov, ktorý využíva druhú deriváciu, je Laplacián Gaussianu (angl. *Laplacian of Gaussian* - LoG) [21].

### Laplacián Gaussianu (angl. *Laplacian of Gaussian* - LoG)

Z dôvodu vyššej citlivosti Laplaceovho operátora sa šum najskôr v snímke zredukuje pomocou low-pass filtra. V prípade LoG sa ako low-pass filter používa Gaussov filter. Potom sa aplikuje Laplaceov operátor na nájdenie hrán. Gaussova funkcia je definovaná ako (4.12), kde hodnota  $\sigma$  určuje stupeň vyhladenia snímky.

$$h(x, y) = -e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (4.12)$$

Kombinácia Laplaceovho operátora a Gaussovho filtra sa ďalej definuje ako (4.13).

$$\nabla^2 g(x, y) = \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (4.13)$$

Masky pre LoG sú definované ako (4.14) [21].

$$G_x = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.14)$$

### 4.1.3 Optimálne hranové detektory

Vstupom optimálnych hranových detektorov je šedotónový obrázok a ich výstupom je binárny obraz s hranovými bodmi. Typickým príkladom optimálneho hranového detektora je Cannyho a Shen-Castanov detektor [21].

## Cannyho detektor

Cannyho detektor je považovaný za jeden z najlepších detektorov hrán vďaka jeho schopnosti dobre zvládať šum a detekovať reálne hrany s minimálnou chybou [25]. Bol navrhnutý Johnom Cannyom v roku 1986. Zároveň špecifikoval tri kritériá, ktoré musí splniť každý detektor hrán:

1. **Miera chýb:** Hranový detektor by mal detekovať všetky hrany s minimálnym množstvom chýb.
2. **Lokalizácia:** Vzďialenosť medzi hranovými bodmi nájdenými hranovým detektorom a skutočnými by mala byť čo najmenšia.
3. **Odozva:** Hranový detektor by nemal identifikovať viaceré hranové pixely v prípade že existuje len jedna hrana [22].

Cannyho algoritmus sa skladá z nasledujúcich krokov:

1. **Vyhľadanie obrazu:** Aby sa odstránil šum, obraz sa rozmazá pomocou konvolúcie s Gaussovým filtrom.
2. **Hľadanie hrán:** Detekcia hrán v bodoch, v ktorých veľkosť gradientu nadobudla najväčšiu hodnotu pomocou konvolučných masiek vo vertikálnom aj horizontálnom smere (napr. Sobelov operátor).
3. **Potlačenie minimálnych hodnôt:** Na detekciu sa použijú iba lokálne maximá v smere gradientu, čím sa minimalizuje počet falošných hrán.
4. **Využitie dvojitého prahovania:** Potenciálne hrany sa určia prahovaním. Cannyho algoritmus pritom namiesto statickej hodnoty prahu využíva *hysteresis thresholding* (prahovanie s hysterezou), kedy sa hodnota prahu prispôbuje lokálnym hodnotám. Zároveň zavádza dve prahové úrovne, vysokú  $t_h$  a nízku  $t_l$ , pričom platí, že  $t_h > t_l$ . Hodnoty pixelov nad hodnotou  $t_h$  sú automaticky klasifikované ako súčasť hrany.
5. **Stanovenie finálnych hrán:** Identifikácia konečných hrán, potlačením bodov, ktoré sa nedotýkajú hrán s veľkou hodnotou [26].

### 4.1.4 Výhody a nevýhody jednotlivých hranových detektorov

Na základe spomenutých vlastností môžeme porovnať využitie jednotlivých detektorov.

- **Gradient obrazovej funkcie (Sobelov, Prewitt, a pod.)**

Výhodou je jej jednoduchosť v používaní a dostupnosť informácie ohľadom orientácií hrán, vďaka použitiu prvej derivácie. Nevýhodou je jej vysoká citlivosť voči šumu.

- **Laplacián obrazovej funkcie**

Laplacián obrazovej funkcie úspešne lokalizuje hrany a na testovanie využíva aj lokálne okolie hrán. Vďaka Laplaceovmu operátoru, ale nemožno získať informáciu ohľadom smeru hrany.

- **Optimálne hranové detektory (Canny, a pod.)**

Využíva sa pravdepodobnosť na nájdenie chybovej miery, lepšia detekcia aj v podmienkach s väčším šumom. Nevýhodou je časová náročnosť a zložitosť [25].

## 4.2 Houghova transformácia

Houghova transformácia je metóda používaná pri počítačovom videní, analýze a spracovaní obrazu. Jej cieľom je detekcia a segmentácia jednoduchých tvarov v obraze ako sú kružnice a priamky. Patentovaná bola v roku 1962, Paulom Houghom [23].

### Princíp Houghovej transformácie

Princíp detekcie priamok v obraze za použitia Houghovej transformácie spočíva v transformácii bodov z obrazového priestoru  $(x, y)$  na priamky v parametrickom priestore  $(k, q)$ . Priamka v obrazovom priestore  $(x, y)$  je definovaná všeobecnou rovnicou v smernicovom tvare (4.15), kde  $k$  je smernica a  $q$  je priesečník s osou  $y$ .

$$y = kx + q \quad (4.15)$$

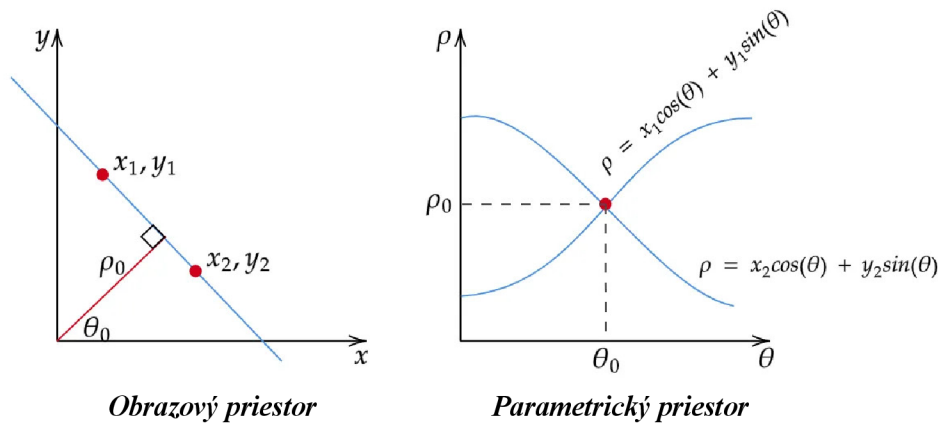
Pri transformácii aspoň dvoch bodov  $A = (x_1, y_1)$ ,  $B = (x_2, y_2)$  z tejto priamky do parametrického priestoru vzniknú priamky (4.16).

$$q = -kx_1 + y_1, q = -kx_2 + y_2 \quad (4.16)$$

Priesečník týchto dvoch priamok informuje o hľadaných konštantách  $k$  a  $q$ , pomocou ktorých sa definuje priamka spájajúca tieto dva body v obrazovom priestore [15].

Pri tomto postupe nastanú komplikácie, keď hľadaná priamka v obrazovom priestore bude orientovaná vertikálne, čo znamená, že smernica  $k$  bude nekonečná. Riešením je použitie polárnych súradníc. Priamka bude určená rovnicou (4.17), kde konštanta  $\theta$  značí uhol, zvierajúci s osou  $x$  a  $p$  vzdialenosť priamky od začiatku súradnicovej sústavy.

$$x \cos \theta + y \sin \theta = p \quad (4.17)$$



Obr. 4.3: Houghova transformácia. Prevzaté a upravené z [3].

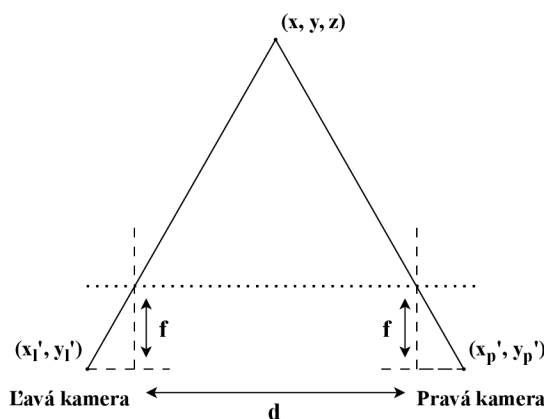
Každý bod z obrazového priestoru bude pri tomto postupe riešenia reprezentovaný v priestore parametrov  $(\theta, p)$  sínusoidou. Sínusoidy ktoré reprezentujú body ležiace na jednej priamke v obrazovom priestore, sa pretnú v jednom bode v parametrickom priestore.

Na základe tohto priesečníka sa definujú konštanty  $\theta$ ,  $p$  a určí sa hľadaná rovnica priamky (obr.: 4.3) [16].

Pri detekcii priamok sa filtrácia šumu zaistuje nastavením hodnoty prahu, ktorý reprezentuje počet pretnutí v jednom bode, ktoré musí priamka spĺňať, aby bola detekovaná. Je to spôsobené tým, že v parametrickom priestore vyšší počet priesečníkov v jednom bode signalizuje vyššiu pravdepodobnosť, že ide o priamku a nie o náhodný šum. Zvýšením prahu sa tak zaistí filtrácia len tých lokálnych maxím, ktoré zodpovedajú jeho hodnote. Pri takomto postupe sa zvýši filtrácia šumu, ale zníži citlivosť voči detailom [30].

### 4.3 Triangulácia

Jednou z metód, ktorými je možné detekovať a lokalizovať zdroj lasera v obraze, je pomocou triangulácie. Triangulácia patrí medzi techniky stereovidenia a zaoberá sa procesom nájdenia bodu v priestore na základe dvoch snímok rovnobežných kamier s predom definovanou vzdialenosťou od seba. Pokiaľ sa podarí nájsť k bodu v obraze ľavej kamery zodpovedajúci bod v pravej, potom z jednoduchých rovníc sa ľahko získa informácia aj o hĺbke bodu od pozorovateľa [15]. Tento princíp znázorňuje obrázok 4.4.



Obr. 4.4: Geometria triangulácie.

Nech vzdialenosť medzi optickými kamerami je  $d$  (disparita) a ich ohnisková vzdialenosť  $f$ . Hĺbku bodu  $z$  potom sa zmeria pomocou konštrukcií o nerovnosti trojuholníkov (4.18) [28].

$$\frac{x_l'}{f} = \frac{x + d/2}{z}, \frac{x_p'}{f} = \frac{x - d/2}{z}, \frac{y_l'}{f} = \frac{y_p'}{f} = \frac{y}{z} \quad (4.18)$$

Z týchto rovníc sa už jednoducho odôvodní hĺbka  $z$  (4.19).

$$z = \frac{df}{x_l' - x_p'} \quad (4.19)$$

Alternatívnou metódou ku triangulácii je použitie epipolárnej geometrie. Geometria rieši problém strojového videnia s dvoma kamerami, kedy ale neboli poskytnuté žiadne trojrozmerné metrické informácie. Jediné dostupné informácie sú obsiahnuté na dvoch obrázkoch tej istej scény, ale z rôznych uhlov.



## Kapitola 5

# Návrh a implementácia systému

### 5.1 Vytvorenie dátovej sady

Vytvorenie dátovej sady sa uskutočnilo v rámci troch rozsiahlych meraní v prostredí Vojenského výcvikového priestoru Brdy-Jince. Merania boli realizované v spolupráci s Ústavom leteckej dopravy Dopravnej fakulty Českého vysokého učenia technického v Prahe a s Univerzitou obrany, Katedrou zbraní a munície. Pri každom meraní bol použitý na oslňovanie nízkoenergetický laser triedy 3B s výkonom 100 mW a vlnovou dĺžkou 532 nm (zelená farba). Dôvodom výberu tejto farby lasera je jej najčastejšie využitie pri nebezpečnom ožiarovaní lietadiel. Na snímanie laserového lúča sa použili rôzne druhy kamier, aby sa porovnali ich výsledky a zabezpečilo sa dostatočné množstvo dát na následnú detekciu lúča.

#### 5.1.1 Popis merania 1

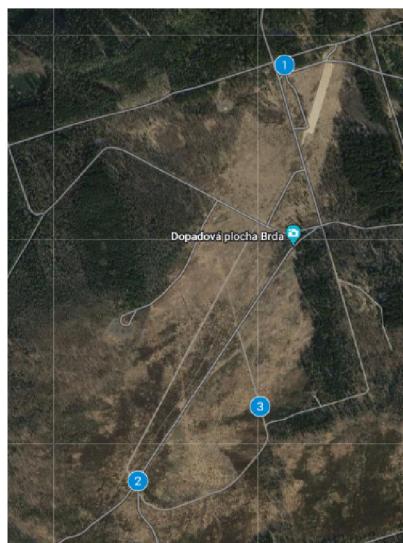
Prvé meranie sa uskutočnilo v júni 2022, ktoré malo za cieľ zoznámiť sa, porovnať rôzne kamerové systémy a vytvoriť prvotné snímky pre návrh systému na detekciu laserového lúča. Na snímanie sa použili kamerové systémy uvedené v tabuľke 5.1.

číslo	názov	typ snímača	maximálne rozlíšenie	typ kamery
1	Basler acA4024-29-um	IMX183	5472 x 3648	monochromatická
2	Prosilica GT6400C	IMX342	6480 x 4860	farebná
3	Olympus OM-D E-M10 Mark II	Live MOS	4608 x 3456	farebná
4	Basler acA4112-30uc	IMX253	4096 x 3000	farebná so zeleným filtrom

Tabuľka 5.1: Použité druhy kamier v 1. meraní.

Meranie sa uskutočnilo z dvoch pozícií - pozície č. 1 a pozície č.2, ktoré sú tiež zobrazené na obrázku 5.1. Tieto miesta sú od seba vzdialené približne 1,3 km. Z pozície č. 1 sa použil nízkoenergetický laser na oslnenie a z pozície č. 2 sa vytvorený laserový lúč zosnímal. Laser potom postupne menil svoj uhol naklonenia, ktorého lúč sa následne zaznamenával pomocou kamier.





Obr. 5.1: Zobrazenie miest vo Vojenskom výcvikovom priestore Brdy - Jince, z ktorých bolo uskutočnené meranie. *Prevzaté z [1].*

Výsledok merania ukázal, že najviditeľnejší laserový lúč bol zachytený farebnými kamerami č. 2 a č. 3. V ostatných snímkach z kamier č. 1 a č. 4 bol viditeľný iba zdroj lasera a jeho lúč bol viditeľný len v určitých uhloch. Pri prvom návrhu systému sa tiež zistilo, že najpresnejšie výsledky pri detekcii lúča boli dosiahnuté pomocou farebných kamier, pretože bolo možné jednoducho zvýrazniť lúč prahovaním a zelenou maskou.

Celkovo prvotné meranie bolo veľmi časovo náročné, pretože sa manuálne menil uhol lasera a nastavenie parametrov kamier. Napriek tomu sa podarilo porovnať snímky z rôznych kamerových systémov, čo viedlo k zostaveniu optimálnejšieho setu kamier pre ďalšie meranie.

### 5.1.2 Popis merania 2

V decembri 2022 sa uskutočnilo druhé meranie, ktoré pozostávalo z dvoch fáz. Jeho cieľom bolo získať veľký objem dát, porovnať vplyv rôznych parametrov na výsledné snímky (napr. veľkosť expozície a zisku) a vytvoriť snímky laserového lúča z rôznych uhlov s cieľom umožniť následnú lokalizáciu zdroja. Na zachytenie obrazu sa použili kamerové systémy uvedené v tabuľke 5.2.

číslo	názov	typ snímača	maximálne rozlíšenie	typ kamery
1	Basler acA4024-29-um	IMX183	5472 x 3648	monochromatická
2	Prosilica GT6400C	IMX342	6480 x 4860	farebná
3	Basler acA4112-30-uc	IMX253	4096 x 3008	farebná
4	Prosilica GT1600	ICX274	1620 x 1220	monochromatická

Tabuľka 5.2: Použité druhy kamier v 2. meraní.

Meranie sa uskutočnilo z troch pozícií, ktoré sú zobrazené na obrázku 5.1. Oslňovanie bolo v oboch fázach uskutočnené z pozície č. 1 pomocou nízkoenergetického lasera. V prvej fáze sa snímky laserového lúča snímali z pozície č. 2, v druhej fáze súčasne z pozícií č. 2

a č. 3. Vzďialenosť medzi miestom osľňovania (pozícia č. 1) a pozíciou snímania (č. 2) bola 1,70 km. Medzi pozíciou č. 3 a č. 1 bola táto vzdialenosť 1,3 km.

Pri meraní sa najskôr nastavil uhol naklonenia laseru a potom sa zosnímal laserový lúč kamerami. Následne sa uhol naklonenia laseru zmenil a lúč sa opäť zosnímal. Tento postup sa opakoval viackrát, až kým nebolo zhotovené dostatočné množstvo dát s rôznymi uhlami. Naklonenie laseru sa menilo pomocou rotačnej plošiny postupne po 5 až 10-tych stupňoch horizontálne a vertikálne, čo zabezpečilo presnosť a rýchlosť nastavenia.

Kamery snímali laserový lúč s dobou expozície v rozmedzí od 1 ms do 4 s a so ziskom v rozmedzí od 0 percent až po maximálnu možnú hodnotu zisku danej kamery. Snímanie prebehlo pomocou vopred pripravených programov, ktoré menili postupne hodnoty parametrov na jednotlivých kamerách a zabezpečili tak rýchly proces merania.

Vďaka tomuto postupu sa získal veľký objem dát, vďaka čomu sa mohli ďalej porovnať vplyvy použitia jednotlivých kamier a ich parametrov na výslednú snímku. V dôsledku toho sa zúžil počet parametrov, s ktorými sa obraz zosnímal, vďaka čomu sa proces ďalšieho merania urýchlil. Navyše sa mohlo vykonať ďalšie testovanie systému na detekciu laserového lúča v obraze.

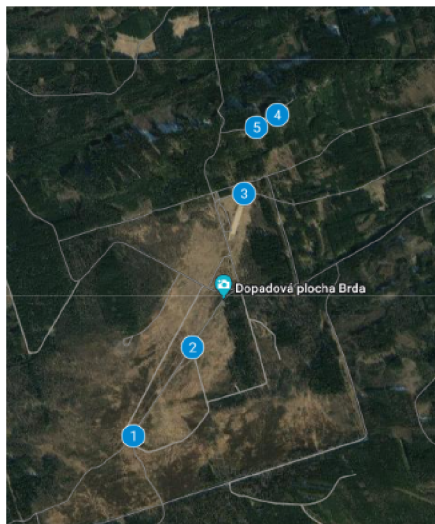
### 5.1.3 Popis merania 3

Tretie meranie sa uskutočnilo v marci 2023 s cieľom vytvoriť snímky laserových lúčov z rôznych uhlov, aby sa pomocou nich mohol lokalizovať ich zdroj. Taktiež sa zaoberalo zistením vplyvu použitia zeleného filtra v monochromatických kamerách, keďže v týchto kamerách nie je možné použiť zelenú masku. V tabuľke 5.3 sú opísané použité kamerové systémy.

číslo	názov	typ snímača	maximálne rozlíšenie	typ kamery
1	Basler acA5472-17um	IMX183	5472 x 3648	monochromatická so zeleným filtrom
2	Basler acA5472-5gm	IMX183	5472 x 3648	monochromatická so zeleným filtrom
3	Basler acA5472-5gc	IMX183	5472 x 3648	farebná
4	Allied Vision Alvium 1800 U-2050c	IMX178	1936 x 1216	farebná

Tabuľka 5.3: Použité druhy kamier v 3. meraní.

Snímanie prebiehalo v troch fázach, pričom každá fáza bola zrealizovaná z rôznych pozícií, ktoré sú zobrazené na obrázku 5.2, v rôzne dni. V prvom kroku sa osľňovalo z pozície č. 1 a snímalo sa z pozície č. 3. V druhom kroku sa taktiež osľňovalo z pozície č. 1, ale zároveň sa snímalo z pozícií č. 4 a 5. V tretej fáze sa poloha lasera zmenila na pozíciu č. 2.



Obr. 5.2: Zobrazenie miest vo Vojenskom výcvikovom priestore Brdy - Jince, z ktorých bolo uskutočnené tretie meranie. *Prevzaté z [1].*

Oproti predchádzajúcemu meraniu došlo k zmenám v expozičnej dobe snímania snímok, ktoré boli nastavené na rozmedzie 100 ms až 2 s. Táto zmena bola potrebná, pretože ostatné snímky získané pri rôznych expozičných časoch boli buď príliš tmavé alebo príliš svetlé.

Vzhľadom na potrebu ukladať veľké množstvo dát v krátkom čase, bola každá kamera vybavená zariadením na ukladanie zosnímaných dát, aby dáta mohli byť nahrávané každou kamerou zvlášť a neboli stahované na jedno centrálné miesto. Pri inom druhu ukladania, napríklad, prostredníctvom USB 3.0, je maximálna prenosová rýchlosť 5 Gb/s, čo umožňuje získať iba 17 FPS pri maximálnom rozlíšení. Prenosová rýchlosť Gigabit Ethernetu je ešte nižšia a dosahuje iba 1 Gb/s.

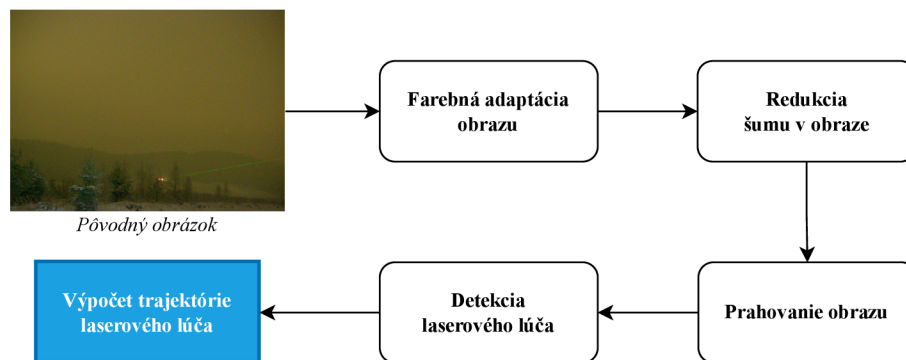
Pomocou merania sa podarilo dosiahnuť výsledky, ktoré umožnili porovnať schopnosti rôznych typov kamier pri snímaní laserového lúča. Výsledky potvrdili, že použitie zeleného filtra umožnilo zvýšenie viditeľnosti lúča v monochromatickej kamere, ale viedlo to k zníženiu množstva dopadajúceho svetla na chip kamery. Pri rovnakej dobe expozície tak mali farebné kamery viac informácií. Z tohto dôvodu, farebná kamera s použitím masky stále poskytuje najlepšie zvýraznenie laserového lúča v porovnaní s ostatnými kamerami.

## 5.2 Návrh systému

Cieľom návrhu systému bolo vytvoriť základný postup spracovania obrazu za účelom detekcie laserového lúča a zoznámiť sa s metódami, ktoré zjednodušia jeho detekciu a výpočet trajektórie. Návrh je popísaný na obr. 5.3 a pozostáva z krokov: farebná úprava obrazu, redukcia šumu v obraze, prahovanie obrazu, detekcia a výpočet trajektórie laserového lúča.

V rámci detekcie laserových lúčov sa návrh zameriava na identifikáciu laserových lúčov zelenej farby. Dôvodom pre toto obmedzenie je skutočnosť, že táto farba sa najčastejšie používa pri ožarovaní lietadiel v reálnych podmienkach a preto táto farba laserového lúča bola použitá aj pri snímaní testovacích dát.

Na návrh systému bol použitý skriptovací jazyk Python, kvôli jeho jednoduchosti a ľahkej použiteľnosti. Tento jazyk bol rozšírený o knižnicu OpenCV, ktorá poskytuje potrebné funkcie pre spracovanie obrazu.



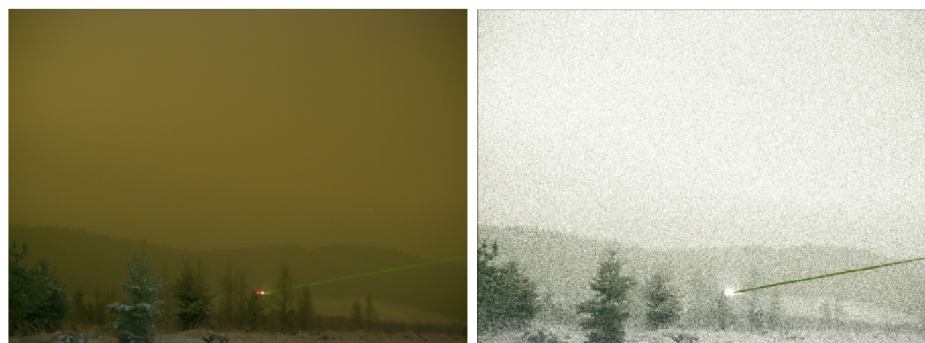
Obr. 5.3: Návrh spracovania obrazu so zachytením laserovým lúčom.

### 5.2.1 Predspracovanie obrazu

Na predspracovanie obrazu sa v návrhu systému kladie nemalý dôraz, pretože snímky boli nasnímané v otvorenom priestore vo večerných hodinách za rôznych atmosferických podmienok, aby čo najviac pripomínali reálne dáta. V dôsledku toho snímky obsahujú množstvo šumu. Predspracovanie obrazu preto zahŕňa farebnú úpravu obrazu, použitie vyhladzovacieho filtra na odstránenie šumu a prahovanie obrazu.

#### Farebná adaptácia obrazu

Farebná adaptácia sa používa na úpravu obrazu s cieľom zvýrazniť laserový lúč v snímke a tým zjednodušiť jeho detekciu. Jedným z možných spôsobov farebnej úpravy obrazu je použitie farebnej masky, ktorá využíva vlastnosť vysokého kontrastu zelenej farby laserového lúča v porovnaní s okolím v obraze. Aby sa na snímku mohla aplikovať farebná maska, musí sa najskôr transformovať do HSV modulu. Následne sa na snímku aplikuje zelená maska, čím sa potlačia z obrazu všetky pixely, ktoré nespádajú do jeho farebného rozpätia. Po tejto úprave ostanú iba body, ktoré môžu tvoriť laserový lúč. Vplyv aplikácie zelenej masky na pôvodný obraz je viditeľný na obrázku 5.4.



Obr. 5.4: Aplikácia zelenej masky (obr. vpravo) na pôvodnú snímku (obr. vľavo).  
Obrázok vpravo bol upravený invertovaním čiernych pixelov pre zlepšenie viditeľnosti.

Jedným z ďalších spôsobov, ako farebne upraviť obraz, je zmena jasnosti a kontrastu, čo môže byť výhodné pri tmavých obrazoch, ktoré sa snímali pri krátkej dobe expozície a nízkom zisku. Vďaka tomu sa v obraze dokáže zlepšiť viditeľnosť aj tých detailov, ktoré by inak boli málo viditeľné ako je, napríklad, laserový lúč.



Princíp tohto procesu spočíva v zvýšení kontrastu a postupnom pridávaní jasů, pričom sa pri každej zmene vypočíta farebnosť obrazu pomocou algoritmu na zistenie farebnej metriky. Algoritmus pre farebnú metriku bol navrhnutý Davidom Haslerom a Sabinou Süstrunkovou a závisí od uvedených výpočtov (5.1) a (5.2).

$$rg = R - G \quad (5.1)$$

$$yb = \frac{1}{2}(R + G) - B \quad (5.2)$$

Dané rovnice (5.1) a (5.2) reprezentujú farebný priestor RGB modelu, kde  $R$  je červená,  $G$  zelená a  $B$  modrá farba. V prvej rovnici značí  $rg$  rozdiel medzi červeným a zeleným kanálom a v druhej rovnici  $yb$  rozdiel medzi žltým, ktorý vznikol zložením zelenej a červenej, a modrým kanálom. Následne sa pomocou týchto hodnôt vypočíta štandardná odchýlka  $\sigma_{rgyb}$  (5.3), priemer  $\mu_{rgyb}$  (5.4) a výsledná konečná metrika farebnosti  $C$  (5.5) [14].

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \quad (5.3)$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \quad (5.4)$$

$$C = \sigma_{rgyb} + 0.3 * \mu_{rgyb} \quad (5.5)$$

Hodnota, pri ktorej bol dosiahnutý najväčší farebný kontrast, sa aplikuje na výsledný obraz, vďaka čomu sa zvýši viditeľnosť aj tých detailov, ktoré sú inak ťažko viditeľné. Výhodou tohto procesu je, že výsledný laser je ľahko viditeľný voľným okom aj v tmavých obrazoch. Na druhej strane, nevýhodou je vysoká časová náročnosť tohto procesu. Vplyv tejto funkcie na pôvodný obraz je zobrazený na obrázku 5.5.



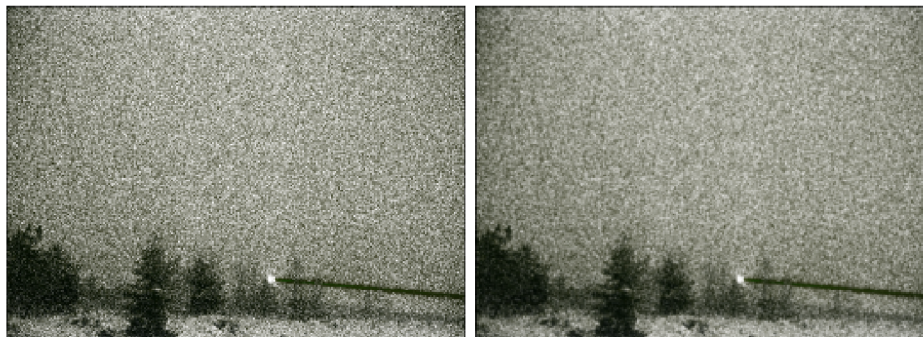
Obr. 5.5: Zmena farebnosti obrazu oproti pôvodnej snímke.

### Možné prístupy k odstráneniu šumu v obraze

Vzhľadom na to, že snímky boli nasnímané na otvorenom priestranstve počas večerných hodín, obsahujú veľké množstvo šumu, ktorý môže zhoršiť presnosť výsledok detekcie. Na odstránenie šumu sa zvyčajne používa vyhladzovací filter. Tento návrh obsahuje dva filtre, ktoré v tomto prípade dosiahli najlepšie výsledky pri odstraňovaní šumu – Gaussov filter a bilaterálny filter.

Gaussov filter je výhodný vďaka svojej rýchlosti a jednoduchšej implementácii. Jeho princíp spočíva v tom, že pre každý pixel vypočíta priemernú hodnotu jeho okolia pomocou

Gaussovej funkcie. Avšak, jeho efekt vyhladzovania spôsobuje rozmazanie obrazu aj v miestach s ostrými hranami, ktoré tvoria laserový lúč, čo môže viesť k nesprávnemu výpočtu výsledného vektora. Podrobnejšie informácie o Gaussovom filtri sú uvedené v kapitole 3.1.1, a jeho vplyv na snímku je zobrazený na obrázku 5.6.



Obr. 5.6: Aplikácia Gaussovho filtra (obr. vľavo) na obrázok s aplikovanou zelenou maskou (obr. vpravo). Obrázky boli pre lepšiu viditeľnosť upravené invertovaním čiernych pixelov.

Ďalším účinným filtrom je bilaterálny filter. Bilaterálny filter, rovnako ako Gaussov filter, vypočíta priemernú hodnotu najbližších pixelov, ale používa k tomu dve funkcie – jednu pre výpočet priestorovej vzdialenosti a druhú pre výpočet rozdielov v intenzite. Týmto spôsobom je schopný zachovať detaily a ostré hrany a stále spoľahlivo redukovať šum v obraze. Kvôli tejto odlišnosti však býva Bilaterálny filter pomalší ako Gaussov filter. Bilaterálny filter je podrobnejšie popísaný v kapitole 3.1.2.

### Výpočet prahu a jeho aplikácia

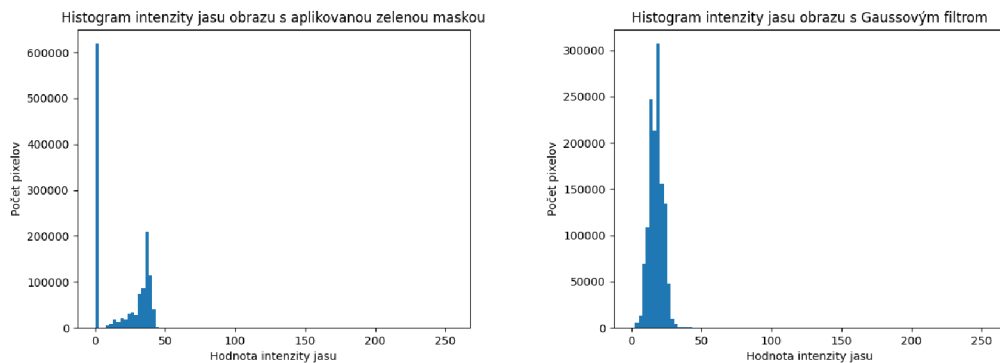
Prahovanie (angl. *thresholding*) je metóda, ktorá porovnáva vopred zvolenú hodnotu prahu s každou jasovou hodnotou pixelu v obraze. Ak je hodnota pixelu väčšia ako hodnota prahu, pixel sa zmení na hodnotu 1, v opačnom prípade na hodnotu 0. Týmto spôsobom sa využije vlastnosť laserového lúča, ktorý sa svojou vysokou intenzitou jasu výrazne odlišuje od okolia. Zároveň sa obraz prevedie do binárnej podoby, čo je nevyhnutné pre následnú detekciu laserového lúča a výpočet jeho trajektórie.

Nájdenie správnej prahovej hodnoty je kľúčové pre presné detekovanie a zjednodušenie následného spracovania obrazu. Existuje viacero spôsobov, ako túto hodnotu určiť. Jeden z možných postupov je určiť statickú hodnotu prahu vzhľadom na prostredie, ktoré kamera sníma. Tento postup je výhodný pre svoju rýchlosť, no nevýhodou sú meniace sa atmosférické podmienky, ktoré môžu skresľovať intenzitu jasu laseru, napríklad vplyvom hmly alebo vlhkosti vzduchu.

Ďalšou užitočnou metódou na určenie prahovej hodnoty je využitie histogramu. Histogram reprezentuje distribúciu hodnôt pixelov v obraze a poskytuje informácie o tom, v akom množstve sú zastúpené jednotlivé intenzity jasu. Na jeho vytvorenie sa najprv pretransformuje obraz do HSV modelu, ktorý sa následne rozdelí na jednotlivé zložky: H (hue, v prekl. odtieň), S (saturation, v prekl. sýtosť), V (value, v preklade hodnota). Histogram sa následne vytvorí na základe hodnoty V, ktorá reprezentuje intenzitu jasu pixela. Na základe

tvaru histogramu a jeho vrcholov sa určí vhodná prahová hodnota, ktorá je špecifická pre konkrétny obraz.

Pri úpravách vstupného obrazu sa mení aj tvar jeho histogramu. Aby sme zvýraznili laserový lúč, je vhodné histogram vytvoriť z obrazu, na ktorom bola aplikovaná zelená maska. Takýto obraz má vysoký kontrast v množstve jednotlivých hodnôt pixelov, s najväčším počtom tmavých a druhým najväčším počtom svetlých pixelov, ako je zobrazené na obrázku 5.7. Pri takomto rozdelení stačí z histogramu odčítať druhú najvyššiu hodnotu vrcholu, čím získame vhodnú prahovú hodnotu pre thresholding. Týmto spôsobom sa odfiltrujú zo snímky všetky body s nízkou intenzitou jasu.



Obr. 5.7: Vytvorenie histogramu intenzity jasu z obrázka, na ktorom bola aplikovaná zelená maska (obr. vľavo) a z obrázka, na ktorom bol aplikovaný aj vyhladzovací algoritmus (obr. vpravo).

V niektorých prípadoch je potrebné pri úprave obrazu eliminovať šum pomocou vyhladzovacieho filtra. Avšak nevýhodou tohto filtra je, že vyrovnáva hodnoty intenzity jasu pixelov, čím sa potláčia aj ich maximálne hodnoty. V dôsledku toho nie je možné z histogramu obrazu po aplikovaní vyhladzovacieho filtra jednoducho určiť vhodnú hodnotu prahu. Tento problém je zreteľne viditeľný aj na obr. 5.7.

Vzhľadom na to, že vyhladzovacie filtre potláčajú maximálne hodnoty histogramu intenzity jasu obrazu, najšť vhodnú prahovú hodnotu môže byť náročné. Jednou z možností, ako ju nájsť, je zobrať hodnotu z hornej hranice histogramu početností. Tento prístup je ukázaný aj v algoritme 1, ktorý vypočíta hodnotu percentilovej hranice z usporiadaného poľa pixelových hodnôt v HSV modeli obrazu.

V praxi sa najskôr usporiadajú hodnoty pixelov podľa ich veľkosti a následne sa zistí index pixela na percentilovej vrchnej hranici. Tento index sa použije na získanie hodnoty percentilu v usporiadanom poli.

## 5.2.2 Detekcia laserového lúča a výpočet trajektórie

### Cannyho hranový detektor

Cannyho hranový detektor je algoritmus, ktorý zvýrazňuje hrany v obraze a zároveň ho prevádza do binárnej podoby. Jeho funkčnosť je podrobne popísaná v kapitole 4.1.3. Aplikácia zjednodušuje následnú detekciu laserového lúča pomocou Houghovej transformácie, no napriek jeho účinnosti je časovo veľmi náročný. Ďalším problémom, ktorý môže nastať

---

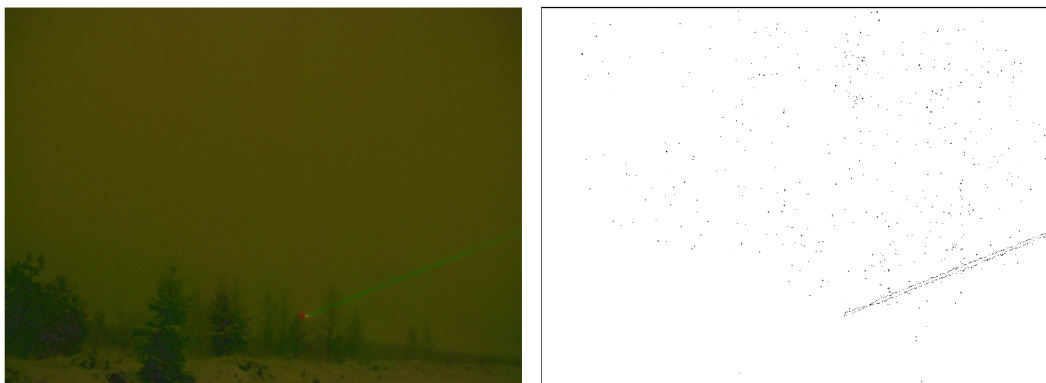
**Algoritmus 1:** Nájdenie prahovej hodnoty

---

**Vstup** :  $p \leftarrow$  percentilová hranica,  $arr \leftarrow$  usporiadané pole pixelov,  $n \leftarrow$  počet pixelov obrázka  
**Výstup**: hodnota prahu  
 $sum \leftarrow \text{sum}(arr)$ ;  $partialSum \leftarrow n \times \frac{p}{100}$ ;  $count \leftarrow 0$ ;  
**for**  $val$  *in*  $arr$  **do**  
     $count \leftarrow count + val$ ;  
    **if**  $count \geq partialSum$  **then**  
        **return**  $val$ ;  
    **end**  
**end**

---

je zaoblenie hrany v dôsledku rozptylu svetla laserového lúča, čo je nežiadúce, nakoľko následná Houghova transformácia takéto hrany nevie detekovať. Tento problém je znázornený na obrázku 5.8.

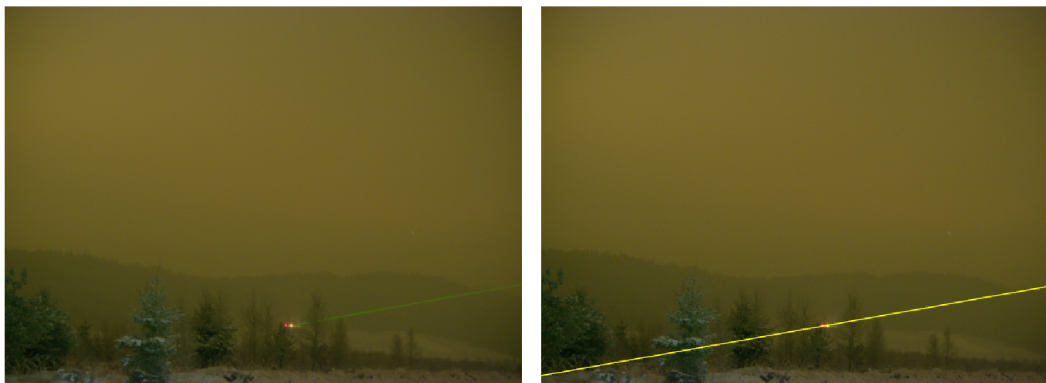


Obr. 5.8: Aplikácia Cannyho hranového detektoru. *Obrázky boli upravené pre zlepšenie viditeľnosti. Na obrázku vľavo sa zvýšil jas a obrázok vpravo sa invertoval.*

### Houghova transformácia

Houghova transformácia sa aplikuje na predspracovaný binárny obraz, aby sa detekoval vektor výslednej čiary. V určitých prípadoch pri vyznačení hrubšej laserovej čiary, alebo pri použití Cannyho algoritmu sa môže stať, že sa zdetekuje viac ako jedna čiara, teda jej hrany a čiary, z ktorých tento laserový lúč pozostáva. Z tohto dôvodu sa popri výpočte ukladajú hodnoty vektorov zdetekovaných čiar. Z nich sa zistí ich priemerný vektor a jeho priamka sa označí za výslednú. Houghova transformácia, aj napriek dôkladnému predspracovaniu obrazu, nedokáže vždy rozlíšiť čiaru vytvorenú laserom a horizontálne čiary, vytvorené pozadím. Z tohto dôvodu sa berie do úvahy, že v reálnom prípade by laserový lúč nenabudol horizontálny tvar, a preto pri každej zdetekovanej priamke sa vypočíta aj uhol naklonenia jej vektora. Pri malej hodnote uhla sa tento vektor nezaráta do výsledných. Týmto postupom sa filtrujú možné horizontálne čiary pozadia. Zdetekovanú čiaru v pôvodnej snímke je vidieť na obrázku 5.9.





Obr. 5.9: Zvýraznenie výsledného laserového lúča v pôvodnej snímke.

## 5.3 Implementácia systému a testovanie

Popis implementácie a testovania sa zameriava na analýzu vývoja systému, jeho kľúčových problémov a najmä na dôvody voľby konkrétnej metódy. Okrem toho sa sústreďuje na zlepšenie akcelerácie systému a presnosti detekcie.

Pri porovnaní rýchlosti programu, pri použití rôznych metód vo vývojovom procese, sa prevedie čiastočné testovanie pomocou vybranej dátovej sady, ktorá obsahuje snímky zachytávajúce laserový lúč. Sada pozostávala z rovnakého počtu tmavých a bledých snímok, aby sa zaručila objektivita. Toto testovanie sa vykonáva na počítačovej zostave, ktorej špecifikácia je uvedená v prílohe [A.1](#).

### 5.3.1 Použité technológie

Systém bol pôvodne navrhnutý v skriptovacom jazyku `Python`, pričom pre spracovanie obrazu sa využila knižnica `OpenCV`. Jazyk `Python` bol zvolený kvôli zjednodušeniu návrhu a vývoja, pretože ide o objektovo-orientovaný jazyk s vysokou úrovňou abstrakcie a množstvom zabudovaných funkcií. Knižnica `OpenCV` (Open Source Computer Vision Library), napísaná v jazykoch `C` a `C++`, bola použitá pre množstvo algoritmov na spracovanie obrazu a rozpoznávanie objektov.

S cieľom zvýšiť rýchlosť a výkon bol neskôr návrh prevedený do jazyka `C++`. Jazyk `C++` je kompilovaný jazyk, čo znamená, že sa kód prekladá do strojového kódu pred spustením programu, na rozdiel od interpretovaného jazyka `Python`, kde sa kód prekladá počas behu programu. `C++` tiež využíva statické typovanie a manuálne riadenie pamäte, čo môže prispieť k rýchlejšiemu vykonávaniu programu v porovnaní s dynamickým typovaním a automatickým riadením pamäte v jazyku `Python`.

### 5.3.2 Vývojový proces

Vo vývojovom procese implementácie existuje množstvo rôznych metód, ktoré môžu byť použité na riešenie danej problematiky. Z tohto dôvodu je potrebné vybrať správnu metódu a postup, prípadne ich kombináciu, ktorá zabezpečí rezistentnosť systému aj voči nehomogenite prostredia v snímkach, ktorá je spôsobená meniacimi sa atmosférickými podmienkami.

Celková implementácia sa skladá z niekoľkých základných krokov. Týmito krokmi sú farebná adaptácia obrazu, redukcia šumu, prahovanie obrazu, detekcia laserového lúča a vý-

počet trajektórie laserového lúča. Tieto kroky sa počas vývoja nezmenili, no menila sa postupnosť ich použitia a použité metódy, aby sa zlepšila detekcia a zrýchlenie programu. Jednotlivé kroky a metódy sú bližšie špecifikované v návrhu 5.2.

V prvom kroku sa zmenší rozmer vstupnej snímky, aby sa urýchlil výpočet systému. Týmto sa zároveň odstránia aj nepodstatné detaily i prípadný šum a umožní sa efektívnejšie spracovanie laserového lúča. Komprimácia vstupného obrazu má veľký vplyv na následnú rýchlosť programu, pričom na chybovosť v detekcii lúča má vplyv iba pri nízkych rozmeroch.

Následne bola realizovaná farebná adaptácia obrazu, ktorá pozostávala z úpravy jasu a kontrastu a aplikácie zelenej masky pre zvýraznenie laserového lúča. Tento postup bol však časovo náročný kvôli cyklickým výpočtom farebného kontrastu obrazu pre každú zmenu hodnoty jasu až do momentu, kedy bol laserový lúč dostatočne viditeľný a odlišný od okolia. Táto časová zložitosť sa zároveň zvyšovala v závislosti od rozmeru vstupného obrazu. Úprava jasu a kontrastu vstupného obrazu je viac špecifikovaná v časti 5.2.1.

Aby sa tento proces urýchlil, krok modifikácie jasu a kontrastu sa vynechal. Je to z toho dôvodu, že táto modifikácia má minimálny vplyv na výsledok detekcie, pretože okolie laserového lúča je možné orezať iba prostredníctvom masky, len s inými parametrami. Zjednodušením tohto procesu sa podarilo zrýchliť program o približne 800 ms (testované na zostave A.1 s rozmermi obrazu 768 x 562).

V prvotnom návrhu sa parametre zelenej masky, ktorá sa aplikuje prostredníctvom funkcie `cv::inRange()`, upravovali na základe priemernej intenzity jasu obrazu. Najskôr sa obraz pretransformoval do HSV modelu a vypočítala sa priemerná hodnota jasu zo zložky V (value, hodnota). V prípade tmavšieho obrazu sa tak aplikovali iné parametre ako pri bledšom. Avšak vplyv týchto odlišných parametrov bol minimálny, preto sa pre aplikáciu zelenej masky vo výslednom systéme použili statické hodnoty a tým sa eliminoval aj prevod do HSV modelu, čím sa Táto úprava urýchľuje program o približne 3000  $\mu$ s (testované na počítačovej zostave A.1).

Po aplikácii zelenej masky na snímku program spočíta počet jej nenulových bodov, vďaka čomu sa efektívne identifikuje prítomnosť zeleného lúča a minimalizuje možné výpočty, v prípade jeho absencie. Ak je hodnota súčtu nulových bodov nízka, systém vyhodnotí, že na danej snímke sa laserový lúč nenachádza a jej ďalšie spracovanie sa už nevykoná. V reálnom prípade sa predpokladá, že táto hodnota sa porovná so súčtom nenulových bodov v predchádzajúcej snímke. Ak predchádzajúca snímka nezachytila laserový lúč a aktuálna hodnota súčtu je výrazne vyššia ako v predchádzajúcom obraze, systém pokračuje v spracovaní aktuálnej snímky. Po identifikácii prítomnosti laserového lúča sa obraz prevedie na šedotónový, pomocou funkcie `cv::cvtColor()`.

Zachytenie laserového lúča v snímkach často zahrňuje veľký podiel nežiadúceho šumu, ktorý výrazne komplikuje následnú detekciu a spracovanie. Aby sa minimalizoval vplyv šumu, je nevyhnutné použiť vyhladzovací filter, ktorý ho zredukuje.

Pri výbere vhodného filtra je kľúčové zabezpečiť, aby bol dostatočne účinný pri redukcii šumu, ale zároveň neodstránil samotný laserový lúč, ktorý sa rozptyľuje do okolitého prostredia. Medzi najvhodnejšie filtre patrí Gaussov filter, ktorý sa vyznačuje jednoduchosťou a rýchlosťou spracovania. Zároveň ale nezachováva hrany a preto môže spôsobiť rozmazanie laserového lúča, čo môže skomplikovať jeho výslednú detekciu.

Medzi ďalšie vhodné filtre patrí bilaterálny filter, ktorý redukuje šum v obraze, ale zároveň zachováva hrany a detaily. Jeho nevýhodou je ale vyššia výpočtová náročnosť. Pri testovaní

vaní týchto filtrov, výpočet bilaterálneho vyhladzovacieho filtra, `cv::bilateralFilter()`, trval okolo 15 200  $\mu\text{s}$ . Pri použití iba Gaussovho filtra, `cv::GaussianBlur()`, bol čas výpočtu okolo 200  $\mu\text{s}$ .

Z tohto dôvodu sa v implementácii používajú oba filtre, ktoré sa aplikujú na snímku, v závislosti od jej priemernej hodnoty intenzity jasu. Jasnejšie snímky obsahujú viac svetelného šumu a preto sa spracujú pomocou Bilaterálneho filtra, zatiaľ čo na tmavšie snímky sa aplikuje Gaussov filter. Tento postup zaručí optimálne využitie výhod oboch filtrov. V ich kombinácii sa ich časy spriemerovali na približne 7 000  $\mu\text{s}$ . Tieto algoritmy boli testované na predpripravenej dátovej sade, ktorá obsahuje rovnaký počet tmavých a svetlých snímok. Testovanie prebehlo na počítačovej zostave uvedenej v prílohe A.1.

Pôvodný návrh algoritmu následne zahŕňal prevod obrazu do HSV-modelu a výpočet vhodnej hodnoty prahu zo zložky V (value, hodnota), ktorá reprezentuje intenzitu jasu pixelov. Potom sa obraz previedol do šedotónového a aplikovala sa nájdená hodnota prahu. Tento postup sa následne paralelizoval a bližšie je popísaný v 5.3.3. Počas ďalšieho vývoja bol postup modifikovaný tak, že obraz sa priamo prevádzal na šedotónový a z neho sa vypočítala vhodná hodnota prahu, čím odpadla nutnosť prevodu do HSV-modelu a algoritmus sa mierne zrýchlil. V konečnej implementácii sa obraz prevádza na šedotónový hneď po aplikovaní masky, ktorá je tiež popísaná v časti 5.3.2, a tým sa eliminuje potreba prevodu do HSV-modelu. Tým sa celkovo urýchlilo nájdenie vhodnej hodnoty prahu a aplikácia prahovania. Prahovanie sa vykonáva pomocou funkcie `cv::threshold()` z knižnice `OpenCV`.

Prahová hodnota mala pôvodne statickú hodnotu, čo v dôsledku toho spôsobovalo nepresnú detekciu laserového lúča na snímkach s rôznou hodnotou expozície. Jedným z riešení tohto problému bolo využitie histogramu intenzity jasu obrazu s aplikovanou zelenou maskou. Avšak pri využití vyhladzovacích filtrov sa vyhladili aj maximálne hodnoty jasu v obraze. Tento problém je podrobne popísaný v časti 5.2.1.

Ďalším riešením je odčítať hodnotu prahu z hornej časti histogramu jasu. Tento postup je tiež viac špecifikovaný v časti 5.2.1. Na hľadanie tejto hranice sa môže použiť sortovací algoritmus, ktorý usporiada obraz podľa jeho intenzity. Nevýhodou tohto algoritmu je vysoká časová náročnosť. Z tohto dôvodu sa zvolila iná metóda, ktorá najskôr vytvorí pole o veľkosti najväčšej možnej hodnoty prahu, teda 255. Následne prechádza pole pixelov obrazu a pri každom z nich zvýši výskyt hodnoty na indexe, ktorý korešponduje s jeho hodnotou v poli.

Týmto riešením sa vyšlo sortovaciemu algoritmu a program sa zrýchlil o približne 7000  $\mu\text{s}$  (testované s použitím sortovacieho filtra `std::sort()` z knižnice `algorithm` na počítačovej sade A.1). Horná hranica bola určená experimentálne s hodnotou 98, pri ktorej výsledok thresholdu dosahuje najlepšie výsledky.

Aby sa dosiahla vyššia efektivita detekcie, používa sa obvykle Cannyho hranový detektor spolu s Houghovou transformáciou. Detektor pomáha definovať hrany v obraze a oddeliť ich od ostatných častí obrazu a tiež slúži na redukciu šumu a iných irelevantných detailov. Následne sa Houghova transformácia použije na identifikáciu geometrických tvarov na základne týchto hrán. Avšak kvôli vyššej rézii Cannyho algoritmu, ktorej princíp je špecifikovaný v časti 4.1.3 a jeho následnému zaobleniu hrán, ktoré vzniká kvôli rozptylu svetla laserového lúča, ako je popísané aj v podkapitole 5.2.2, bol odstránený z výslednej implementácie. Vďaka jeho redukcii je v obraze viac rovných čiar, čo pomáha Houghovej transformácii detekovať laserový lúč s väčšou presnosťou. Po nevyužití Cannyho hranového detektoru sa program zrýchlil okolo 700  $\mu\text{s}$  (testované na počítačovej sade A.1 s využitím Cannyho algoritmu `cv::Canny()` z knižnice `OpenCV`).



Na detekciu laserového lúča sa využíva Houghova transformácia, pre ktorú knižnica OpenCV poskytuje dve verzie: štandardnú Houghovu transformáciu a pravdepodobnostnú Houghovu transformáciu. Štandardná Houghova transformácia sa implementuje pomocou funkcie `cv::HoughLines()`, ktorá vracia vektor dvojíc  $(\theta, \rho)$ , z ktorých sa dá jednoducho získať rovnica parabolickej priamky (5.6).

$$x \cos \theta + y \sin \theta = \rho \quad (5.6)$$

Funkcia `cv::HoughLines()` má niekoľko parametrov, ktoré ovplyvňujú výsledok detekcie priamok v obraze, medzi ktoré patria `Rho` a `Theta`, ktoré určujú presnosť parametrického priestoru, a `Threshold`, ktorý určuje minimálny počet priesečníkov v parametrickom priestore, potrebných na detekciu čiary.

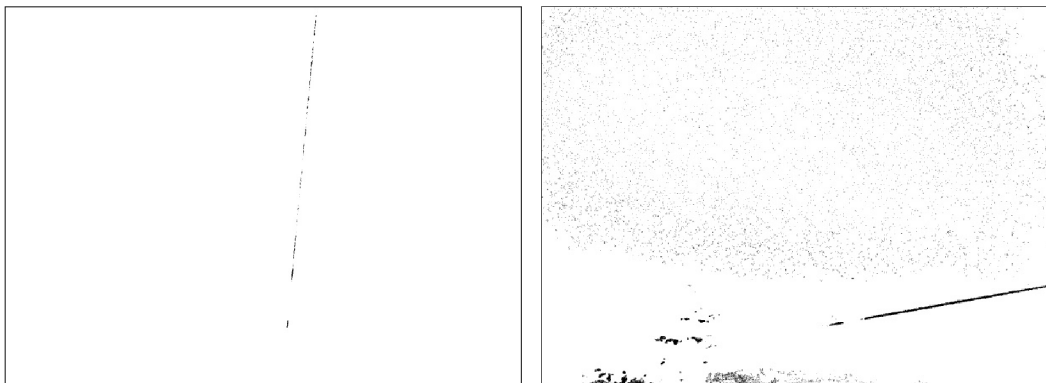
Pravdepodobnostná Houghova transformácia predstavuje modifikáciu štandardnej Houghovej transformácie a je implementovaná pomocou funkcie `cv::HoughLinesP()` v knižnici OpenCV. Na rozdiel od štandardnej Houghovej transformácie umožňuje špecifikovať ďalšie parametre, ako je minimálna dĺžka čiary a maximálna medzera medzi jednotlivými bodmi čiary, čím sa zvyšuje presnosť detekcie. Výstupom Houghovej transformácie sú súradnice začiatku a konca úsečiek.

Pravdepodobnostná Houghova transformácia je tiež rýchlejšia a efektívnejšia ako štandardná Houghova transformácia, pretože detekuje iba úseky línií v obraze, namiesto jej celej veľkosti. Rozdiel v aplikácii štandardnej Houghovej transformácie a pravdepodobnostnej Houghovej transformácii popisuje obrázok 5.10. Implementácia funkcie `cv::HoughLinesP()` z knižnice OpenCV sa následne modifikovala, aby sa dosiahla jej vyššia efektivita, čo je popísané v sekcii 5.3.3.



Obr. 5.10: Rozdiel v použití štandardnej Houghovej transformácie (obr. vľavo) a pravdepodobnostnej Houghovej transformácie (obr. vpravo) v skutočnom obraze.

Parametre Houghovej transformácie, ako `threshold`, minimálna dĺžka čiary a maximálna vzdialenosť bodov na čiare sa určujú v závislosti od konkrétnej snímky. Po aplikácii `threshold` sa rozhodne o tom, ako prísne bude táto metóda detekovať čiary. Preto sa po aplikácii `threshold` spočíta priemerná hodnota nenulových bodov obrazu. Ak je táto hodnota nízka, znamená to, že budú zvýraznené iba niektoré body čiary, a z tohto dôvodu treba použiť nižšie hodnoty parametrov. Pri vyššej priemernej hodnote nenulových bodov sa nastavujú vyššie parametre, aby sa eliminoval zvyšný zostávajúci šum. Rozdiel medzi vyššou a nižšou priemernou hodnotou obrazu po aplikácii `threshold` ukazuje obrázok 5.11.



Obr. 5.11: Porovnanie výsledných obrazov po aplikácii prahovania. Obrázky sú invertované pre lepšiu viditeľnosť.

Z detekovaných súradníc začiatku a konca úsečiek sa postupne vypočíta ich vektor a pre každú z nich sa určí uhol naklonenia vzhľadom na okraj snímky. V prípade, že je tento uhol veľmi nízky, tento vektor sa nezahrnie medzi výsledné vektory. Tento proces je bližšie popísaný v časti 5.2.2. Následne sa zo zostávajúcich vektorov vypočíta ich priemer, ktorý predstavuje výsledný vektor laserového lúča.

### 5.3.3 Paralelizácia systému

Paralelizácia je proces rozdeľovania úloh na menšie časti, ktoré môžu byť vykonané súčasne na rôznych výpočtových zariadeniach alebo procesoroch. Jej cieľom je zrýchliť výpočet, ktorý by inak musel byť vykonávaný sekvenčne. V niektorých prípadoch, pri použití paralelizácie, môže dôjsť k spomaleniu programu, a to v prípade, ak je čas potrebný na synchronizáciu paralelizovaných procesov vyšší ako samotný čas potrebný na výpočet týchto procesov alebo vplyvom zdieľaných výpočtových prostriedkov. Aby sa predišlo tomuto spomaleniu, všetky modifikované časti boli otestované a to na počítačovej zostave uvedenej v prílohe A.1.

V tomto systéme bola paralelizácia realizovaná v jazyku C++ pomocou knižnice `thread`, ktorá umožnila rozdeliť jednotlivé funkcie a spustiť ich na samostatných vláknach. Oproti iným metódam paralelizmu v C++, táto knižnica bola vybraná z dôvodu jej nízkej náročnosti na systémové prostriedky a umožnenie vysokej úrovne kontroly.

Jednou z oblastí programu, kde bola implementovaná paralelizácia, bolo pri aplikácii thresholdu. V pôvodnej verzii sa najskôr vypočítala optimálna hodnota prahu, potom sa obraz previedol do šedotónového a nakoniec sa naň aplikovalo prahovanie s vypočítanou hodnotou prahu. Tieto kroky sa rozdelili na dva procesy tak, že jeden počítal hodnotu prahu a druhý prevádzal pôvodný obraz na šedotónový. Výhodou tejto úpravy je, že procesy si vzájomne nezasahujú do zdieľanej pamäte a nemôže tak dôjsť k nežiadúcemu vzájomnému ovplyvneniu. Po otestovaní tejto zmeny, sa ale systém patrne spomalil. Vzhľadom na tento výsledok testovania sa táto zmena vo finálnej implementácii neaplikovala.

### Paralelizácia Houghovej transformácie

Paralelizmus sa využil najmä pri Houghovej transformácii, ktorej princíp spočíva v prevádzaní bodov z obrazového priestoru na krivky v parametrickom priestore, čo umožňuje

detekciu priamok v obraze. Podrobný popis sa nachádza v podkapitole 4.2. Konkrétne bola modifikovaná funkcia `cv::HoughLinesP()` z knižnice `OpenCV`, ktorá sa od klasického algoritmu líši náročnejšími požiadavkami na priamku, ako je jej dĺžka a maximálna vzdialenosť medzi bodmi. Funkcia `cv::HoughLinesP()` z knižnice `OpenCV` pozostáva z krokov:

### 1. Výpočet uhlov

V prvom kroku Houghovej transformácie sa pre všetky možné hodnoty uhlov  $\theta$  vypočíta hodnota funkcií  $\cos\theta$  a  $\sin\theta$  a následne sa uloží do zoznamu. Cieľom je obmedziť opakované výpočty  $x\cos\theta + y\sin\theta = p$  pre každý nenulový bod  $(x, y)$  v obrazovom priestore.

### 2. Vytvorenie masky a zoznamu nenulových bodov

Nenulové body z obrazu sa prenesú do masky, ktorá slúži ako priestor pre ich ďalšie spracovanie a súradnice týchto nenulových bodov  $(x, y)$  sa uložia do zoznamu.

### 3. Výpočet parametrov priamky z nenulových bodov obrazu

Postupne algoritmus prechádza zoznamom nenulových bodov, ktorý sa vytvoril v kroku 2. Pre každý bod v tomto zozname sa iteratívne prechádza cez zoznam vypočítaných hodnôt  $\cos\theta$  a  $\sin\theta$ , ktorý sa vytvoril v bode 1. Následne sa pre daný bod v každej iterácii vypočíta hodnota  $p$  pomocou jeho súradníc a rovnice  $x\cos\theta + y\sin\theta = p$ . Počet priesečníkov v parametrickom priestore na súradnici  $(\theta, p)$  sa zvýši o jedna. Ak má bod dostatočný počet priesečníkov, zodpovedajúce súradnice  $(\theta, p)$  predstavujú parametre možnej priamky v obraze. Tento algoritmus je tiež znázornený v kóde 2.

### 4. Nájdenie priamky a jej bodov v obrazovom priestore

Pri nájdení konštanty  $\theta$  a  $p$  priamky algoritmus overuje, či táto priamka spĺňa ďalšie kritéria, ako minimálnu dĺžku a maximálnu vzdialenosť jej bodov. Postupne prechádza bodmi, ktoré ležia na danej priamke, a ak je vzdialenosť medzi nimi menšia ako vstupná hodnota pre maximálnu medzeru a zároveň je dĺžka čiary väčšia ako vstupná hodnota minimálnej dĺžky, tak táto čiara sa považuje za výslednú a jej parametre sa uložia do koncového zoznamu.

### 5. Odstránenie redundancie

Algoritmus po spracovaní čiary znovu prechádza pozdĺž nej a odstraňuje body masky, ktoré ležia na danej čiare a ich vykreslené funkcie v parametrickom priestore. Týmto spôsobom sa zabezpečí, že tieto body už nebudú viac spracované.

Jednou z častí, ktoré boli v Houghovej transformácii modifikované pre paralelizáciu systému, sú kroky 1 a 2. Tieto kroky pracujú nezávisle na sebe a nezasahujú do zdieľanej pamäte. Vďaka tomu nie je potrebné použiť zámky alebo iné synchronizačné úpravy. Pomocou tejto zmeny sa ale program spomalil o približne 20%, preto táto časť nebola vo finálnej podobe paralelizovaná.

Následne boli upravené aj ostatné body, kde sa vyskytol problém so vzájomným zasahovaním do spoločnej pamäte. Medzi tieto kritické miesta patrí zoznam nenulových bodov obrazu, maska pôvodného obrazu a pole parametrického priestoru. Je preto veľmi dôležité použiť optimálny spôsob úprav, zabezpečiť použitie zámkov a rozdeliť procesy tak, aby sa predišlo nežiadúcemu vzájomnému prepisu hodnôt.

Tretí krok, ktorý je ukázaný v algoritme 2, je možné upraviť tak, že každý proces prechádza nenulové body obrazu, kým pole nenulových bodov nie je prázdne. Proces najprv

---

**Algoritmus 2:** Výpočet parametrov priamky z nenulových bodov obrazu

---

**Vstup :** *nonZeroPoints*  $\leftarrow$  pole nenulových bodov obrazu;  
*\*numNonZeroPoints*  $\leftarrow$  počet nenulových bodov v obraze;  
*angleCounted*  $\leftarrow$  pole predpočítaných hodnôt sínusu a kosínusu pre celú šírku parametrického priestoru; *parameterSpace*  $\leftarrow$  parametrický priestor Houghovej transformácie; *threshold*  $\leftarrow$  minimálny počet priesečníkov v parametrickom priestore pre výslednú priamku  
**Výstup:** parametre výslednej priamky

```
while *numNonZeroPoints > 0 do  
    point  $\leftarrow$  nonZeroPoints[ $-$  *numNonZeroPoints];  
    for n  $\leftarrow$  0 to angleCounted.size() - 1 do  
        r  $\leftarrow$  point.y  $\times$  angleCounted[n  $\times$  2] + point.x  $\times$  angleCounted[n  $\times$  2 + 1];  
        parameterSpace[n][r] ++;  
        if parameterSpace[n][r] > threshold then  
            return r, n;  
        end  
    end  
end
```

---

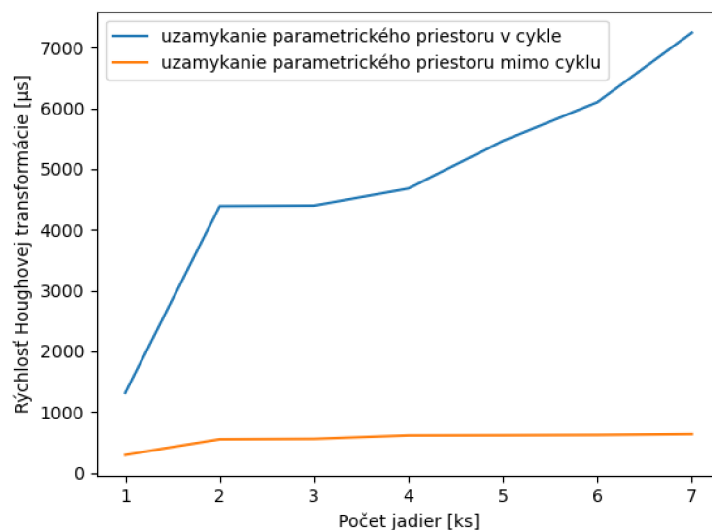
vyberie jeden nespracovaný bod z tohto poľa, zníži veľkosť poľa a skontroluje, či sa bod nachádza v maske obrázku. Ak sa bod nenachádza v maske, znamená to, že sa už spracoval pri kontrole kritérií čiary v kroku 4. Následne proces zvýši počet priesečníkov v parametrickom priestore, na základe súradníc spracovávaného bodu a výsledných hodnôt funkcie. Ak je počet priesečníkov väčší ako stanovený prah, proces skontroluje ďalšie kritéria pre výslednú čiaru. Potom program opäť vyberie ďalší nenulový bod z poľa a spracuje ho. Problémom pri tejto modifikácii je nutnosť uzamknúť parametrický priestor a veľkosti poľa nenulových bodov pri ich úprave.

Jednou z možností je uzamknúť tieto miesta priamo pred ich modifikáciou, aby sa zabránilo súbežnému zápisu. Avšak, keďže modifikácia parametrického priestoru sa nachádza vo vnútri cyklu, zamykanie sa opakuje v každej iterácii a spôsobuje vysokú časovú náročnosť na réžiu procesov. V dôsledku tejto vysokej réžii sa výrazne spomaľuje výpočet Houghovej transformácie o najmenej 500%.

Aby sa tomuto spomaleniu programu predišlo, je nutné uzamknúť prístup do spoločnej pamäte pred vstupom do cyklu, zapisujúceho priesečníky do parametrického priestoru. Okrem toho je potrebné zistiť optimálny počet procesov, ktorý zabezpečí rýchly priebeh programu bez zbytočného spomalenia, v dôsledku vysokej réžie ostatných procesov. Na základe testovania sa zistilo, že najoptimálnejšie je využiť iba jedno jadro, a tak táto časť programu nebude využívať paralelizmus. Vplyv počtu jadier na rýchlosť Houghovej transformácie a porovnanie rýchlosti s uzamykaním spoločnej pamäte parametrického priestoru v cykle a mimo cyklu porovnáva graf 5.12.

V kroku 4 proces postupne prechádza pozdĺž čiary od nájdeného bodu v kroku 3 a postupne kontroluje, či táto čiara spĺňa ďalšie požiadavky. Tento krok je možné paralelizovať tak, že každý jeden smer bude spracovávať jeden proces. V tejto časti by bola paralelizácia vhodná, pretože nedochádza k modifikácii spoločnej pamäte. V prípade, že sa zistí, že čiara





Obr. 5.12: Porovnanie vplyvu počtu využitých jadier s uzamykaním zdieľanej pamäte parametrického priestoru v cykle a mimo cyklu na rýchlosť Houghovej transformácie.

nesplňa niektorú z podmienok, spracovanie tejto čiary sa ukončí a následne sa odstráni. Aby sa zabránilo viacnásobnej kontrole kritérií jednej čiary, je nutné uzamknúť tento krok. Okrem toho je potrebné zabezpečiť opakovanú kontrolu existencie nájdeného bodu v maske obrázka, pretože už mohol byť spracovaný a odstránený iným procesom. Pri testovaní sa ale zistilo, že použitím paralelizácie sa výpočet Houghovej transformácie spomalil o približne 70 %.

V kroku 5 program postupne odstraňuje spracovanú čiaru z masky obrazu, aby sa zabránilo jej opätovnému spracovaniu. Taktiež sa pre každý bod na danej čiare odstráni jeho vykreslená funkcia v parametrickom priestore. Program postupuje rovnako ako v bode 4, kedy proces postupne prechádza čiaru v oboch smeroch od bodu, ktorý sa ako prvý našiel na danej čiare. V tomto kroku by sa dala paralelizácia uplatniť podobne ako v predchádzajúcom kroku, takže každý proces bude prechádzať jeden smer.

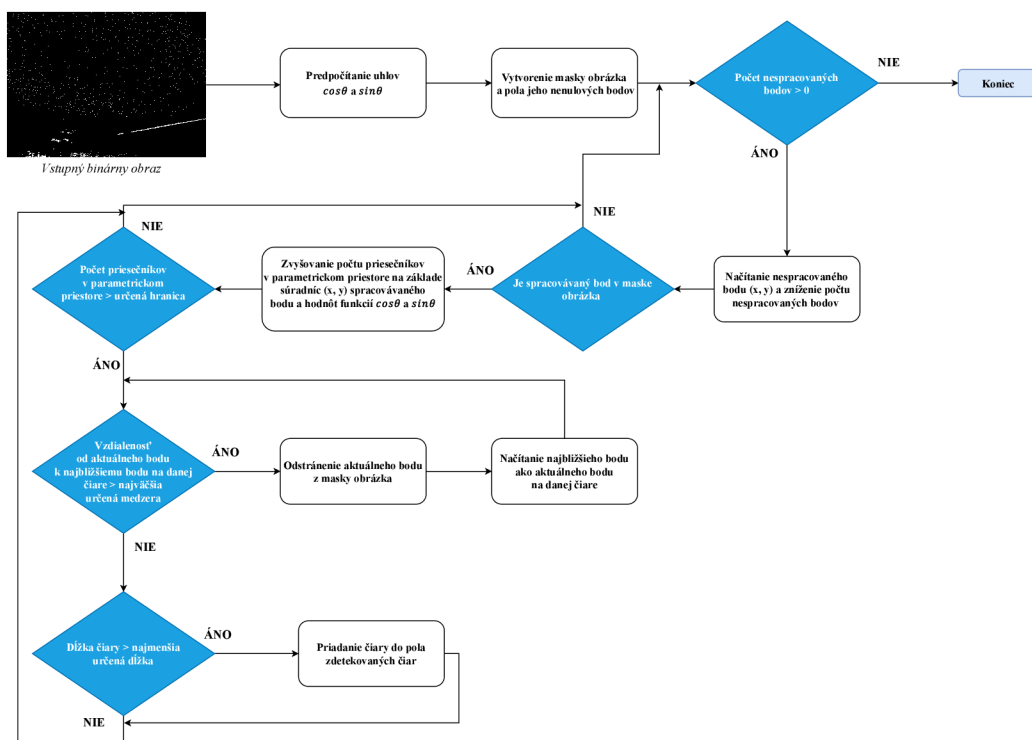
V tejto časti ale nastáva problém s modifikáciou spoločnej pamäte, ktorou je maska obrazu a parametrický priestor. Jedným zo spôsobov, ako upraviť tento proces, je neodstraňovať krivky z parametrického priestoru, kvôli jeho časovej náročnosti (pri každom bode sa musí iterovať cez celú šírku parametrického priestoru a znížiť počet priesečníkov) a taktiež kvôli vysokému množstvu uzamykaní procesov.

Pri odstraňovaní bodov z masky obrázka sa môže buď uzamknúť daný priestor pred jeho úpravou, alebo si každý proces vytvorí lokálnu pamäť, kde si uloží súradnice bodov, ktoré sa majú odstrániť. Tieto body sa z masky odstránia až po tom, ako sa všetky potrebné body nájdu a uložia. Pri otestovaní sa zistilo, že aj táto modifikácia za použitia paralelizácie spomaľuje program o približne 50%.

Ďalšou možnosťou úpravy algoritmu môže byť odstránenie celého 5. kroku, čím nebude nutné opätovne prechádzať spracovanú čiaru. Pri nájdení bodov v kroku 4, ktoré patria danej čiare, si každý proces tieto body uloží zvlášť do svojej pamäte. Po spracovaní danej

čiar sa tieto uložené body odstránia z masky obrazu. Pri použití paralelizácie sa Houghova transformácia, pri tejto úprave, spomalí o približne 30% oproti pôvodnej funkcii.

Naopak, pri použití sekvenčného programu v tejto modifikácii je Houghova transformácia o približne 25% rýchlejšia, čo na testovacom zariadení A.1 je okolo 100  $\mu$ s, v porovnaní s pôvodnou funkciou `cv::HoughLinesP()` z knižnice `OpenCV`. Program v bode 4 tak spracováva body, ležiace na nájdenej čiare, pokiaľ ich vzdialenosť nie je menšia ako vstupná hodnota pre maximálnu medzeru. Po spracovaní každého bodu sa zároveň tento bod odstráni aj z masky obrazu. Týmto spôsobom sa upravila obecná funkcia Houghovej transformácie a vytvorila sa špecifickejšia verzia pre účely detekcie laserového lúča, ktorej postup je zjednodušene popísaný na vývojovom diagrame 5.13.

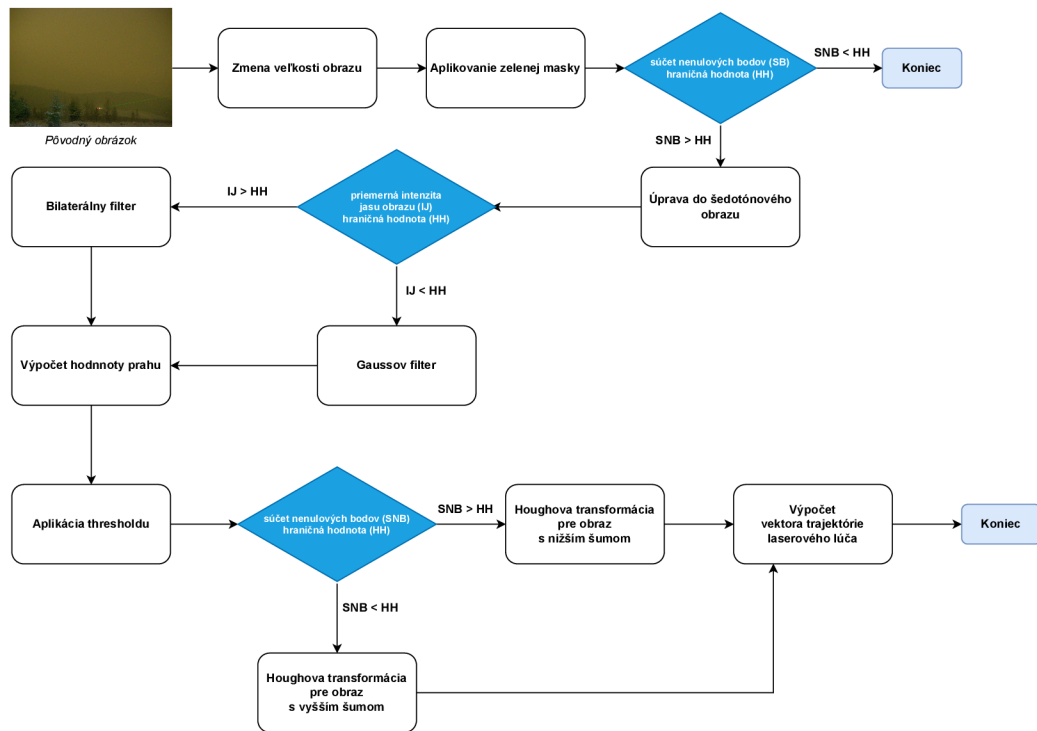


Obr. 5.13: Popis Houghovej transformácie vo výslednej implementácii.

## 5.4 Popis výslednej implementácie

Na základe vývoja systému a spomenutých metód sa vytvorila zostava výslednej implementácie, ktorá sa skladá z krokov popísaných v diagrame 5.14. Týmito krokmi sú: 1. zmena veľkosti obrazu, 2. aplikácia zelenej masky (po aplikácii masky sa spočíta počet nenulových bodov; ak je tento súčet menší ako stanovená hraničná hodnota, obraz sa už ďalej nebude spracovávať), 3. transformácia do šedotónového obrazu, 4. aplikácia vyhladzovacieho filtra (typ vyhladzovacieho filtra sa aplikuje v závislosti na intenzite jasu obrazu), 5. výpočet hodnoty prahu, 6. aplikácia thresholdu, 7. aplikácia modifikovanej Houghovej transformácie (hodnota parametrov Houghovej transformácie závisí od množstva zostávajúceho šumu v obraze) a 8. výpočet výsledného vektora laserového lúča.

Vzniknutý systém sa tak vďaka postupu týchto bodov zameriava na dosiahnutie presnej a rýchlej detekcie laserového lúča, ktorý dokáže pracovať s rôznymi typmi snímok, pochádzajúcimi z rozličných atmosferických podmienok.



Obr. 5.14: Postup použitia metód vo výslednej implementácii.

## Kapitola 6

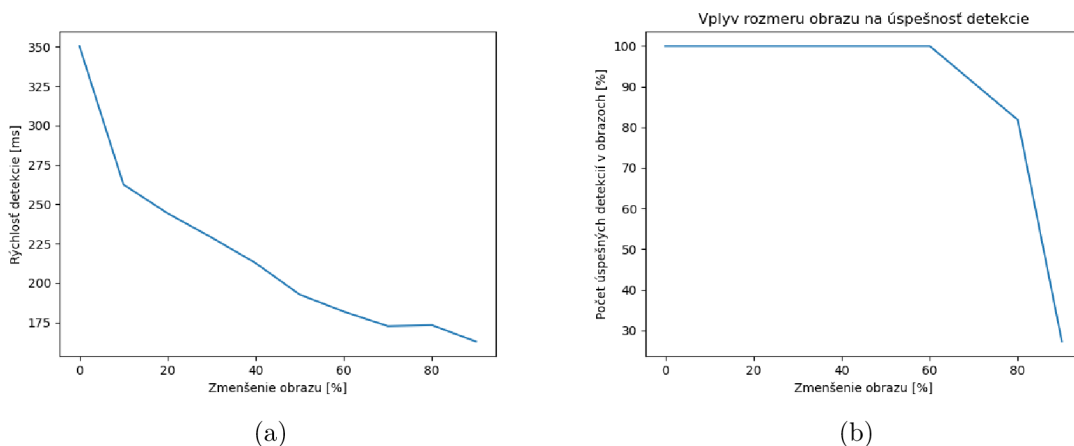
# Testovanie systému a zhrnutie dosiahnutých výsledkov

### 6.1 Testovanie systému

Po dokončení finálnej implementácie programu na detekciu laserových lúčov sa previedlo testovanie s cieľom zistiť vplyv veľkosti vstupnej snímky a parametrov na rýchlosť spracovania obrazu. Z tohto dôvodu sa vytvorili rôzne verzie programu s upravenými metódami, aby bolo možné porovnať tieto vplyvy a v prípadnom ďalšom postupe upraviť program na základe výsledkov z týchto testov.

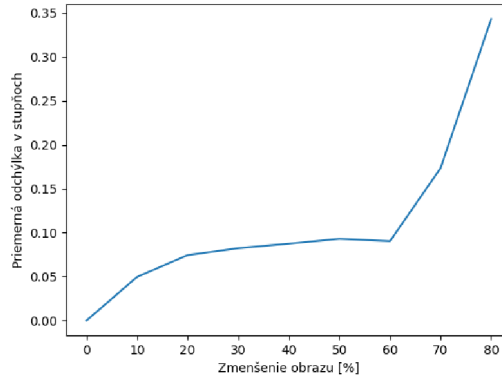
#### 6.1.1 Vplyv rozmeru veľkosti obrazu na detekciu laserového lúča

Vstupná snímka v programe na detekciu laserového lúča najskôr zmení svoju veľkosť, s cieľom zrýchliť celkové spracovanie obrazu. Veľkosť snímky má vplyv na rýchlosť detekcie, ale aj na jej presnosť. Preto bolo vykonané testovanie, ktoré malo za úlohu zistiť veľkosť zrýchlenia a narastajúce chyby pri postupnom znižovaní veľkosti snímky. Testovanie sa uskutočnilo na počítačovej zostave, uvedenej v prílohe A.1, s použitím nameranej dátovej sady, s veľkosťou snímok 4096 x 3000. Výsledky testovania sú zobrazené v grafoch 6.1.



Obr. 6.1: Porovnanie vplyvu rozmeru obrazu na rýchlosť (a) a úspešnosť (b) detekcie.

Po otestovaní programu na rôznych veľkostiach snímok sa zistilo, že vplyv na úspešnosť detekcie sa prejavuje až pri zmenšení pôvodnej snímky o 70% a rýchlosť detekcie sa kvadraticky znižuje v závislosti od veľkosti obrazu.



Obr. 6.2: Vplyv zmenšovania rozlíšenia obrazu na odchýlku hodnoty uhla vektora od veľkosti uhla, v pôvodnej veľkosti obrazu.

Zmenšovanie rozlíšenia obrazu má vplyv aj na presnosť detekcie, čo vyjadruje odchýlka hodnoty uhla vektora oproti veľkosti uhla, v pôvodnej veľkosti obrazu. Túto zmenu popisuje graf 6.2, ktorý porovnáva priemernú odchýlku veľkosti uhla  $\overline{\Delta x}$ , pre každú veľkosť obrazu vzhľadom k nameranému uhlu v pôvodnej veľkosti  $X$ . Odchýlka pre každý obraz  $\Delta x$  je daná rozdielom medzi nameraným uhlom v pôvodnej veľkosti  $X$  a veľkosťou uhla vektora v danom zmenšenom obraze  $x$ , čiže  $\Delta x = |X - x|$ . Zo zaznamenaných odchýliek pre každú veľkosť obrazu sa vypočítala priemerná hodnota  $\overline{\Delta x}$  pomocou aritmetického priemeru (6.1), kde  $n$  je počet testovaných snímok.

$$\overline{\Delta x} = \frac{1}{n} \sum_{i=1}^n \Delta x_i = \frac{1}{n} \sum_{i=1}^n |X - x_i| \quad (6.1)$$

Z otestovaných dát vyplýva, že sa výsledný uhol vektora mení veľmi málo a to aj pri nízkych rozmeroch obrázka. Avšak, čím je menšie rozlíšenie obrazu tým je nižšia pravdepodobnosť detekcie laserového lúča.

### 6.1.2 Porovnanie rýchlosti verzií programu na rôznych systémoch

Z programu na detekciu laserového lúča sa vytvorili 4 rôzne verzie implementácie, aby sa porovnal vplyv odlišných parametrov a metód na rýchlosť spracovania. Tieto verzie boli testované na rôznych počítačových zostavách, ktorých špecifikácie sú uvedené v prílohe A. Použitie viacerých zariadení tiež umožnilo zistiť, čo rýchlosť spracovania najviac ovplyvňuje pre prípadné určenie najvhodnejšieho zariadenia pre spracovanie.

Testovanie bolo prevedené na dvoch rôznych dátových sadách, obsahujúce záznamy laserového lúča z rôznych uhlov z viacerých kamier. Jedna z týchto sád obsahuje jasnejšie snímky, ktoré boli zosnímané s vyšším expozičným časom a vyššou hodnotou zisku. Druhá sada pozostáva zo snímok s nižším jasom, ktoré boli zosnímané s nižším expozičným časom

a nižšou hodnotou zisku. Toto rozdelenie na dve dátové sady umožňuje porovnať vplyv intenzity jasu snímky na rýchlosť spracovania, keďže na snímky s vyšším jasom sa využívajú zložitejšie metódy spracovania, napríklad Bilaterálny filter na redukciu šumu v obraze.

Pomocou testovacieho skriptu sa postupne spúšťal program na detekciu s jednotlivými vstupnými snímkami z dátovej sady. Toto testovanie sa na každej zostave zopakovalo viackrát, aby sa dosiahli čo najpresnejšie výsledky. Pre každú zostavu sa zaznamenali hodnoty rýchlosti, z ktorých sa určil medián, maximálna a minimálna hodnota a smerodajná odchýlka.

## Verzia 1

Prvá testovacia verzia je špecifikovaná v časti 5.14 a predstavuje finálnu implementáciu programu, kde sa veľkosť obrazu na spracovanie zmenší o 40%.

V testovaní sa potvrdilo, že táto akcelerovaná verzia dosahuje najnižšiu priemernú rýchlosť z pomedzi všetkých verzií programu. Avšak treba zohľadniť, že kvôli zmenšeniu vstupného obrazu, môže dôjsť k menej presnej detekcii priamky laserového lúča v porovnaní s druhou verziou, ktorá spracováva vstupný obraz v pôvodnej veľkosti.

Typ zostavy	Tmavé snímky [ms]				Svetlé snímky [ms]			
	Medián	Min	Max	Odchýlka	Medián	Min	Max	Odchýlka
A.1	216	160	297	33,05	261	250	282	6,44
A.2	293	206	435	48,3	383	343	443	24,61
A.3	472	355	653	81,55	608	586	630	11,06
A.4	308	237	401	44,64	372	353	395	7,8

Tabuľka 6.1: Porovnanie rýchlosti pre verziu 1 programu na detekciu laserového lúča na rôznych počítačových zostavách.

## Verzia 2

Druhá verzia programu je modifikáciou prvej verzie, ktorá je tiež špecifikovaná v časti 5.14. Oproti nej sa líši v tom, že zachováva pôvodnú veľkosť vstupného obrazu, čo má za následok pomalšie spracovanie. Priemerný nárast rýchlosti je tak o 15 %. Avšak vzhľadom na zachovanie pôvodnej veľkosti obrazu, táto verzia poskytuje najpresnejšiu detekciu laserového lúča.

Typ zostavy	Tmavé snímky [ms]				Svetlé snímky [ms]			
	Medián	Min	Max	Odchýlka	Medián	Min	Max	Odchýlka
A.1	289	233	414	42,32	354	340	440	13,75
A.2	391	299	643	65,1	570	506	620	22,47
A.3	593	488	835	88,2	792	769	838	14,94
A.4	404	333	636	52,0	481	458	514	10,25

Tabuľka 6.2: Porovnanie rýchlosti pre verziu 2 programu na detekciu laserového lúča na rôznych počítačových zostavách.



### Verzia 3

V tretej verzii programu na detekciu laserového lúča sa porovnával vplyv modifikovanej Houghovej transformácie a pôvodnej funkcie `cv::HoughLineP()` z knižnice OpenCV. Táto verzia predstavuje modifikovanú prvú verziu, kde sa obraz zmenšuje o 40%, avšak na detekciu laserového lúča sa používa pôvodná funkcia `cv::HoughLineP()`. Po testovaní sa zistilo, že modifikácia Houghovej transformácie má vplyv na rýchlosť programu, ale nie výrazný. Priemerná hodnota rýchlosti pri použití pôvodnej funkcie sa zvýšila o 4%.

Typ zostavy	Tmavé snímky [ms]				Svetlé snímky [ms]			
	Medián	Min	Max	Odchýlka	Medián	Min	Max	Odchýlka
<b>A.1</b>	221,5	159	303	34,89	271	259	283	5,63
<b>A.2</b>	306	203	486	59,08	414	364	470	24,42
<b>A.3</b>	494	356	664	85,25	629	599	648	12,41
<b>A.4</b>	321	243	411	46,1	385	363	405	8,71

Tabuľka 6.3: Porovnanie rýchlosti pre verziu 3 programu na detekciu laserového lúča na rôznych počítačových zostavách.

### Verzia 4

Štvrtá verzia programu porovnáva rýchlosť pôvodného návrhu na detekciu laserového lúča s jeho akcelerovanou verziou. Oproti pôvodnému návrhu sa ale táto verzia líši tým, že nezahŕňa farebnú adaptáciu pomocou metriky farebnosti obrazu, ktorá je opísaná v časti 5.2.1. Dôvodom je jej pomalá rýchlosť spracovania, ktorá sa pohybuje okolo 800 ms pri rozmeroch snímky 768 x 562. Z testovania vyplýva, že v priebehu vývoja sa podarilo zrýchliť program o približne 40%.

Typ zostavy	Tmavé snímky [ms]				Svetlé snímky [ms]			
	Medián	Min	Max	Odchýlka	Medián	Min	Max	Odchýlka
<b>A.1</b>	450	360	803	50,4	867	580	1250	98,95
<b>A.2</b>	493	417	595	47,97	726,5	574	922	104,97
<b>A.3</b>	907	635	1 036	143,7	1 347	1 054	1 739	210
<b>A.4</b>	516	378	842	86,81	909	571	1 328	198,52

Tabuľka 6.4: Porovnanie rýchlosti pre verziu 4 programu na detekciu laserového lúča na rôznych počítačových zostavách.

Program dosahuje najnižšiu priemernú rýchlosť na zostave **A.1**. Následne na zostave **A.2**, **A.4** a nakoniec **A.3**. Hodnoty rýchlosti sú ovplyvnené rôznymi vlastnosťami týchto zariadení, ako je najmä taktovacia frekvencia procesora. Zariadenie **A.1** dosahuje maximálnu frekvenciu 4,7 GHz, čo je najvyššia hodnota z pomedzi ostatných zostáv. Zostava **A.2** má frekvenciu 3,6 GHz s maximálnou turbo frekvenciou 3,9 GHz, **A.4** má 2,10 GHz s maximálnou frekvenciou 3,2 GHz a **A.3** má 1,8 GHz bez možnosti turbo frekvencie.

Okrem taktovacej frekvencie môžu ďalšie vlastnosti zostáv ovplyvniť rýchlosť programu, ako napríklad veľkosť pamäte cache a inštrukčná sada alebo typ výrobcu. Počet jadier, prípadne vlákien procesoru by však nemal mať vplyv na rýchlosť programu, pretože program sa vykonáva sekvenčne.

## 6.2 Úspešnosť detekcie laserového lúča

Na zistenie úspešnosti detekcie laserového lúča sa použili snímky, ktoré zachytili laserový lúč z nameranej dátovej sady. Táto dátová sada vznikla v rámci troch rozsiahlych meraní, ktoré sú bližšie špecifikované v podkapitole 5.1. Z týchto snímok sa vybrali iba tie, ktoré boli zhotovené farebnou kamerou bez použitia farebného filtra a to z dôvodu najlepšej viditeľnosti laserového lúča.

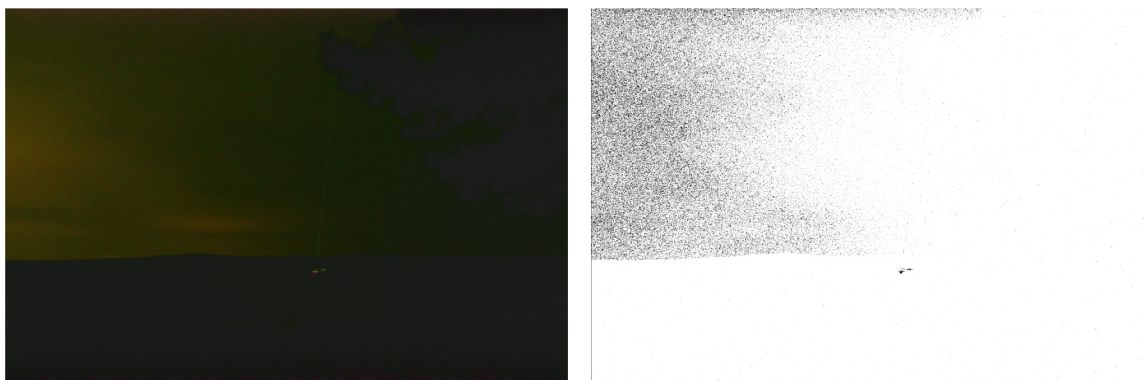
Súčasťou druhého a tretieho merania bola sekvencia snímok s narastajúcim časom expozície a pre každú dĺžku expozície bol menený aj zisk kamery. Takáto sekvencia snímok vytvára snímanie laserového lúča v jednej pozícii. V rámci testovania úspešnosti sa tak otestuje každá snímka danej sekvencie. Ak sa aspoň na jednej snímke danej sekvencie podarí zdetekovať laserový lúč, ide o úspešnú detekciu laserového lúča v danej sekvencii. Celková úspešnosť meraní, ktorých výsledky sú zobrazené v tabuľke 6.5, je daná pomerom počtu úspešných sekvencií k počtu všetkých sekvencií.

Číslo merania	Meranie 1	Meranie 2	Meranie 3
Úspešnosť detekcie	84%	81,81%	22,45%

Tabuľka 6.5: Porovnanie úspešnosti detekcie v jednotlivých meraniach.

Znížená úspešnosť detekcie laserového lúča v treťom meraní je spôsobená tým, že sa zmenil smer snímania lasera oproti ostatným meraniam. Tým sa narazilo na problém, kedy pozadie laserového lúča bolo nerovnomerne osvetlené, čo sťažilo nájdenie vhodnej hodnoty prahu na zvýraznenie laserového lúča. V jednej časti snímok, ako je ukázané aj na obrázku 6.3, sa tak ocitlo množstvo šumu, ktoré bolo problém odstrániť. Jedným z možných riešení tohto problému by mohlo byť, v prípadnom následnom postupe, rozdelenie vstupného obrazu na viac častí, ktoré by mohli byť spracovávané paralelne. Laserový lúč by sa tak mohol hľadať vo všetkých častiach súčasne a snímka by nebola ovplyvnená nerovnomerným osvetlením okolia.

Úspešnosť detekcie laserového lúča závisí nielen na rovnomernom osvetlení pozadia, ale aj na parametroch snímania ako je expozičný čas a zisk. Tabuľka 6.6 zobrazuje vplyv expozičného času na úspešnosť detekcie len druhého merania, pretože pri prvom meraní sa tieto hodnoty nezaznamenali a tretie meranie bolo ovplyvnené najmä nehomogenitou prostredia. Získané výsledky jasne ukazujú, že program vie, že zvýšenie expozičného času vedie k výraznému zlepšeniu úspešnosti detekcie. Je ale potrebné poznamenať, že v reálnom prípade pri snímaní osliňovania lietadla, bude laserový lúč nestabilný a preto by mal byť snímaný s čo najnižšou uzávierkou.



(a) Pôvodná snímka. Obrázok je upravený pomocou zvýšenia jasnosti, kvôli lepšej viditeľnosti. (b) Snímka po aplikácii prahovania. Upravená invertovaním, kvôli zvýšeniu viditeľnosti.

Obr. 6.3: Aplikácia prahovania na obrázok s nerovnomerným osvetlením okolia.

Hodnota expozičného času	0.1 - 0.2 s	1 - 2 s	3 - 4 s
Úspešnosť detekcie	21%	68,18%	81,81%

Tabuľka 6.6: Porovnanie úspešnosti detekcie na snímkach z druhého merania, ktoré boli zhotovené s rôznym expozičným časom.

Úspešnosť taktiež ovplyvňuje množstvo šumu, spôsobené atmosférickými podmienkami ako je, napríklad, hmla. Vtedy sa môže stať, že takto spôsobený väčší šum nevie program rozoznať od laserového lúča a môže byť tento lúč odstránený spolu s ním. Ďalším faktorom je veľkosť zastúpeného laserového lúča v snímke. Ak je laserový lúč príliš krátky, môže sa stať, že nebude zdetekovaný kvôli nastaveným parametrom v Houghovej transformácii, ktoré definujú jeho najmenšiu možnú dĺžku. Toto opatrenie slúži na predchádzanie chybných detekcií pri zastúpení väčšieho množstva šumu.

Celkovo vie program laserový lúč zdetekovať v rôznych druhoch snímkov. Napríklad aj v tmavších, zostrojených s nižšou hodnotou expozičného času a nižším ziskom, kedy laserový lúč nie je viditeľný voľným okom. Túto detekciu, ale môžu ovplyvniť rôzne faktory, na ktoré treba brať ohľad.

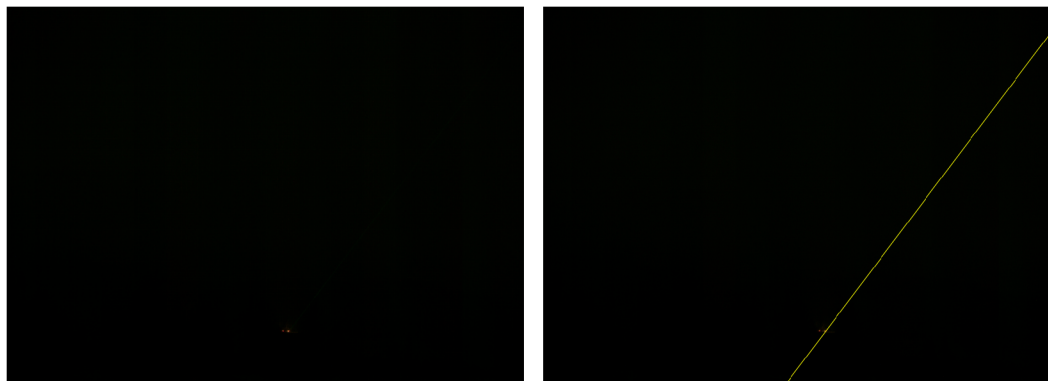
### 6.3 Vyhodnotenie

Výsledkom tejto práce je aplikácia, ktorá sa zameriava na detekciu zeleného laserového lúča v otvorenom priestore a výpočet jeho trajektórie. Úspešnosť detekcie ovplyvňuje niekoľko faktorov, ako napríklad množstvo šumu, veľkosť laserového lúča, homogenita prostredia a rozmer vstupnej snímky. Pri znižovaní obrazu tak klesá úspešnosť detekcie, keďže veľkosťou zastúpenia laserového lúča sa znižuje a dochádza k odstráneniu podstatných detailov.

Naopak, výhodou tohto programu je, že dokáže detekovať laserový lúč aj v tmavých snímkach, keď je laser málo viditeľný voľným okom, ako je, napríklad, ukázané na obrázku 6.4. Avšak úspešnosť detekcie klesá spolu so znižovaním expozičného času, pri ktorom bola snímka snímaná. Pri snímkach, zachytených s nízkym expozičným časom, sa tiež môže stať, že laserový lúč sa na fotke nenachádza a nie je tak možné ho zvýrazniť a detekovať.

Rýchlosť detekcie závisí najmä od rozmerov vstupného obrázka a kvadraticky klesá v závislosti od jeho veľkosti. Program je ale rýchlejší pri spracovaní tmavších snímok, keďže používa menej časovo náročné metódy.

Implementácia programu na detekciu zeleného laserového lúča v otvorenom prostredí je v zásade úspešná a dokáže efektívne a rýchlo zvýrazniť a zdetekovať tento lúč. Je však dôležité brať do úvahy faktory, ktoré môžu túto detekciu ovplyvniť, aby sa predišlo chybným a nežiadúcim výsledkom, ako je napríklad veľké množstvo šumu v obraze alebo množstvo ambientného osvetlenia.



Obr. 6.4: Zvýraznenie laserového lúča v tmavej snímke.

# Kapitola 7

## Záver

Cieľom tejto práce bolo spracovať obraz za účelom detekcie laserového lúča v otvorenom priestore. Z tohto dôvodu som sa najskôr zamerala na štúdium materiálov týkajúcich sa vlastností svetla a lasera ako aj jeho funkčnosťou a nebezpečnými účinkami. Následne som analyzovala rôzne metódy v rámci spracovania obrazu, za účelom odstránenia šumu a detekcií čiar. Vzhľadom na to, že lúč lasera je v obraze zobrazený ako čiara, zoznámila som sa predovšetkým s hranovými detektormi a algoritmami na detekciu priamok. Taktiež som našla metódy ako detekovať a lokalizovať zdroj lasera pomocou spracovania obrazu a dvoch snímok tej istej scény.

Aby sa získali potrebné dáta, uskutočnilo sa niekoľko rozsiahlych meraní, pri ktorých sa snímala laserový lúč v otvorenom priestore. Tieto merania boli realizované v spolupráci s Českým vysokým učením technickým v Prahe a Univerzitou obrany. Cieľom bolo vytvoriť rozsiahlu dátovú sadu, ktorá bude najvernejšie simulovať snímky oslnenia lietadiel pomocou laserového lúča. Pri meraniach boli použité rôzne kamerové systémy na to, aby sa porovnali ich vplyvy na kvalitu získavaných dát.

Vďaka nameraným dátam sa vytvoril prvotný návrh implementácie, využívajúci preštudované metódy. Tento návrh zahŕňal základné kroky spracovania obrazu, ako sú farebná korekcia, redukcia šumu, prahovanie, detekcia laserového lúča a výpočet výslednej trajektórie. Počas vývoja sa tieto kroky nemenili, avšak použité metódy v jednotlivých krokoch boli postupne modifikované za účelom zabezpečenia presnejšej detekcie a akcelerácie výpočtov.

V priebehu vývoja sa venovala pozornosť najmä optimalizácii programu. Jednotlivé metódy sa postupne menili a vo výslednej implementácii sa prispôbujú vstupným hodnotám snímky, keďže na každú snímku s rôznymi hodnotami je potrebné aplikovať odlišný postup, aby sa zaručila úspešná detekcia výsledného laserového lúča. Jednou z upravených častí bola tiež Houghova transformácia, pri ktorej sa modifikovala všeobecná funkcia `cv::HoughLinesP()` z knižnice OpenCV a vytvorila sa špecifickejšia verzia pre rýchlejšiu detekciu laserového lúča. Na zabezpečenie rýchlejšieho výpočtu bola vyskúšaná aj paralelizácia systému, konkrétne bola paralelizovaná najmä detekcia laserového lúča pomocou Houghovej transformácie. Avšak, vzhľadom na zvýšenú náročnosť na správu vlákien sa vo výslednej implementácii nepoužila.

Po dokončení implementácie programu bolo prevedené testovanie pomocou nameranej dátovej sady na rôznych počítačových zostavách. V rámci testovania bolo porovnaných niekoľko verzií programu, pričom sa skúmali rôzne aspekty, vrátane vplyvu optimalizovanej Houghovej transformácie na rýchlosť programu. Z testovania vyplynulo, že vďaka modifikácii Houghovej transformácie sa program zrýchlil o 4%. Rovnako bola zistená silná závislosť rýchlosti programu na veľkosti vstupnej snímky, pričom zmenšenie snímky viedlo k jeho

kvadratickému zrýchleniu. Pri porovnaní pôvodného návrhu s výslednou implementáciou sa podarilo program zrýchliť o 40%.

Spôľahlivosť detekcie laserového lúča závisí najmä od kvality vstupnej snímky, pričom úspešnosť klesá s rastúcim množstvom šumu a s klesajúcimi rozmermi obrazu. Napriek tomu sa podarilo vytvoriť efektívny algoritmus, ktorý dokáže detekovať a vypočítať trajektóriu laserového lúča v otvorenom priestore, aj v tmavých snímkach s málo viditeľným lúčom.

Alternatívou by mohlo byť spracovanie obrazu snímaného kamerou so zeleným filtrom, vďaka čomu by sa potlačilo prichádzajúce okolité svetlo ostatných vlnových dĺžok, ktoré sú odlišné od zeleného laserového lúča. Ďalšou možnosťou, ako zminimalizovať vplyv ambientného osvetlenia, je rozdeliť vstupný obraz na viac častí, ktoré by boli spracovávané paralelne. Laserový lúč by sa tak hľadal vo všetkých častiach súčasne a snímka by nebola ovplyvnená nerovnomerným osvetlením okolia. Tento spôsob by tiež mohol zrýchliť spracovanie obrazu a detekciu lúča. Akcelerovať program by sa tiež dalo použitím GPU pre spracovanie obrazových dát namiesto CPU, z dôvodu väčšej možnosti paralelizácie. Avšak rýchlosť môže byť ovplyvnená prenosom dát na a z GPU.



# Literatúra

- [1] Google Maps [cit. 2022-03-22]. Dostupné z: <https://www.google.com/maps>.
- [2] *Edge Detection: Zero X Laplacian*. [cit. 2022-05-01]. Dostupné z: [https://mipav.cit.nih.gov/pubwiki/index.php/Edge\\_Detection:\\_Zero\\_X\\_Laplacian](https://mipav.cit.nih.gov/pubwiki/index.php/Edge_Detection:_Zero_X_Laplacian).
- [3] *Lines Detection with Hough Transform*. Towards Data Science [cit. 2022-05-01]. Dostupné z: <https://towardsdatascience.com/lines-detection-with-hough-transform-84020b3b1549>.
- [4] *OSU Laser Safety Manual*. Oregon State University [cit. 2022-12-28]. Dostupné z: [https://ehs.oregonstate.edu/sites/ehs.oregonstate.edu/files/pdf/laser\\_safety\\_manual.pdf](https://ehs.oregonstate.edu/sites/ehs.oregonstate.edu/files/pdf/laser_safety_manual.pdf).
- [5] BIZJAK, G. *Light sources*. Laboratory of Lighting and Photometry, Faculty of Electrical Engineering, University of Ljubljana [cit. 2022-12-29]. Dostupné z: [http://lrf.fe.uni-lj.si/e\\_sv\\_tehnika/EN/i\\_LightSources.pdf](http://lrf.fe.uni-lj.si/e_sv_tehnika/EN/i_LightSources.pdf).
- [6] BRADSKI, G. a KAEHLER, A. *Learning OpenCV*. 1. vyd. O'Reilly Media, Inc., 2008. ISBN 978-0-596-51613-0.
- [7] BURGER, W. a BURGE, M. J. *Digital Image Processing*. 2. vyd. Springer London, 2016. ISBN 978-1-4471-6685-6.
- [8] COLIBAN, R.-M., MARINCAȘ, M., HATFALUDI, C. a IVANOVICI, M. Linear and Non-Linear Models for Remotely-Sensed Hyperspectral Image Visualization. *Remote Sensing*. August 2020.
- [9] *The Electromagnetic Spectrum* [online]. 2016 [cit. 2022-01-22]. Dostupné z: [https://www.miniphysics.com/electromagnetic-spectrum\\_25.html](https://www.miniphysics.com/electromagnetic-spectrum_25.html).
- [10] FEYNMAN, R. P. *Feynmanovy přednášky z fyziky s řešenými příklady. 1/3*. 1. vyd. Havlíčkův Brod: Fragment, 2000. ISBN 80-7200-405-0.
- [11] GEVERS, T., GIJSENIJ, A., WEIJER, J. van de a GEUSEBROEK, J.-M. *Color in Computer Vision: Fundamentals and Applications*. Somerset: Wiley, 2012. ISBN 9781118350089.
- [12] GRAHAM SMITH, F. *Optics and photonics : an introduction*. Chichester: John Wiley & Sons, 2000. ISBN 0-471-48925-5.
- [13] HALLIDAY, D. *Fyzika : vysokoškolská učebnice obecné fyziky. Část 4, Elektromagnetické vlny-Optika-Relativita*. Vyd. 1. V Brně : Praha: VUTIUM ; Prometheus, 2000. Překlady vysokoškolských učebnic. ISBN 80-214-1868-0.

- [14] HASLER, D. a SUSSTRUNK, S. Measuring colourfulness in natural images.
- [15] HLAVÁŠ, V. a ŠONKA, M. *Počítačové vidění*. 1. vyd. Grada a.s., 1992. ISBN 80-85424-67-3.
- [16] KARRI, S. T. *Hough Transform* [online]. Medium, 2019 [cit. 2022-11-13]. Dostupné z: <https://medium.com/@st1739/hough-transform-287b2dac0c70>.
- [17] HÁJOVSKÝ, R., PUSTKOVÁ, R. a KUTÁLEK, F. *Zpracování obrazu v měřicí a řídicí technice*. 1. vyd. Vysoká škola báňská – Technická univerzita Ostrava, 2012. ISBN 978-80-248-2596-0.
- [18] *Lesson Explainer: Coherent Light* [online]. [cit. 2022-4-22]. Dostupné z: <https://www.nagwa.com/en/explainers/367175916245/>.
- [19] LI, P., WANG, H., YU, M. a LI, Y. Overview of Image Smoothing Algorithms. *Journal of Physics: Conference Series*. IOP Publishing. apr 2021, zv. 1883, č. 1, s. 012024.
- [20] JUNG, T., MATTSSON, M.-O. a BRIDGES, J. Light Sensitivity. 2008.
- [21] MOHAN, V. A Performance Comparison of Edge Detection Techniques for Printed and Handwritten Document Images. Máj 2016.
- [22] PARKER, J. R. *Algorithm for image processing and computer vision*. 1. vyd. Wiley computer publishing, 1997. ISBN 0-471-14056-2.
- [23] PAVLOVIČOVÁ, J. *Číslíkové spracovanie obrazu*. 1. vyd. FELIA s. r. o, 2016. ISBN 80-7302-0007-6.
- [24] PUROHIT, G. Overview of Lasers. September 2020, s. 193–203.
- [25] RAMAN, M. a AGGARWAL, H. Study and Comparison of Various Image Edge Detection Techniques. *International Journal of Image Processing*. Marec 2009, zv. 3.
- [26] SAIF, J. A. M., HAMMAD, M. a ALQUBATI, I. Gradient Based Image Edge Detection. *International Journal of Engineering and Technology*. Marec 2016, s. 153–156.
- [27] SALEH, B. E. A. *Základy fotoniky = fundamentals of photonics. svazek 3*. Vyd. 1. Praha: Matfyzpress, 1995. ISBN 80-85863-05-7.
- [28] SHARMA, K., KIM, S. a SINGH, M. An improved feature matching technique for stereo vision applications with the use of self-organizing map. *International Journal of Precision Engineering and Manufacturing*. August 2012, zv. 13.
- [29] SIEGMAN, A. E. *LASERS*. 1. vyd. Mill Valley, California: University Science Book, 1986. ISBN 0-935702-11-03.
- [30] VOZIKIS, G. a JANSÁ, J. Advantages and Disadvantages of the Hough Transformation in the Frame of Automated Building Extraction. Január 2008.

# Príloha A

## Počítačová zostava pre testovanie

### A.1 Zostava 1

**CPU:** 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz

**RAM:** 16GB DDR4 2133 MHz

**HDD:** 1TB SSD

**OS:** Ubuntu 22.04.2 LTS

### A.2 Zostava 2

**CPU:** AMD Ryzen 5 2400G with Radeon Vega Graphics

**RAM:** 16GB DIMM DDR4 2666MHz

**HDD:** 500GB SSD

**OS:** Ubuntu 22.04.6 LTS

### A.3 Zostava 3

**CPU:** Intel(R) Celeron(R) 6305E @ 1.80GHz

**RAM:** 8GB DDR4 2400 MHz

**HDD:** 128GB SSD

**OS:** Ubuntu 20.04.6 LTS

### A.4 Zostava 4

**CPU:** Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz

**RAM:** 16GiB DIMM DDR4 3200 MHz

**HDD:** 1TB HDD

**OS:** Ubuntu 22.04.1 LTS

## Príloha B

# Obsah pamäťového média

- `app/` - zdrojové súbory aplikácie s návodom k spusteniu
- `doc/` - zdrojové súbory technickej správy
- `data/` - testovacia sada nameraných snímok
- `xsokov01.pdf` - technická správa