



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

# NÁVRH A REALIZACE METODY VYHÝBÁNÍ SE PŘEKÁŽKÁM V OUTDOOR PROSTŘEDÍ PRO MOBILNÍ ROBOT

DESIGN AND IMPLEMENTATION OF AN OBSTACLE AVOIDANCE METHOD IN AN OUTDOOR  
ENVIRONMENT FOR A MOBILE ROBOT

## DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

**Bc. Tomáš Fargač**

### VEDOUCÍ PRÁCE

SUPERVISOR

**doc. Ing. Stanislav Věchet, Ph.D.**

**BRNO 2024**

# Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	<b>Bc. Tomáš Fargač</b>
Studijní program:	Mechatronika
Studijní obor:	bez specializace
Vedoucí práce:	<b>doc. Ing. Stanislav Věchet, Ph.D.</b>
Akademický rok:	2023/24

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

## **Návrh a realizace metody vyhýbání se překážkám v outdoor prostředí pro mobilní robot**

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

Práce se zabývá návrhem a implementací metody určené pro navigaci ve volném venkovním prostředí mobilních robotů. Navigace v otevřených prostorách které nejsou detailně zmapovány a obsahují množství nekategorizovaných objektů zvyšuje nároky na navigační metody užívané mobilními roboty. Cílem práce je implementovat dílčí metodu zaměřenou na vyhýbání se resp. objíždění překážek při pohybu v přímém směru.

### **Cíle diplomové práce:**

Specifikujte prostředí ve kterém se daný mobilní robot pohybuje.

Navrhněte vhodnou metodu vyhýbání se překážkám.

Vybranou metodu implementujte a otestujte.

### **Seznam doporučené literatury:**

SOLEM, J. E., Programming Computer Vision with Python: Tools And Algorithms For Analyzing Images, 1st edition, 2012.

LIGHT, R. A., Mosquitto: server and client implementation of the MQTT protocol, The Journal of Open Source Software, vol. 2, no. 13, May 2017, DOI: 10.21105/joss.00265.

BALÁTĚ, J.: Technické prostředky automatického řízení. Praha, SNTL 1986.

ROS.org. ROS.org | Powering the world's robots. [online]. 2.11.2016 [cit. 2016-11-02]. Dostupné z: <http://www.ros.org/>.

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

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

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

## Abstrakt

Tato práce se věnuje zkoumání využitelnosti metody optického toku při zpracování obrazu. Nejprve je tato metoda představena teoreticky a poté je matematicky odvozen její základ. Následně je představena myšlenka její implementace do rozhodovacích algoritmů a potenciálních oblastí využití. V práci je také možno nalézt elaboraci nad vhodným prostředím pro takovou aplikaci jak ve virtuálním tak i reálném světě. Praktická část pak ukazuje vývojový proces krok za krokem a také zdokonalování práce s touto metodou a jejími výstupy. Práce využívá programovacího prostředí Matlabu a detailní práci na úrovni jednotlivých komponentů v tomto programovacím jazyce, obohaceného o pomocné balíčky, zejména ze světa počítačového vidění. Celé bádání je nakonec přehledně shrnuto a všechny podniknuté kroky jsou vyobrazeny ve vývojovém diagramu. Závěrem jsou jasně prezentovány všechny zkoumané přístupy s jejich silnými stránkami a nedostatky, jež byly v průběhu identifikovány.

## Summary

This thesis focuses on investigating the usability of the optical flow method in image processing. Firstly, this method is introduced theoretically, followed by its mathematical derivation. Subsequently, the idea of implementing it into decision-making algorithms and potential areas of application is presented. The thesis also elaborates on suitable environments for such applications in both virtual and real worlds. The practical part demonstrates the step-by-step development process and the refinement of working with this method and its outputs. The work utilizes the Matlab programming environment and detailed work at the level of individual components in this programming language, enriched with auxiliary toolboxes, especially from the field of computer vision. The entire research is summarized clearly at the end, and all undertaken steps are depicted in a flowchart. Finally, all explored approaches with their strengths and weaknesses, identified throughout the process, are clearly presented.

## Klíčová slova

Počítačové vidění, optický tok, zpracování obrazu, navigace, robotika v terénu

## Keywords

Computer vision, optical flow, image processing, navigation, field robotics

## Bibliografická Citace

FARGAČ, T. *Návrh a realizace metody vyhýbání se překážkám v outdoor prostředí pro mobilní robot*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2024. 73 s., Vedoucí diplomové práce: doc. Ing. Stanislav Věchet, PhD.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně za pomoci rad a doporučení mého vedoucího práce a s použitím odborné literatury a pramenů uvedených v seznamu použité literatury.

**Tomáš Fargač**

Brno . . . . .

. . . . .

## Poděkování

Hlavní poděkování patří mému vedoucímu práce panu doc. Ing. Stanislavu Věchetovi, PhD. za jeho odborné rady a příkladné vedení v době realizace této diplomové práce a také mé rodině, jenž mi byla vždy oporou po celou dobu mého studia.

**Tomáš Fargač**

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Navigace ve venkovním prostředí</b>	<b>10</b>
2.1	State of art . . . . .	11
2.1.1	GPS . . . . .	11
2.1.2	Lidar . . . . .	12
2.2	Testovací prostředí . . . . .	12
2.2.1	Trackmania . . . . .	13
2.2.2	Editor . . . . .	13
<b>3</b>	<b>Optický tok</b>	<b>14</b>
3.1	Popis metody . . . . .	14
3.2	Odvození . . . . .	14
3.2.1	Geometrická interpretace . . . . .	16
3.2.2	Štěrbínový problém . . . . .	18
3.3	Lucas Kanade . . . . .	19
3.4	Praktické aplikace optického toku . . . . .	21
<b>4</b>	<b>Heterogenní rozhodovací systémy</b>	<b>22</b>
4.1	Rozhodovací metody . . . . .	22
4.2	Optický tok v rozhodovacím systému . . . . .	23
4.2.1	Návrh rozhodovacího modelu . . . . .	24
<b>5</b>	<b>Matlab a optický tok</b>	<b>26</b>
5.1	Computer vision toolbox . . . . .	26
5.2	Výpočet optického toku . . . . .	26
5.2.1	Kartézský výstup . . . . .	28
5.2.2	Polární výstup . . . . .	28
5.3	Pilot a šipky . . . . .	29
5.3.1	Ideální virtuální prostředí . . . . .	31
5.4	Problém s rotací . . . . .	32
<b>6</b>	<b>Vidění robota</b>	<b>34</b>
6.1	Izolace šipek . . . . .	34
6.2	Binární využití magnitudy . . . . .	35
6.2.1	Histogram videa . . . . .	41

<b>7</b>	<b>Hloubka obrazu</b>	<b>44</b>
7.1	Potenciálová mapa . . . . .	44
7.1.1	Analýza potenciálových map . . . . .	45
7.2	RGB rozlišení . . . . .	49
7.2.1	Přínos RGB zpracování . . . . .	51
7.2.2	Efekt clustrování . . . . .	55
7.2.3	Vliv světla . . . . .	57
<b>8</b>	<b>Úhlová informace</b>	<b>59</b>
8.1	Definice hypotézy . . . . .	59
8.2	Implementace . . . . .	60
8.3	Ověření . . . . .	61
<b>9</b>	<b>Rekapitulace metod</b>	<b>64</b>
9.1	Evoluce práce . . . . .	64
9.2	Hodnocení metod . . . . .	65
<b>10</b>	<b>Závěr</b>	<b>67</b>
	<b>Bibliografie</b>	<b>68</b>
	<b>Seznam obrázků</b>	<b>71</b>
	<b>Přílohy</b>	<b>73</b>



# 1 Úvod

V posledních letech je patrný trend postupné implementace robotických systémů s různým stupněm autonomie do většiny oblastí lidské činnosti. Trend lze pozorovat od domácností, přes výrobní haly, dokonce až po profese konzervativnějšího charakteru, kdy už dnes ani moderní farmy nejsou výjimkou.

Autonomní navigace mobilních robotů, ať už se jedná o konvenční AGV (*autonomous guided vehicles*) nebo průmyslových či civilních vozidel patří k aktuálně řešené problematice, která přináší možnosti samostatného provozu či snadnější administrace flotily pracovních vozidel. Zejména v oblasti moderních farem se jedná o slibný trend do budoucna, vzhledem k predikovanému nedostatku pracovních sil v tomto sektoru a neustále se zvyšujícím nárokům na produkci potravin.

Optický tok je metoda zpracování obrazu z oblasti počítačového vidění [1], která využívá relativních pohybů mezi prostředím a pozorovatelem, díky čemuž může poskytnout informaci o prostorovém uspořádání objektů v zorném poli pozorovatele. Od standardních metod na bázi senzorů se optický tok liší tím, že využívá vizuální informace o pohybu v obraze [2].

Hlavními body této práce je specifikace prostředí, ve kterém se mobilní robot může vyskytovat. Výběr a popis vhodné metody vyhýbání se překážkám a také návrh možné implementace dané metody.

Práce je členěna do několika kapitol, kdy na úvod je provedena rešerše prostředí a metod. Poté je představena a vysvětlena teorie stojící za zvolenou metodou Optického toku. Druhá a hlavní část práce je věnována konkrétnímu praktickému zpracování problematiky navigace ve venkovním prostředí pro aplikaci v oblasti tzv. *Field robotics*. Praktická část práce detailně popisuje vývoj navigačního modulu využívající metodu optického toku na zpřesnění povědomí robotu o blízkém okolí.

Dosažené výsledky jsou shrnuty do přehledné tabulky prozkoumaných sub-metod spolu s identifikovanými přednostmi a slabinami pro každou z nich. Závěrem je shrnutí práce s důrazem na dosažené výsledky v kontextu vytyčených cílů.

## 2 Navigace ve venkovním prostředí

Navigace ve venkovním prostředí se zaměřením na oblast robotiky se zabývá vývojem algoritmů a technologií s cílem zautomatizovat, zjednodušit a vdechnout jistý stupeň autonomie takovým aplikacím. Jedná se o aplikace, které umožňují robotům pohybovat a orientovat se v neznámém a nespecifickém terénu. Klíčové pojmy pro celou tuto oblast jsou následující:

- Lokalizace a mapování

Toto je proces, při kterém robot současně vytváří mapu okolí a lokalizuje sám sebe a také objekty v tomto prostředí

- Plánování trasy

Rozhodování o nejlepší cestě k dosažení globálního cíle za využití lokálně získaných informací umožňující okamžitou reaktivitu a adaptaci na nenadálé události.

- Detekce překážek

Velká kapitola sama o sobě, která zahrnuje několikero metod jak detekovat překážky různého typu v prostředí, ve kterém se robot nachází. Hlavním cílem je předejít kolizím, jež by ohrozily splnění globálního cíle a nebo by mohly ohrozit jak fungování robota, tak jeho integritu.

- Senzorika

V návaznosti na předchozí bod, je důležitá oblast týkající se interakce robota a jeho okolí. Typické směřování informačního toku je od okolí k robotovi, kdy na jeho rozhraní musí docházet ke zpracování zachycených vjemů a signálů přicházejícího z okolí. Možností je několik, jedna typická je LiDAR, o které se pojednává v kapitole 2.1.2, ale také další metody jako například ultrazvukové senzory nebo kamery.

- Optimalizace cesty

Disciplína zabývající se vyhledáváním nejefektivnější cesty s ohledem na různé faktory jako jsou překážky (přesah předchozího bodu *Detekce překážek*), terén, rychlost nebo taky energetická efektivita.

- Globální navigační systémy

Navigace a automatizace může využívat globální systémy, jako nejznámější zástupce GPS, o které je pojednáváno v kapitole 2.1.1. Nicméně se může jednat i o další systémy jako GLONASS nebo Galileo.

- Využití umělé inteligence

A v neposlední řadě toto prostředí nabízí široké spektrum možností využít i metod umělé inteligence, jež zažívají rapidní vzestup, a to zejména posledních pár let. Velice oblíbené je jejich užití pro detekci překážek, klasifikaci nebo plánování trasy.

## 2.1 State of art

Polní robotika neboli *Field Robotics* se zabývá automatizací provozování vozidel a platform v drsném, nestrukturovaném prostředí. Zahrnuje automatizaci mnoha pozemních, námořních a vzdušných platform v aplikacích, jako je těžba, manipulace s nákladem, zemědělství [3], podvodní průzkum, dálnice, průzkum planet, dohled nad pobřežím a mnoho dalších.

Polní robotika se vyznačuje aplikací nejpokročilejších principů robotiky ve snímání a ovládání v nestrukturovaných a nelítostných prostředích. Přitažlivost terénní robotiky spočívá v tom, že je náročná na vědu, zahrnuje nejnovější inženýrství a principy navrhování systémů a nabízí skutečnou výzvu robotiky s možností přispět podstatným způsobem k hospodářskému a sociálnímu vývoji [4].

Polní roboti jsou obecně mobilní platformy, které často pracují venku a silně interagují se svým prostředím, bez dozoru člověka. Klasickým příkladem jsou bagry, které automaticky nakládají kamiony, popř. automatizované buldozery, které čistí pole. Některé příklady úplně nesplňují všechna kritéria, ale stále jsou považovány za polní roboty, jako například automatizovaná hlubinná těžba, tyto stroje technicky nepracují venku ale stále sdílejí většinu technických problémů společných pro polní roboty [5]. Dále například robotické jeřáby také nejsou nutně mobilní v obvyklém smyslu, ale přesto musí vnímat překážky, vypořádat se s rozměry prostředí, proměnlivým osvětlením a poryvy větru.

### 2.1.1 GPS

GPS (*Global positioning system*) je globální navigační systém, který umožňuje určit polohu, rychlost a čas kdekoli na planetě Zemi, za předpokladu, že je k dispozici dostatečný počet satelitů GPS [6]. Každý GPS satelit vysílá signály obsahující informace o své poloze a čase. GPS přijímače na Zemi tyto signály zachytávají z několika satelitů zároveň a na základě časových rozdílů mezi signály určují svoji polohu pomocí techniky trilaterace [7]. Signál GPS se nejčastěji používá po navigaci, mapování, geocaching, sledování vozidel a osob atd.

Pro tuto práci je relevantní například publikace kombinující GPS a autonomní řízení auta od Rahimana a Zainala *An Overview of Development GPS Navigation for Autonomous Car*, kdy cílem projektu bylo vytvořit navigační model auta, který dokáže projet známými popř. předprogramovanými souřadnicemi autonomně bez jakékoli kontroly člověkem [8].

GPS navigace poskytuje následující seznam výhod: je globálně dostupná a může být používána kdekoli na Zemi (v případě dostatečně silného pokrytí signálem), má relativně vysokou přesnost (řádově jednotky metrů) a má širokou škálu využití od dopravy, vojenství, přes sport až po zemědělství.

Nicméně skýtá i některé nedostatky jako je například omezená spolehlivost v určitých podmínkách (signál může být blokován fyzickými strukturami a nebo elektromagneticky rušen ať už chtěně nebo nechtěně), což může vést ke ztrátě přesnosti a nebo k úplnému výpadku. Kvůli odrazům může docházet ke zpoždění a výpadkům signálu a jako každý systém může být ohrožen technickými poruchami a výpadky.

Obecně je GPS velmi silný a spolehlivý nástroj, pokud si je člověk vědom jeho limitací a umí jim vědomě předcházet a nebo alespoň minimalizovat problémy spojené s jejich výpadkem.

### 2.1.2 Lidar

LIDAR originálně z angličtiny *Light Detection and Ranging* je technologie využívající laserové paprsky k měření vzdálenosti a vytváření trojrozměrných map okolí [9]. Funguje na principu vysílání laserových paprsků směrem k povrchu a měření času do návratu vyzářeného paprsku zpět k senzoru. Na základě času, který trvá, než se odražený signál vrátí k senzoru, určuje LIDAR vzdálenost mezi povrchem a senzorem. Tato data mohou být následně využita k vytvoření trojrozměrných map, detekci překážek a nebo navigaci.

Možná aplikace LIDARů ve *Field robotics* je naznačena v práci od Malavaziho a spol. *LiDAR-only based navigation algorithm for an autonomous agricultural robot*. Účelem práce v tomto článku bylo vyvinout obecný přístup pro navigaci autonomních robotů pouze pomocí dat z LIDARU. Aby byla robotická navigace co nejrobustnější, byla práce zaměřena pouze na informace získané tímto způsobem. Nebylo použito žádné předchozí informace o poli, jako je například velikost a šířka řádků [10].

Výhody této metody mohou být shrnuty jako přesnost a flexibilita. LIDAR poskytuje velmi přesné měření vzdáleností na lokální úrovni a velmi detailní mapování prostoru. Moderní LIDARY mají také široké zorné pole, což umožňuje rychlé a efektivní skenování okolí. V neposlední řadě je výhodou nezávislost na světelných podmínkách, kdy tato technologie může fungovat jak ve dne tak i v noci.

Cena za tyto benefity je reflektována v nákladech na pořízení, kdy cenovka je leckdy překážkou v jejich masovém nasazení. Nezávislost na osvětlení je výhodou, nicméně další venkovní faktory, jako silný déšť, mlha nebo sněžení mohou potenciálně snižovat přesnost. Dalším omezením může být problém s detekcí transparentních materiálů, jako jsou vodní hladiny nebo sklo [11].

Opět lze tvrdit, že LIDAR je velmi solidní nástroj pro specifické aplikace, jako je mapování neznámého terénu a také může být dobrý pomocník při autonomní navigaci ve venkovním prostředí. Podobně jako u GPS, pokud si je člověk vědom možných úskalí a limitů této technologie, je možné najít aplikace, které mohou benefitovat z implementace této technologie.

Samozřejmě, žádná technologie není neomylná, natož vše spásná. To je důvodem vrstvení různých přístupů, tak aby se nespolehalo pouze na jeden zdroj informací. Další potenciální možnosti vhodných přímo do těchto podmínek autonomní navigace v nespécifikovaném a různorodém venkovním prostředí může být metoda analýzy a zpracování Optického toku, již se detailně věnuje kapitola 3.

## 2.2 Testovací prostředí

Předchozí část popisuje překážky, složitosti a možnosti navigace ve venkovním prostředí. Z důvodu komplexnosti reálného prostředí ve smyslu sběru dat nebo například izolace jednotlivých faktorů je pro vývoj vhodné zařazení mezikroku a využití testovacího prostředí.

Tato práce ve své praktické části využívá virtuálního prostředí a možností počítačové hry *Trackmania*. Svými širokými možnostmi (detailně v následující kapitole 2.2.1) poskytla ideální mezikrok a platformu pro testování jednotlivých iterací.

### 2.2.1 Trackmania

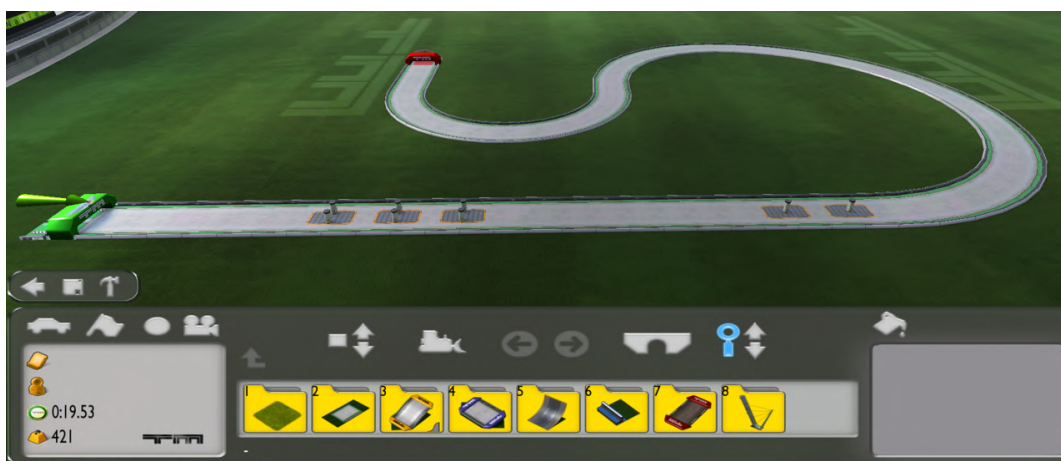
TrackMania Nations Forever (TMNF nebo TMF) je 3D závodní hra, která byla vydána v roce 2008 společností Nadeo. Je součástí závodní herní série TrackMania a byla navržena pro *Electronic Sports World Cup*, což je každoroční mezinárodní mistrovství profesionálních her. V Trackmanii je cílem hráče dokončit trať co nejrychleji. Hráč musí nejprve projet všemi kontrolními body, což může provést v libovolném pořadí, a poté se dostat k cílové bráně [12].

Výběr Trackmanie pro účel testování přináší hned několik výhod:

- Ve hře je hlavním cílem řízení formule, jež může být také považována za mobilního robota.
- Perspektiva hry/kamery příhodně simuluje pozici a možnosti snímání prostředí jako v případě reálného venkovního robota.
- Trackmanie je často používaným nástrojem pro podobné simulace a vývoje AI aplikací. Velmi oblíbeným předmětem zájmu jsou například metody *Reinforcement learningu* [13] a [14].
- Čtvrtým a největším benefitem této volby je editor (kapitola 2.2.2), který poskytuje široké možnosti testování konceptů, izolování jednotlivých parametrů, jako je například světelnost, typy překážek, ... a v neposlední řadě urychluje sběr dat pro následnou analýzu.

### 2.2.2 Editor

Editor map v Trackmanii je verzatilní nástroj, který dává možnost vytvořit trať plně dle vlastních potřeb. Trať se skládá ze sady 3D bloků, které jsou umístěny ve 3D prostoru. Na počátku se závodní formule objeví na startovním bloku. K dokončení trati musí hráčovo auto projít všemi bloky kontrolních bodů a poté dosáhnout cílového bloku [12].



Obrázek 2.1: Editorový mód se základní trati

Kromě startovních, kontrolních a cílových bloků existuje mnoho dalších silničních bloků a překážek, jako rovné silnice, zatáčky, svahy, přepážky, kužely, skoky, díry atd. Obrázek 2.1 ukazuje trať v editorovém módu se základními bloky a překážkami. Později je tohoto editoru využito pro vývoj a testování jednotlivých iterací algoritmu.

## 3 Optický tok

Tato kapitola navazuje na otevřenou možnost využití Optického toku vedle metod GPS a LIDARU z konce kapitoly 2.1.2 a detailně popisuje její teoretický základ. V kapitole je odvozena matematická páteř metody a také ukázány praktické výsledky a nedostatky tohoto přístupu, při aplikaci v reálném světě.

Dvourozměrný pohyb obrazu je projekce trojrozměrného pohybu objektů vzhledem k vizuálnímu senzoru (pozorovateli) na jeho obrazovou rovinu. Analýza v čase takových sekvencí obrazu umožňuje odhad okamžité rychlosti pohybu v pozorovaném dvourozměrném obrazu. Toto je obvykle nazváno jako *Optický tok* [15].

Za předpokladu, že Optický tok je spolehlivou aproximací pohybu dvourozměrného obrazu, lze jej pak použít také například k vizualizaci trojrozměrného pohybu a nebo trojrozměrné struktury povrchu. Optický tok může být také použit k provádění detekce pohybu, segmentace objektů nebo také výpočtu doby do kolize (*TTC - Time to contact*, jako například v [16]).

### 3.1 Popis metody

Optický tok je popis zjevného pohybu objektů, povrchů a hran ve vizuální scéně způsobený relativním pohybem mezi pozorovatelem a scénou [17].

Koncept optického toku zavedl americký psycholog James J. Gibson ve 40. letech 20. století, aby popsal vizuální podněty poskytované zvířatům pohybujícím se světem [18]. Gibson zdůraznil důležitost optického toku pro vnímání afordance, schopnost rozeznat možnosti jednání v prostředí. Stoupenci Gibsona dále prokázali roli optického proudového stimulu pro vnímání pohybu pozorovatelem ve světě: vnímání tvaru, vzdálenosti a pohybu předmětů ve světě; a řízení pohybu [19].

Termín optický tok je také používán robotiky, což je klíčová sféra aplikace pro tuto práci, a zahrnuje související techniky ze zpracování obrazu, řízení a navigace, jako je již zmíněná detekce pohybu, segmentace objektu, informace o čase do kontaktu, zaměření expanzních výpočtů, jas a nebo pohybově kompenzované kódování [20].

### 3.2 Odvození

Při odvozování této metody je nutno brát v potaz několik základních předpokladů, na kterých tyto výpočty stojí:

1. Posun sledovaného pixelu o souřadnicích  $(x, y)$  o krok  $(\delta x, \delta y)$  mezi jednotlivými za sebou jdoucími snímky je malý.
2. Časový krok  $(\delta t)$  mezi jednotlivými za sebou jdoucími snímky je malý.
3. Intenzita pixelu o souřadnicích  $x, y$  v čase  $t$ , značená jako:  $I(x, y, t)$  je mezi jednotlivými za sebou jdoucími snímky pro sledovaný pixel konstantní, tedy:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (3.1)$$

Dále pro odvození je potřeba připomenout Taylorův rozvoj, kdy:

$$f(x + \delta x) = f(x) + \frac{\delta f}{\delta x} \delta x + \frac{\delta^2 f}{\delta x^2} \frac{\delta x^2}{2!} + \dots + \frac{\delta^n f}{\delta x^n} \frac{\delta x^n}{n!} \quad (3.2)$$

A z předpokladů výše, kdy  $\delta x$  je malá, dojde k lineární aproximaci rovnice 3.2 na:

$$f(x + \delta x) = f(x) + \frac{\delta f}{\delta x} \delta x \quad (3.3)$$

Na základě této aproximace z rovnice 3.3 se dá zapsat pravá strana rovnice 3.1 jako:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\delta I}{\delta x} \delta x + \frac{\delta I}{\delta y} \delta y + \frac{\delta I}{\delta t} \delta t \quad (3.4)$$

A zjednodušit notaci zápisu jako:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + I_x \delta x + I_y \delta y + I_t \delta t \quad (3.5)$$

Poté od sebe odečteme rovnice 3.1 a 3.5 a dostaneme následující podmínku:

$$I_x \delta x + I_y \delta y + I_t \delta t = 0 \quad (3.6)$$

Tato rovnice se podělí výrazem  $\delta t$  na:

$$I_x \frac{\delta x}{\delta t} + I_y \frac{\delta y}{\delta t} + I_t = 0 \quad (3.7)$$

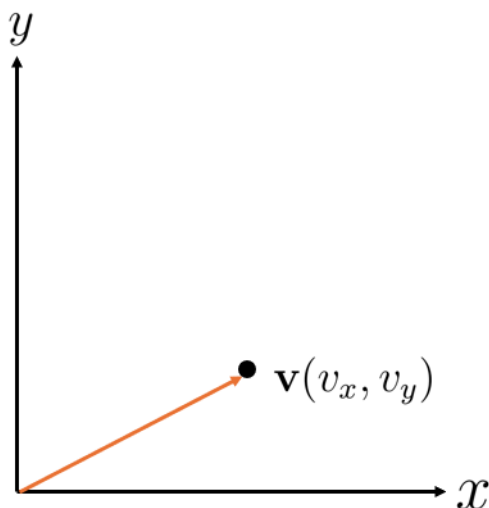
Odkud po úpravě vznikne výsledná podmínka:

$$I_x v_x + I_y v_y + I_t = 0 \quad (3.8)$$

Ze dvou po sobě jdoucích snímků lze numericky zjistit všechny tři derivace  $I_x, I_y, I_t$  a tudíž se stávají známými, to znamená, že rovnice 3.8 se stává rovnicí přímky s neznámými parametry  $v_x$  a  $v_y$ .

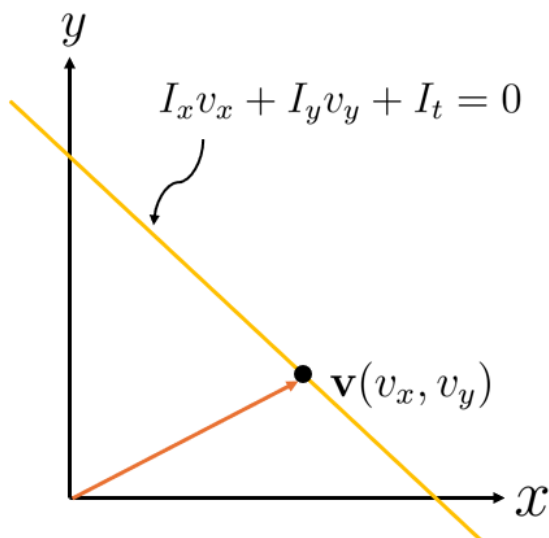
### 3.2.1 Geometrická interpretace

Mějme tedy dvou-dimenzionální prostor  $x, y$ , ve kterém leží hledaný bod z rovnice 3.8 o neznámých souřadnicích  $\mathbf{v}(v_x, v_y)$ .



Obrázek 3.1: Hledaný bod optického toku

Pro nalezení tohoto bodu je k dispozici rovnice přímky 3.8, kdy je známo, že hledaný bod leží někde na této přímce  $I_x v_x + I_y v_y + I_t = 0$ , jak je možno vidět na Obrázku 3.2.



Obrázek 3.2: Známá rovnice přímky

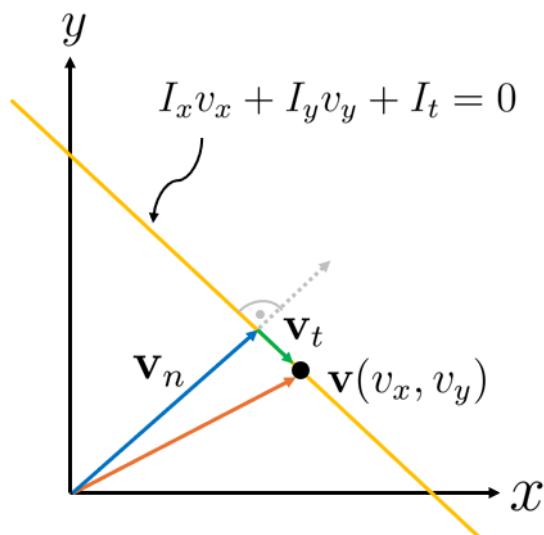
Poloha tohoto bodu na této přímce je ovšem neznámá a jelikož je tento problém definován jednou rovnicí o dvou neznámých, znamená to, že tato úloha je špatně podmíněná a v tomto stavu se nedá jednoznačně vyřešit.

Tento problém v identifikaci složek vektoru  $\mathbf{v}(v_x, v_y)$  lze dále redefinovat jako hledání



normálové  $n$  a tečné  $t$  složky nového vektoru  $\mathbf{v}(v_n, v_t)$  podle 3.9. Tato redefinice je na znázorněna níže na Obrázku 3.3.

$$\mathbf{v}_n + \mathbf{v}_t = \mathbf{v}(v_x, v_y) \quad (3.9)$$



Obrázek 3.3: Rozložení na normálovou a tečnou složku

Normálová složka k rovnici přímky od počátku souřadného systému se dá nalézt vždy. Je k tomu potřeba zjistit její směr a velikost. Směr jakožto jednotkový vektor kolmý ke žluté přímce z Obrázku 3.3 je:

$$\hat{\mathbf{v}}_n = \frac{(I_x, I_y)}{\sqrt{I_x^2 + I_y^2}} \quad (3.10)$$

A velikost jako:

$$|\mathbf{v}_n| = \frac{|I_t|}{\sqrt{I_x^2 + I_y^2}} \quad (3.11)$$

A z tohoto po vynásobení 3.10 a 3.11 lze vypočítat výslednou  $v_n$  jako:

$$\mathbf{v}_n = \frac{|I_t|}{I_x^2 + I_y^2} (I_x, I_t) \quad (3.12)$$

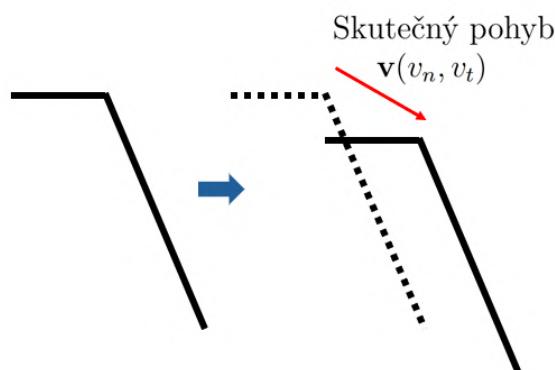
To znamená, že normálový optický tok, lze vypočítat vždy a to pouze za pomoci rovnice 3.8. Odvození prezentováno výše je na základě [21] a [22].

Problémem však zůstává tečná složka  $v_t$ , jelikož ta se může nacházet kdekoli na žluté čáře v Obrázku 3.3, což je jediná informace, která je o této komponentě známá.

### 3.2.2 Štěrbínový problém

Tento nedostatek z neznalosti tečné složky se netýká pouze matematického výpočtu, dokonce je možno demonstrovat jeho vliv i vizuálně. V anglické literatuře je znám pod jménem *Aperture problem* [15] a [22], nicméně v této práci česky jako *Štěrbínový problém*.

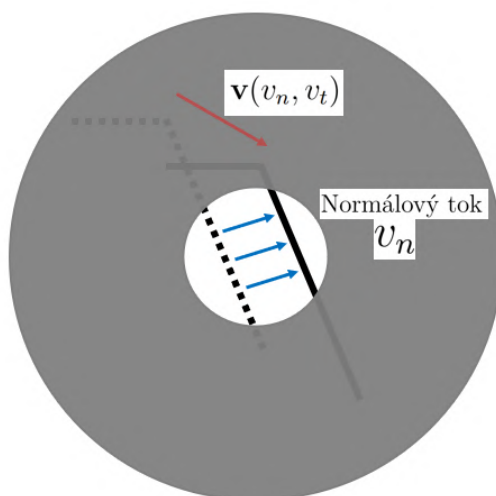
Mějme jednoduchý předmět zahnutého tvaru, který se pohybuje způsobem vyznačeným na Obrázku 3.4



Obrázek 3.4: Skutečný pohyb celého obrazce

V tomto případě lze správně pozorovat skutečný pohyb. Avšak tato ideální situace demonstruje pouze úkaz, kdy je pohyb v celém obrazu stejný pro všechny body. A tak tento předpoklad nelze aplikovat vždy a globálně na celý obraz, jelikož optický tok v celém zorném poli může být na různých místech odlišný. Z tohoto důvodu je potřeba omezit pozorování pouze na malé okolí, kdy už lze aplikovat předpoklad, že v tomto malém okolí je optický tok stejný pro všechny body.

A přesně s tímto předpokladem nastává *Štěrbínový problém* naznačený v Obrázku 3.5, kdy na malém okolí je možno pozorovat/vypočítat pouze normálový optický tok a nikoliv správně celý skutečný pohyb.



Obrázek 3.5: Zaměření na pohyb pouhé části obrazce

Takže jak bylo ukázáno výše, lokálně lze sice správně určit normálový tok při opomenutí tečného toku, což je označováno jako Štěrbinový problém, avšak situace je nyní tedy stále taková, že máme jednu klíčovou rovnici 3.8, nicméně v té jsou stále dvě neznámé. Tuto nevyváženost je možno vyřešit představením dalších podmínek.

### 3.3 Lucas Kanade

Metoda Lucas Kanade využívá již zmíněného předpokladu v kapitole 3.2.2, že optický tok v malém okolí bodu je stejný pro celé okolí. Tato podmínka je poté využita k vyřešení nedostatečně podmíněné úlohy z kapitoly 3.2.

Mějme tedy malou oblast v obraze o celkové velikosti v pixelech  $w(k,l)$ , kde je předpoklad optického toku konstantní pro všechny pixely uvnitř tohoto okolí  $w$ , tedy:

$$\forall(k, l) \in w : I_x(k, l)v_x + I_y(k, l)v_y + I_t(k, l) = 0 \quad (3.13)$$

A protože těchto bodů je v tomto okolí o velikosti  $w = nxn$  více, tak je výsledkem soustava rovnic. V plném zápisu:

$$\begin{bmatrix} I_x(1, 1) & I_y(1, 1) \\ I_x(k, l) & I_y(k, l) \\ \vdots & \vdots \\ I_x(n, n) & I_y(n, n) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_t \end{bmatrix} = \begin{bmatrix} I_t(1, 1) \\ I_t(k, l) \\ \vdots \\ I_t(n, n) \end{bmatrix} \quad (3.14)$$

Počet rovnic/řádků 3.14 bude rovný počtu pixelů v oblasti  $w$ . Tento způsob funguje za předpokladu, že alespoň některé z těchto rovnic budou díky odlišným derivacím ( $I_x, I_y$ ) lineárně nezávislé. A tyto derivace budou odlišné, právě tehdy, když analyzovaná oblast bude texturou obsahovat dostatečně odlišné pixely (ukázka na Obrázcích 3.6 a 3.7), odvození za pomoci [23].

Plný tvar výrazu 3.14 lze zapsat maticově jako:

$$\mathbf{A}\mathbf{u} = \mathbf{B} \quad (3.15)$$

Kde  $\mathbf{A}$  je známá matice o velikosti  $n^2 \times 2$ ,  $\mathbf{B}$  je známá matice o velikosti  $n^2 \times 1$  a  $\mathbf{u}$  je neznámý vektor o velikosti  $2 \times 1$ .

To znamená, že řešení  $n^2$  rovnic o 2 neznámých vede k řešení pomocí metody nejmenších čtverců [24]. Na základě rovnice 3.15 se provedou následující úpravy pro zjištění neznámého vektoru  $\mathbf{u}$ . Nejdříve vynásobení obou stran rovnic zleva transponovanou maticí  $A$ :

$$A^T \mathbf{A}\mathbf{u} = A^T \mathbf{B} \quad (3.16)$$

A po úpravě a osamostatnění vektoru  $\mathbf{u}$  na levé straně vznikne tvar:

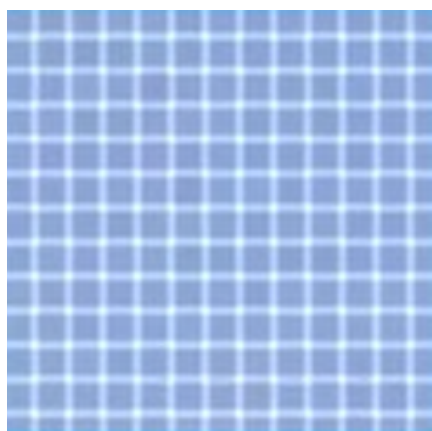
$$\mathbf{u} = (A^T A)^{-1} A^T \mathbf{B} \quad (3.17)$$

Aby tato metoda fungovala musí být splněny následující podmínky:

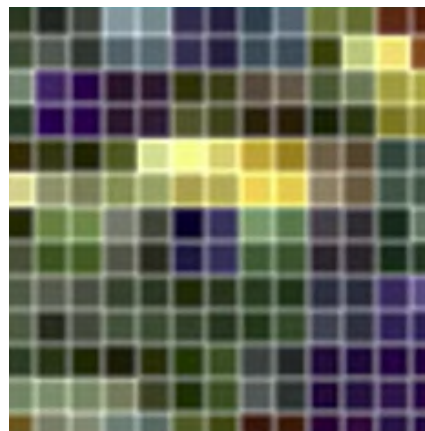
1. K matici  $(A^T A)$  musí existovat matice inverzní, nebo-li:  $\det(A^T A) \neq 0$
2. Matice  $(A^T A)$  musí být dobře podmíněná, to znamená, že v tomto případě vlastní čísla  $(\lambda_1, \lambda_2)$  této matice:
  - (a) Jsou obě dostatečně velká, tzn. nejsou blízka nule:  $\lambda_1 \wedge \lambda_2 \not\approx 0$
  - (b) Jsou si dostatečně blízka, například:  $\lambda_1 > \lambda_2$  avšak ne  $\lambda_1 \gg \lambda_2$

Systém se tedy chová tak, že při změně na vstupu dojde k viditelné změně na výstupu. Nesmí nastat situace, že dojde-li ke změně na vstupu, na výstupu se nestane pozorovatelná změna. Konkrétní příklad jak je možno si tyto matematické podmínky představit vizuálně může být následující:

- Špatné podmínění bude například problém v jednobarevných regionech stejné intenzity, například modré nebe (Obrázek 3.6). V takovém případě by  $\lambda_1, \lambda_2$  byly obě blízke nule.
- Naopak dobré podmínění bude v regionech s pestrým spektrem, například záhon pestrobarevných květin (Obrázek 3.7).



Obrázek 3.6: Snímek nebe a špatné podmínění okolí



Obrázek 3.7: Pestrobarevný snímek s dobrým podmíněním

### 3.4 Praktické aplikace optického toku

Jedna z velice relevantních publikací pro tuhle práci je již o něco dříve zmíněný článek v kapitole 3 článek od Souhila a spol. *Optical Flow Based Robot Obstacle Avoidance*

Tato publikace pojednává o využití optického toku pro navigaci robota (vyhýbání se objektům) ve vnitřním prostředí. Vstupem algoritmu je obrazová sekvence zachycená vestavěnou kamerou na robotu v pohybu. Poté jsou informace o optickém toku extrahovány z obrazové sekvence tak, aby mohly být použity v navigačním algoritmu. Optický tok poskytuje důležité informace o prostředí robota, jako jsou: umístění překážek, směr robota, čas do srážky a hloubka. Strategie spočívá ve vyvážení množství levého a pravého bočního proudění tak, aby se zabránilo překážkám [16].

O dalším praktickém využití optického toku pojednává například publikace od Agarwala, Gupty a Kumara *Review of Optical Flow Technique for Moving Object Detection*. Tento článek obrací problematiku, kdy se tentokrát nepohybuje kamera ve statickém okolí, ba naopak hlavní myšlenkou je odlišení statického okolí od pohybujícího se předmětu v něm, kdy snímání probíhá také na statickou kameru. Zpracování je formou izolace pohybujícího se objektu od okolí, čehož výsledkem je detekce objektů [25].

Závěrem může být zmíněna i práce pocházející z *Chinese Academy of Sciences in Beijing* od autorů Huang a spol. *Optical Flow Based Real-time Moving Object Detection in Unconstrained Scenes*. Tato práce porovnávala několik přístupů (Gaussovský model, Optický tok a Optický gradientní tok) pro detekci pohybujících se objektů na statickém pozadí. Experimentální výsledky dosažené touto prací ukazují, že duální metoda vyhodnocování scény kvalitativně i kvantitativně (tzn. i časově pro *real-time* aplikace) překonává dosavadní prozkoumané metody [26].

## 4 Heterogenní rozhodovací systémy

Heterogenní rozhodovací systémy nebo také skupinové rozhodování (GDM, *group decision making* [27]) jsou systémy a způsob rozhodování, jež kombinují různé typy rozhodovacích modelů nebo algoritmů tak, aby dosáhly co nejlepších výsledků. Tyto systémy využívají diverzitu v rozhodovacích procesech, což může zahrnovat různé metody, jako jsou strojové učení, statistické modely, a tak dále. Tímto způsobem mohou systémy lépe zohlednit různorodost dat a situací a poskytnout tak robustnější a spolehlivější rozhodnutí.

Velkou inspirací pro zařazení tohoto způsobu rozhodování je oblast umělé inteligence a využití takzvaného *ensemble learningu*. Ansámblové učení kombinuje předpovědi více základních modelů. Teoretické a empirické výsledky prokázaly výhodu těchto kombinovaných souborů oproti jednotlivým modelům. Heterogenní souborové metody zahrnují různé typy modelů, hlavní myšlenka je ta, že kombinací různých typů základních modelů je výsledné rozhodnutí kvalitativně nadřazenější [28].

Relevance tohoto přístupu v této práci se vztahuje ke kapitolám 2.1.1 a 2.1.2 výše, jež pojednávají o možnostech navigace ve venkovním prostředí. Tato práce, zejména v praktické části, detailně rozpracovává potenciál metody Optického toku, pro využití k navigaci v tomto prostředí.

Primárním cílem tedy není představit a vyvinout jednu univerzální nadřazenou metodu, jež by se stala novým etalonem kvality, nýbrž hlavní snahou je naopak prozkoumat potenciál a identifikovat silné stránky a nedostatky této metody za účelem dosažení možné synergie s již zaběhlými přístupy.

### 4.1 Rozhodovací metody

Existuje několik metodik a principů jak zacházet s výstupy různých modelů, sensorů a algoritmů. Terminologie zde používána má své kořeny opět ve strojovém učení. Důvod je ten, že ač skupinové rozhodování je všudypřítomné, nejznámějším zástupcem mimo technické aplikace budiž parlamentní nebo jakékoliv jiné volby, jež vsází na rovnost hlasů mezi jednotlivými "modely", tak oblast strojového učení je nejbližší této práci.

Dalším praktickým příkladem jiného typu rozhodování může být například skupinové rozhodování v projektu, kde se může uplatnit princip váhování ve prospěch expertů na dané téma, kdy hlas hlavního SME (*Subject matter expert*) může být tím dominantním.

Jak je vidět, způsobů jak vyhodnotit několikero výstupů, jež se mohou lišit, je mnoho. Proto je zde uvedeno a popsáno pár nejznámějších a nejobecnějších, z nichž poté kapitola 4.2 vybere a navrhne implementaci těch nejslibnějších pro metodu Optického toku.

Pět nejpraktičtějších a nejčastěji užívaných rozhodovacích metod je:

- Hlasování

Jednoduchá metoda, kde každý model má hlas o stejné síle [29] a rozhodnutí je určeno většinovým hlasováním. Ideálem je zde lichý počet modelů aby nedocházelo k rovnosti hlasů v případě binárních problémů.

- Váhování

Každý model může mít přidělenou různou váhu [29] podle jeho přesnosti nebo důležitosti v kontextu dané úlohy. Každý může být specializovaný na trochu jiný typ problému, prostředí, ... Výsledné rozhodnutí je poté váženým průměrem výstupů jednotlivých modelů.

- Boosting

Je metoda, kdy dochází k postupnému učení více modelů tak aby každý následující model se snažil opravit a poučit z chyb modelů předchozích [30]. Tuto vlastnost je možno ještě umocnit tak, že chybná rozhodnutí budou mít v učení vyšší váhu a tak se další model bude specializovat na případy, kdy předchozí modely zaostávaly. Nicméně výsledné rozhodnutí opět bere v potaz všechny (tím pádem i dřívější) modely a rozhodnutí je poté opět dosaženo kombinací všech výstupů.

- Bagging

Znamená velmi obecně paralelní trénování více modelů na různých podmnožinách [31]. V kontextu této práce by bylo možno chápat tento způsob jako specializaci jednoho přístupu pro konkrétní prostředí. Například GPS pro globální navigaci a koordinaci úkolu, Lidary pro práci v noci a Optický tok pro reaktivní navigaci na lokální bázi.

- Stacking

Tato metoda je již relativně komplexní, kdy se natrénuje nový model, jež dokáže vyhodnotit výstupy základních modelů a určí jak je nejlépe kombinovat [32].

Všechny tyto přístupy umožňují efektivněji využít rozmanitost jednotlivých modelů a umožňují tak dosáhnouti lepších synergických výsledků nad robustnější bázi.

## 4.2 Optický tok v rozhodovacím systému

Pro implementaci optického toku k dosavadním metodám se jako nejvhodnější metody jeví první dvě, a to hlasování a nebo váhování, zejména kvůli jejich simplicítnímu charakteru a intuitivní aplikaci (Návrh v kapitole 4.2.1). Další praktické důvody tohoto výběru mohou být následující: Metody jsou 3 (GPS + Lidar + Optický tok), což za předpokladu jejich hodnotové ekvivalence může vytvořit spolehlivý robustní systém. Dále například implementace váhování může být výhodou za předpokladu práce s detailnějšími znalostmi o jednotlivých metodách. Po definování vhodných oblastí pro tu či onu metodu by se

nabízela možnost navrhnout adaptivní filtr, který by dal přednost té nejkvalifikovanější metodě pro danou situaci.

Na druhou stranu metody *baggingu* a *boostingu*, byť se při jistém obecném pohledu mohou jevit aplikovatelně, nebudou prozatím uvažovány, jelikož jejich nátura je více vhodná přímo pro aplikace strojového učení.

Poslední metoda *stackingu* by potenciálně mohla být velice plodná (při dobrém natrénování rozhodovacího modelu), nicméně pro účel této práce je příliš komplexní.

#### 4.2.1 Návrh rozhodovacího modelu

Dva návrhy, jeden pro jednoduché rovnocenné hlasování a druhý pro váhované hlasování, budou předvedeny na následujícím jednoduchém příkladu:

*Před robotem byla detekována překážka, které je nutno se vyhnout. Robot má implementovány následující 3 řídicí systémy: GPS, Lidar a Optický tok. Každý systém je nezávislý a dává jedinečnou výstupní informaci, jak se detekované překážce vyhnout, kdy možnosti jsou dvě "vpravo = 0 nebo vlevo = 1.*

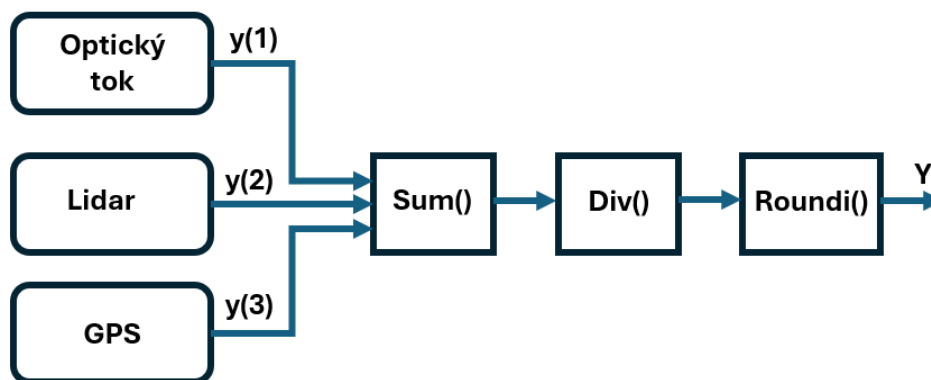
- Hlasování

U klasického jednoduchého hlasování mají všechny modely stejnou váhu a tak se výsledné rozhodnutí řídí většinou. V tomto triviálním příkladu jsou možné výsledky pouze dva a to 3:0 nebo 2:1 ve prospěch jedné či druhé strany.

Matematicky, by se takové rozhodnutí dalo formulovat následovně:

$$\text{round}_i\left(\frac{y(1) + y(2) + y(3)}{3}\right) = Y \quad (4.1)$$

Kde  $y(k)$  je výstup jednotlivého modelu, jenž může nabývat hodnoty pouze 0 nebo 1, jak je definováno v zadání.  $\text{Round}_i()$  je zaokrouhlení výsledku na celé číslo. A výsledek  $Y$  je celkové rozhodnutí heterogenního systému. Graficky je tento model zobrazen na schématu 4.1 níže.



Obrázek 4.1: Heterogenní rozhodovací systém - volba hlasování



- Váhování

U váhovaného hlasování může být situace o něco zajímavější. Každý jednotlivý výstup modelu je zde vynásoben jeho váhou  $w(k)$  v závislosti na konkrétní situaci/prostředí/... ve kterém se děje rozhodnutí. Platí také, že:

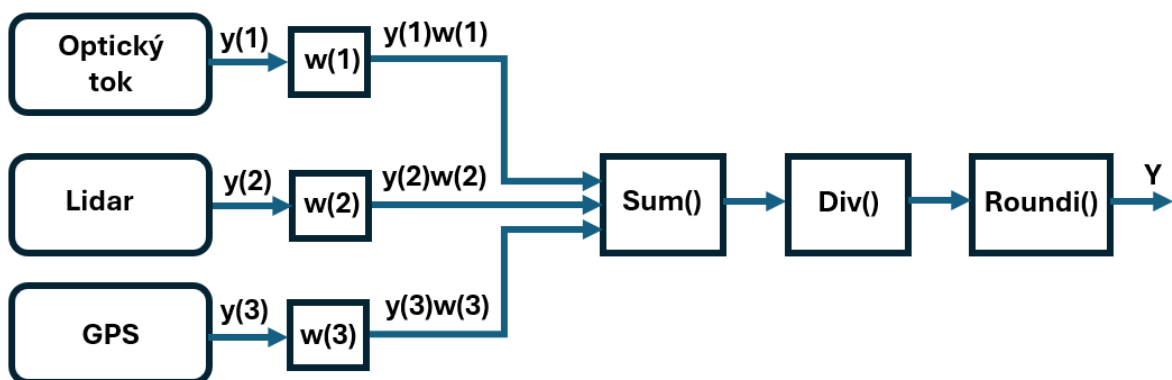
$$w(1) + w(2) + w(3) = 10 \quad (4.2)$$

Rovnice 4.2 říká, že je k dispozici celkem 10 bodů na přerozdělení pro jednotlivé modely (Počet bodů je zvolen čistě jako demonstrativní faktor). A výsledná rozhodovací rovnice se obohatí na:

$$\text{round}(\frac{\sum y(k)w(k)}{\sum w(k)}) = Y \quad (4.3)$$

Mějme tedy příklad, kdy optický tok, specializovaný na rychlé a reakční rozhodnutí, bude mít momentální váhu  $w(1) = 6$ . Lidar váhu  $w(2) = 3$  a GPS váhu  $w(3) = 1$ . V tomto specifickém případě by byl optický tok rozhodující a nepřehlasovatelný a ani poměr hlasů 1:2 by nebyl rozhodující. Klíčovou roli by zde tedy hrála specializace a váhy.

Grafické zobrazení modelu by vypadalo následovně:



Obrázek 4.2: Heterogenní rozhodovací systém - volba váhování

# 5 Matlab a optický tok

Pro praktické zpracování této diplomové práce bylo použito prostředí Matlabu. Tato volba byla udělána na základě několika kritérií:

- Matlab je vysokoúrovňový programovací jazyk, což usnadňuje práci s komplexními datovými soubory jakými jsou v tomto případě videa.
- Toto prostředí nabízí možnost využití pokročilých knihoven, které usnadňují výslednou práci. Konkrétně pro tuto práci je důležité zmínit využití *Computer vision toolboxu*.
- V neposlední řadě Matlab poskytuje velmi dobré možnosti co se týče exportování, sdílení, zálohy a také prezentování výsledků.

## 5.1 Computer vision toolbox

Computer vision toolbox je nadstavba pro Matlab, která poskytuje před připravené funkce, aplikace a další komplexní nástroje a algoritmy zjednodušující analýzu a testování konceptů souvisejících s počítačovým viděním.

Toolbox usnadňuje práci například při detekci objektů, při vizuální inspekci a analýze obrazu nebo také poskytuje možnost využití již před-trénovaných neuronových sítí.

V této práci je využita metoda Lucas Kanade: *opticalFlowLK()*, diskutována v kapitole 3.3 nebo také funkce *estimateFlow()*, jež slouží pro výpočet optického toku ve videích.

## 5.2 Výpočet optického toku

Pro prvotní výpočet optického toku je nutno nejdříve provést několik nezbytných kroků v tomto pořadí:

- Mít k dispozici na svém zařízení video k analýze.
- Uložit absolutní cestu k tomuto videu v podobě stringu.
- Použít vhodnou funkci pro přečtení tohoto souboru, například *VideoReader()*, jež v tomto případě vytvoří objekt obsahující toto video a další jeho vlastnosti.
- Přednastavit vlastnosti funkce *opticalFlowLK()* pro pozdější použití.

Tato příprava může být řešena například takto:

```
videoFile = 'C:\Users\...\Video.mp4';
videoObj = VideoReader(videoFile);
opticFlow = opticalFlowLK('NoiseThreshold', 0.01);
```

Finálním cílem je výpočet ale i zobrazení optického toku na videu (= zpracování videosouboru), avšak optický tok vypočítává a analyzuje posuny mezi jednotlivými snímky ve videu a proto je potřeba vnímat video jako sekvenci za sebou jdoucích jednotlivých obrázků. To znamená, že toto zpracování běží ve *while* smyčce a je nutno projít takto celé video snímek po snímku:

```
while hasFrame(videoObj)
    frame = readFrame(videoObj);
    flow = estimateFlow(opticFlow, rgb2gray(frame));
    ...
end
```

Tato část kódu zpracuje video ve smyslu po sobě jdoucí sekvence snímků dokud všechny nejsou vyčerpány o což se postará příkaz: *hasFrame()* a pro každý jednotlivý snímek *frame* provede výpočet optického toku *estimateFlow()* s předem nastavenými vlastnostmi a to celé na videu převedeném do černo-bílého spektra.

Je zde také odhalena jedna praktická limitace a to, že je tato metoda náročná na výpočetní zdroje, jelikož je analyzován každý snímek ve videu zvlášť a nezpracovává se video jako celek. Lze tedy tvrdit, že rychlost zpracování vysoce koreluje se snímkovací frekvencí původního videa: FPS (= Frames Per Second).

jeden konkrétní výstup z této operace můžeme vidět na obrázku 5.1.

```
Vx: [1080×1920 single]
Vy: [1080×1920 single]
Orientation: [1080×1920 single]
Magnitude: [1080×1920 single]
```

Obrázek 5.1: Výstup výpočtu optického toku

Obrázek 5.1 ukazuje, že výstupem jsou 4 matice o stejném rozlišení jako původní video. Tyto 4 matice tvoří dvě skupiny dávající tentýž výstup v rozdílné formě. První pár tvoří *Vx* a *Vy* (kapitola 5.2.1) a druhý *Orientation* a *Magnitude* (kapitola 5.2.2).

Nutno zdůraznit, že všechny tyto matice sestávají z datového typu *single* a obsahují nenulové číslo právě když pro konkrétní pixel bylo možné vypočítat optický tok (= jeho posun mezi jednotlivými snímky). Pakliže nebylo možné optický tok určit pro daný pixel je hodnota rovna  $0$ .

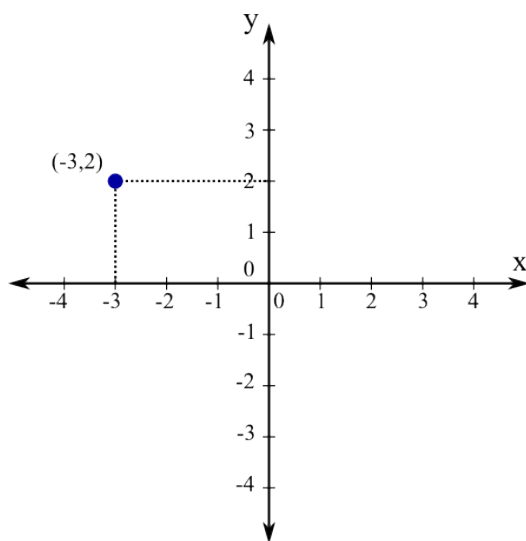
Detail výpočtu optického toku je v kapitole 3.3, nicméně pro vysvětlení tvrzení výše

je možno ve zkratce uvést, že k výpočtu optického toku je nutno aby se v analyzovaném obraze nacházely detekovatelné struktury.

Pakliže bychom měli jednobarevné video velikosti například  $100 \times 100$  px typu  $uint8 = 255$  nenacházely by se zde žádné detekovatelné struktury a proto by byly všechny výstupy (= elementy matic) rovny 0.

### 5.2.1 Kartézský výstup

První dvojice výstupů z obrázku 5.1 je dvojice  $(v_x, v_y)$ , graficky znázorněno na Obrázku 5.2, která popisuje optický tok v pravoúhlých kartézských souřadnicích.

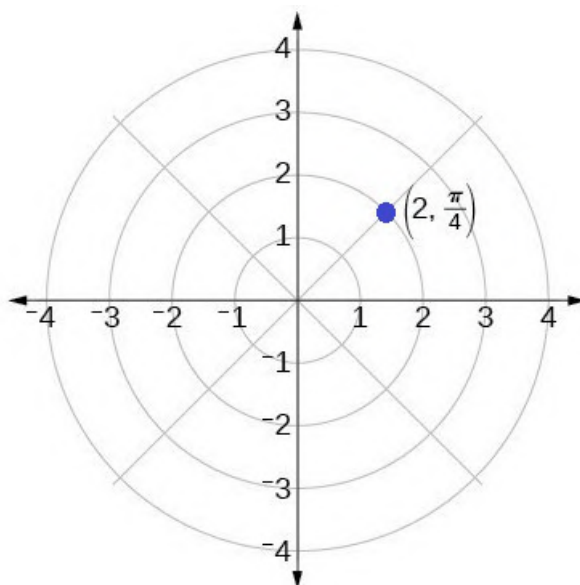


Obrázek 5.2: Standardní zápis bodu  $(-3, 2)$  v kartézském systému

Tento zápis znamená, že každý pixel je popsán dvojicí čísel, která říká jak rychle se pohybuje ve směru osy  $x$ , v tomto případě tedy  $v_x$  a ve směru osy  $y$  ve formě  $v_y$ , a jakým směrem. Velikost obsahuje informaci o rychlosti a znaménko dává údaj o směru v klasické kartézské konvenci kdy  $+$  znamená pohyb v kladném směřování os  $x$ ,  $y$  a znaménko  $-$  proti směru těchto os.

### 5.2.2 Polární výstup

Druhá dvojice výstupů z obrázku 5.1 je dvojice *Orientation, Magnitude*, která popisuje optický tok v polárních souřadnicích. Grafická reprezentace rozdílného zápisu na Obrázku 5.3.



Obrázek 5.3: Standardní zápis bodu  $(2, \pi/4)$  v polárním systému

Pohyb každého pixelu mezi jednotlivými snímky je v tomto případě popsán jeho absolutní velikostí (= Magnitude) a úhlem tohoto pohybu (= Orientation), dále odkazováno jen jako dvojice: (*Velikost, Úhel*).

### 5.3 Pilot a šipky

Prvním průzkumným pilotním videem určeným ke zpracování byl záznam přímo z prostředí potenciální aplikace: louky s křovinami. Zpracování, které spočívalo ve výpočtu a zobrazení optického toku na původní video, ve formě vektorů z pixelů s nejvýraznějším tokem, (ukázka z videa na Obrázku 5.4), stanovilo úvodní *benchmark* pro další práci.



Obrázek 5.4: Ukázka z pilotního zpracování prvního analyzovaného videa

Toto prvotní zpracování obrazu přineslo 2 zásadní poznatky:

1. Potvrzení kompatibilitosti zvoleného softwaru, toolboxů a videozáznamů. Také jistý *check-point* dosavadních přípravných prací v pozadí potvrzující hardwarově-softwarovou proveditelnost této práce.
2. Zviditelnění prvních výzev, jež metoda přináší. Například optická evaluace výsledků, komplexnost (výpočetní) a náročnost (časová) na zpracování videí a výzvu k celkovému porozumění výstupů.

Z Obrázku 5.4 je možno vidět, že prozatímní zpracování pilotního videa působí spíše jako změt náhodných modrých šipek bez hlubší informace. Dalším nepraktickým faktem je to, že tato práce pojednává o zpracování videa, nicméně v této psané podobě je možno prezentovat a odkázat se pouze na jednotlivé snímky pocházející z tohoto videa. Pro kompletní zpracovaná videa je nutno prohlédnout příložené přílohy k práci.

V tomto okamžiku práce došlo k nápadu využít jako doplněk *ideální* testovací prostředí, které pomůže zestříhlit komplexitu úkolu například tím, že nebude zatíženo některými proměnnými z reálného světa (rozličná intenzita osvětlení, nevýrazné překážky, ...).

Z kapitoly 2.2 je již známo, že testovacím prostředím byla zvolena hra *Trackmania* a také z jakých důvodů. V editoru byla vytvořena testovací trať a problematika byla přenesena do virtuálního prostředí.

První testovací koncept v *Trackmanii* měl co nejvíce zjednodušit prostředí, vyřadit nadbytečné proměnné a doručit co možná nejizolovanější podmínky pro pochopení konceptu a porozumění získaných vjemů a výstupů. Toto zachycuje první pilot z *Trackmanie* na Obrázku 5.5



Obrázek 5.5: Ukázka z prvního zpracovaného videa v *Trackmanii*

Obrázek 5.5 je v porovnání s venkovním prostředím lepší vizuální interpretací optického toku. To znamená, že koncept virtuálního a zidealizovaného prostředí se ukázal být vhodný tím, že eliminuje nežádoucí vlivy komplexního prostředí a umožní postupný přechod z teorie do praxe skrze tento parciální krok.

Struktura navržené trati je nejdříve pořád rovně kdy se kužely nachází po stranách a na začátku se jede pouze rovně tak aby se eliminovala rotace obrazu. V pozdější části trati jsou i zatáčky a další kužely postaveny přímo do cesty tak aby bylo nutné se jim vyhnout.

První poznatek z testování je, že kužely nejsou úplně ideální objekty, jelikož jejich barva je jednolitá šedá a nejsou na nich výrazné vzory, jichž by se algoritmus mohl chytit (jen okraje, což je v tomto případě naštěstí dostačující). Ideální překážka má jasné definované přechody a hrany, ne jen strukturálně, ale nejlépe i barevně.

Tato první ukázka také demonstruje a potvrzuje koncept z teorie o optickém toku, přesněji o tom, jak by měl fungovat v tomto speciálním případě, při přímočarém pohybu vpřed. Dle teorie by objekty napravo od pozorovatele měly směřovat napravo, to znamená, že pravý kužel se při přibližování pohybuje po obraze vpravo, přesně jak indikují šipky na Obrázku 5.5. Naopak levý kužel se opticky v čase pohybuje nalevo. Stejně pravidlo platí i pro objekty střechy v dálí, jenž směřují nahoru a trať, po které auto jede, se opticky pohybuje dolů, jelikož pozorovatel, nacházející se za autem, ji mívá nadjetím.

### 5.3.1 Ideální virtuální prostředí

Ještě předtím, než tato práce začne pojednávat o chování optického toku pro zatáčející objekt, neboli jinak řečeno rotujícího pozorovatele a výzvami s tímto souvisejícími, je vhodné vložit zde malou podkapitolu zabývající se evolucí myšlenky pro nalezení *ideálního* testovacího prostředí.

Jak je totiž možno vidět v kapitole 5.3 z Obrázku 5.5, prostředí je stále velmi rušivé. Dráha a krajnice jsou rušivé elementy pro tento způsob zpracování, mřížky ve vozovce jsou také až příliš rušivé a nevhodné pro analýzu a také překážky by mohly být lépe rozlišitelnější od okolí (momentálně například stejná barva s vozovkou).

Z tohoto důvodu byly provedeny dvě iterace nových map adresující přesně tyto problémy. První iterace se zaměřila na problematiku lepších překážek (= jiný typ) a zjednodušení vozovky. Výsledkem je trať na Obrázku 5.6, která přinesla jednolitou šedou vozovku a výraznější překážky bez rušivých elementů jako byly například mřížky.



Obrázek 5.6: První iterace vylepšeného prostředí v Trackmanii

Tento koncept byl lepší než předchozí. Vozovka nebyla tak rušivá a překážky byly lépe diferencovatelné od okolí, byť barevně pořád ještě neidealizované (stále jednolitá bílá). Z tohoto zkoumání vyvstala ještě jedna možná iterace, která řešila tyto nedokonalosti ještě lépe a vzniknout dala konceptu na Obrázku 5.7, kdy vozovka zmizela úplně a byla nahrazena naprosto jednolitou barvou zelené trávy, která jednak pomohla odlišení podkladu od překážek barevně ale také vyřadila veškeré spoje a přechody. A v neposlední řadě byly identifikovány překážky, které obsahovaly jasně viditelné (černo-oranžové) přechody.



Obrázek 5.7: Druhá iterace prostředí v Trackmanii

## 5.4 Problém s rotací

Předchozí podkapitola 5.3 ukázala vhodnost volby testovacího prostředí (= identifikace ideálních podmínek) a také demonstrovala jeden speciální případ pohybu: Přímočarý vpřed. Logickým rozšířením zkoumání je tedy podívat se, co se děje s optickým tokem pakliže se lehce zvětší komplexnost o rotaci pozorovatele. V tomto případě to znamená zatáčení vozidla, jelikož pro úspěšné vyhýbání se překážkám je nutno provést úhybný manévr.

Nejlepší demonstrací, co se děje při zatáčení je video pořízené z dronu ve venkovním prostředí. Pro porovnání Obrázek 5.8 ukazující optický tok při přímočarém pohybu rovně, oproti Obrázku 5.9, ukazující optický tok při rotaci o pár vteřin později ze stejného videa.





Obrázek 5.8: Přímočarý pohyb vpřed



Obrázek 5.9: Optický tok v průběhu rotace

Obrázek 5.8 potvrzuje koncept již demonstrováný z Trackmanie, nicméně tentokrát přenesený do reálného prostředí. Objekty, v tomto případě stromy, napravo při přímočarém letu mají tendenci opticky pohybovat vpravo. Naopak stromy nalevo ukazují tendenci pohybu vlevo.

Problém přichází při představení rotace do rovnice výpočtu optického toku, jak vidno na Obrázku 5.9. Pakliže pozorovatel začne neúměrně rychle rotovat, vůči svému dopřednému pohybu, dojde k tomu, že v obrazu naprosto převládne vliv této rotace nad přímočarým pohybem a opticky to tedy vypadá, že všechny objekty mají v tomto případě tendenci se pohybovat vpravo.

Příčinou tohoto jevu je nepoměr mezi příspěvky do *Velikosti magnitudy* od obou typů pohybu: rotace + přímočarý pohyb.

# 6 Vidění robota

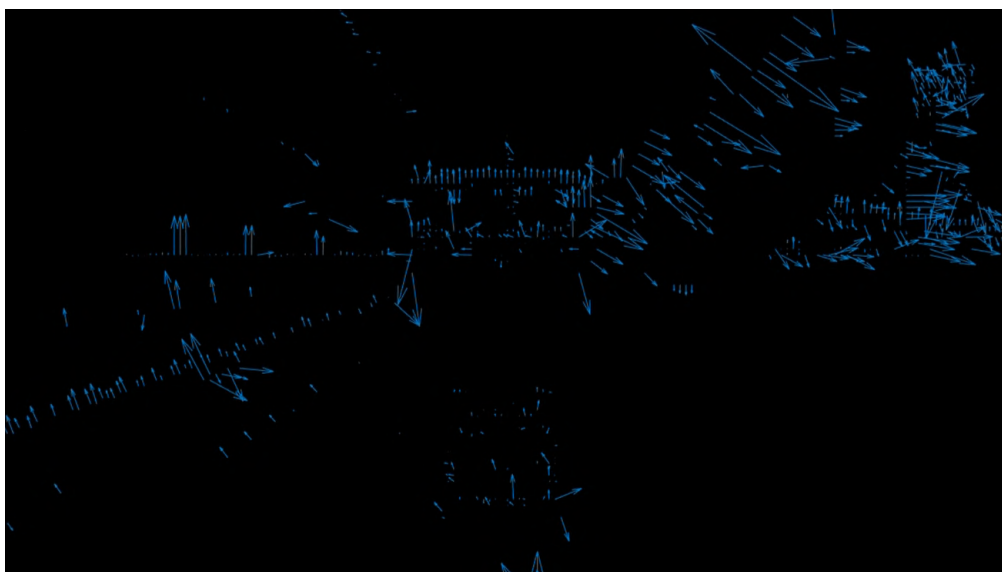
Otevřený konec předchozí kapitoly znamenal zlom v průběhu této práce. Vyplynulo z ní, že není možné se omezit jen na práci s vizuálním výstupem (šipkami) a je nutno pro další postup redefinovat mentalitu jak k datům přistupovat a jak s nimi zacházet.

Zde došlo ke konfrontaci toho co *vidí* člověk a jak je možno vizuálně hodnotit tyto optické výstupy s klíčovou otázkou: *Co vlastně vidí robot?*

## 6.1 Izolace šipek

Pomoc k pochopení otázky *Co vidí robot* přineslo prozkoumání možnosti zobrazení optického toku na plně černém neutrálním podkladu. Toho bylo dosaženo tak, že se vytvoří černá matice  $uint8 = 0$ , o stejné velikosti jako je zpracovávané video a na ní se zobrazí optický tok jako doposud.

```
figure;  
    blanc = uint8(zeros(1080, 1920));  
    imshow(blanc);  
        hold on  
plot(flow, 'DecimationFactor', [10 10], 'ScaleFactor', 50);  
        hold off
```

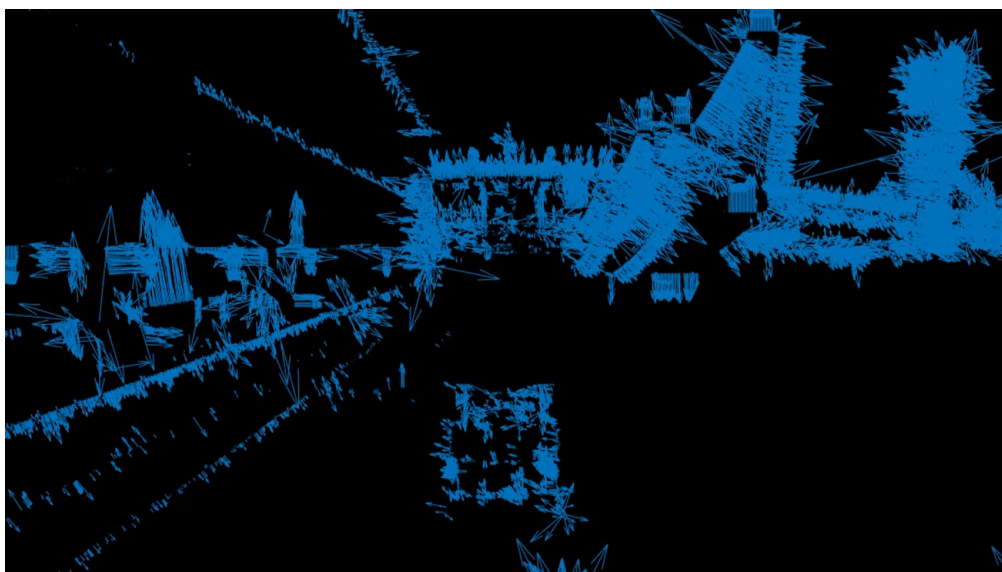


Obrázek 6.1: Optický tok na černém pozadí

Stručná část kódu pro takové zpracování je výše a jeho výsledek je na Obrázku 6.1. Snímek pochází z Trackmanie, a je na něm možno pozorovat dole uprostřed obrysy formule a shluky optických toků ukazují přítomnost překážek, například napravo lze odhadovat přítomnost zdi, nalevo malou krajnicí a vzadu další větší překážku.

Po této analýze se zobrazení přibližuje tomu co vidí robot, nicméně pořadí je část výsledků skryta. Důvodem je například filtrace velmi malých, sotva detekovatelných toků, pocházejících z nastavení *DecimationFactor* na  $[10, 10]$ , což má za následek skrytí těch nejmenších šipek. K tomu jsou dva důvody: zrychlení analýzy a zpřehlednění výstupu.

Avšak pokud je *DecimationFactor* nastaven na  $[2, 2]$  výsledkem na Obrázku 6.2 je daleko více optického toku, kdy dojde k výrazné proměně toho co lze na videu vidět.



Obrázek 6.2: Optický tok na černém pozadí i s menšími *Velikostmi* toku

Následkem je zhruba desetinásobná doba výpočtu a zobrazení videa, nicméně poskytuje daleko zajímavější strukturální pohled na optický tok. Nyní nehraje hlavní prim to, kam šipky směřují, nicméně z videa začínají vystupovat dobře viditelné struktury.

Tohoto pozorování bylo využito v následující kapitole 6.2, kde se redefinuje přístup k vnímání toku a k samotnému zpracovávání získané informace.

## 6.2 Binární využití magnitudy

V kapitole 5.2 již byla naznačena myšlenka kdy je výstup optického toku rovný  $0$  a kdy je naopak nenulový. Při spojení této informace se strukturálním zjištěním z předchozí kapitoly je možné se podívat na otázku optického toku binárně:

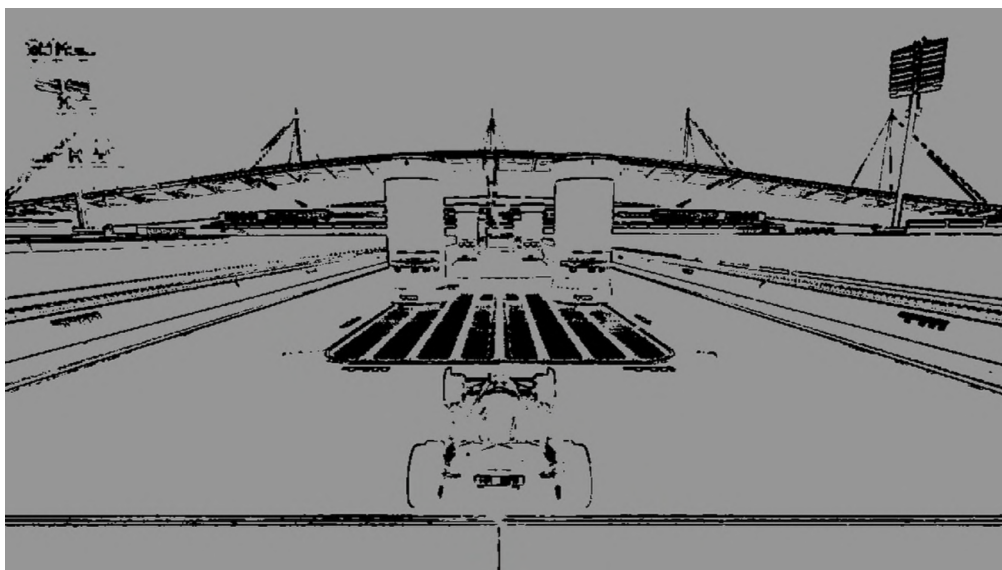
- Při pohledu na optický tok zapsaný v polárních souřadnicích jako:  
Tok = (Velikost, Úhel) zapomeneme na informaci z Úhlu, zapomeneme také, že Velikost může nabývat jakékoli hodnoty od  $0$  po nekonečno. Jediné rozlišení bráno v potaz pro tento okamžik je otázka: Velikost =  $0$ , nebo Velikost  $\neq 0$ ?

Toto znamená, že limitovat se pouze na šipky v obraze by znamenalo nevyužít všechny dostupné informace z výpočtu.

Toto zpracování probíhá následovně:

- Vytvořit černou podkladovou matici.
- Lokalizovat ve vypočtené matici optického toku všechny nulové elementy, která označují prázdná místa bez překážek.
- Přepsat a odlišit tak v původní černé matici tyto elementy na černo-bílý odstín, například  $uint8 = 150$ .
- Zobrazit tuto masku přes původní matici.

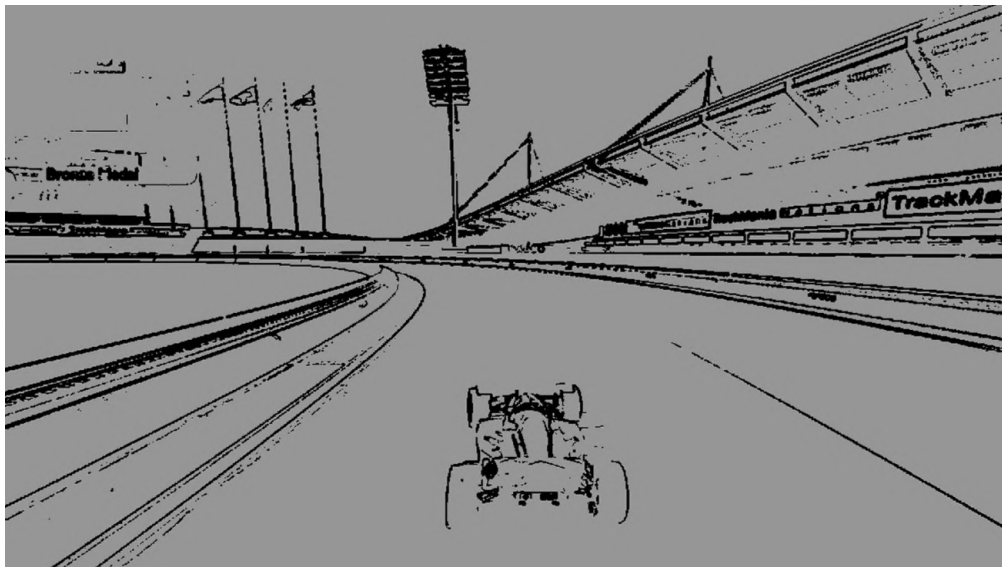
```
blanc = uint8(zeros(1080,1920));           % Base matrix
ZeroIndeces = find(flow.Magnitude == 0);  % Identify Zero elements
blanc(ZeroIndeces) = 150;                 % Mask
imshow(blanc);                             % Visualization
```



Obrázek 6.3: Binární zobrazení *Velikosti* optického toku v Trackmanii

Výsledkem takového zpracování je výše Obrázek 6.3. Toto nové vnímání a zobrazení je klíčovým zlomem pro tuto práci. Obraz je získán pouze z výpočtu optického toku. Nebylo použito žádných aditivních metod, nýbrž jen extrakce informace poskytnuté tímto výpočtem z videozáznamu. Momentálně dosažený výsledek ukazuje velmi dobré vidění z pozice robota. Navíc, kvalita tohoto zpracování není závislá na rotaci (tak jak bylo problémem výše) a nějakých dalších faktorech, viz. Obrázek 6.4.

Většina struktur je jasně viditelná a detekovatelná. Samozřejmě ne všechny hrany jsou dokonalé, ovšem oproti dosavadním výsledkům je toto jasný posun vpřed. Limitací prozatím zůstává hloubka obrazu kvůli použití pouze dvou barev, černé a šedé. Tento problém je adresován v kapitole 7.



Obrázek 6.4: Binární zobrazení *Velikosti* optického toku při zatáčení

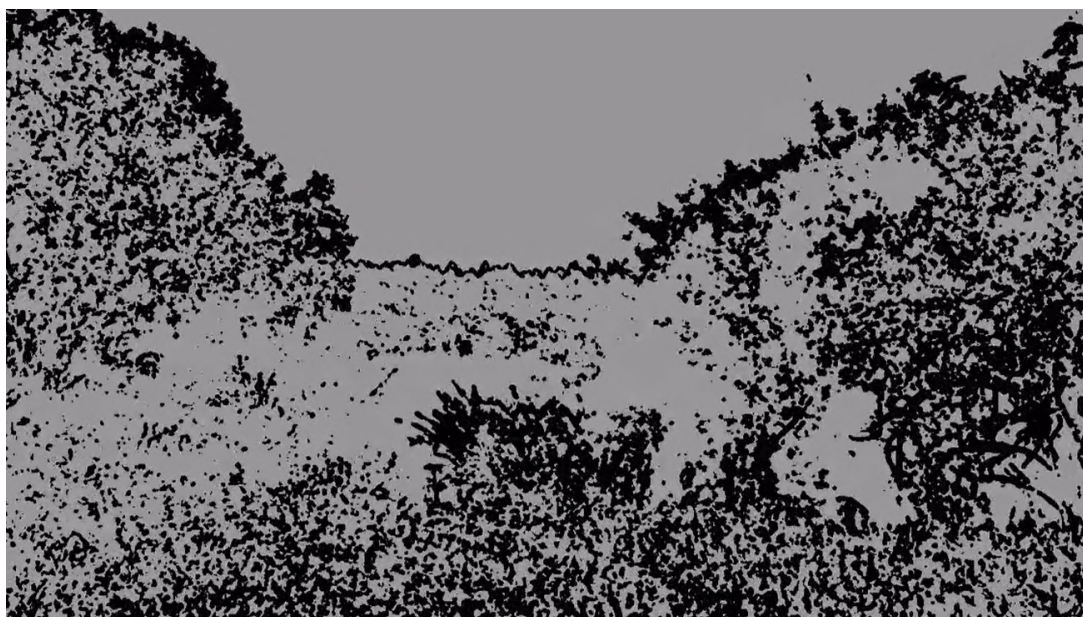
Logickým pokračováním a otevřenou otázkou je nyní přenositelnost tohoto řešení do venkovního prostředí. Trackmania je stále pouze testovacím prostředím a mezikrokem k aplikovatelnosti venku. Následující sekvence obrázků (= ukázek/výseků ze zpracovaných videí) bude demonstrovat přenositelnost tohoto řešení ven do 3 odlišných prostředí:

- Původní video z polí = těžko charakterizovatelné prostředí
- Video pořízené dronem z lesa = prostředí s nižší intenzitou světla
- Vinice = strukturované prostředí s mnoha drobnými překážkami

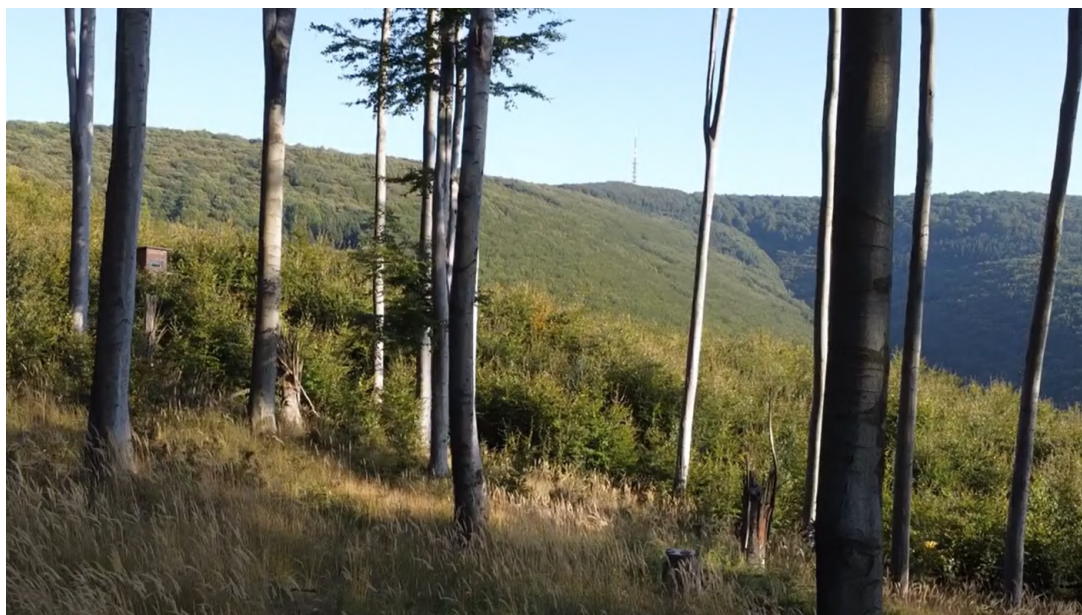
Proto aby čtenář mohl srovnat kvalitu výstupu, všechny tyto ukázky jsou ukázány ve dvojicích pod sebou: původní + zpracovaný obrázek po extrakci informace z výpočtu optického toku.



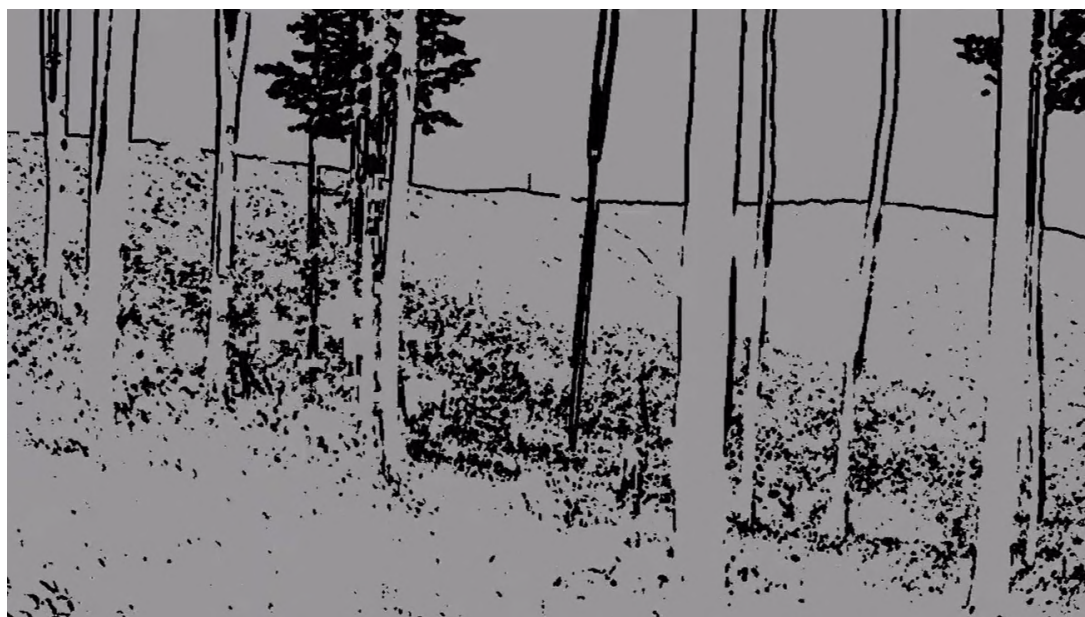
Obrázek 6.5: Originál video z pole



Obrázek 6.6: Video extrahované z výpočtu optického toku



Obrázek 6.7: Originál video z lesa pořízené dronem



Obrázek 6.8: Video z dronu po extrakci obrazu z optického toku



Obrázek 6.9: Originál video z vinic



Obrázek 6.10: Extrahované video z vinic

Možno si povšimnou, že úroveň vidění je na zpracovaných obrázcích, které představují vidění robota, získané pouze výpočtem optického toku a nijak jinak, opravdu přelomová. Jistě, úroveň detailu má ještě znatelné nedostatky ale jako metoda, která dává globální představu o prostoru kolem pozorovatele, již osvědčuje své kvality.

Nutno také podotknout, že zpracování a zobrazení, co se týče časové náročnosti, opět klesla na velmi přijatelnou úroveň. Na jednu vteřinu videa vychází zhruba 2 vteřiny zpracování, což dává poměr 1:2, toto ještě není samozřejmě aplikovatelné na *real-time* aplikace, nicméně je to daleko blíže než metoda *všech šípek* z Obrázku 6.2, která měla poměr zhruba kolem 1:10.



Lze také vidět, že optický tok obsahuje daleko více informační hodnoty, než bylo zřejmé na prvních pilotních videích. Je možné extrahovat daleko větší potenciál *vidění* a detailů než doposud, což znamená kladný posun pro tuto práci.

A pro doplnění dosažených výsledků tohoto postupu je to, že doposud rotace působila spíše jako nechtěný jev, jelikož při rotaci byla výsledná informace ztracena. Toto zde neplatí, ba naopak, rotace dokonce napomáhá se zachycením hran a přechodů a dodává ostrost a detaily do výsledného vidění, což je vidět zejména na Trackmanii, Obrázek 6.4. Nehraje roli, jestli je to rovinka nebo zatáčka, dokud je obraz v pohybu (podmínka nutná pro výpočet optického toku), informace a obrysy jsou vidět.

V neposlední řadě je ale nutno upozornit i na nedostatek jakým se ukazuje být intenzita světla. Pakliže povrch, překážky, atd. nejsou dostatečně osvětlené, stává se, že některé hrany a detaily jsou více náchylné k nedetekovatelnosti. Tento jev je nejlépe vidět na videu z dronu (Obrázek 6.8, vlevo dole), které je pořízeno v lese se zhoršenými světelnými podmínkami a některé detaily, zejména na povrchu ve stínu zůstávají nezachyceny.

### 6.2.1 Histogram videa

Na předchozí úspěšnou myšlenku zobrazení *vidění robota* bylo navázáno dalším zkoumáním ohledně rozdělení pixelů ve videu na ty, které nesou informaci (= mají nenulovou *Velikost*) a na ty, jež naopak nenesou informaci o optickém toku (= mají nulovou *Velikost*).

Toto zkoumání bylo provedeno dynamicky skrze celé video, kdy byl pro každý snímek zobrazen poměr těchto pixelů do histogramu a sledoval se vývoj tohoto poměru v době trvání videa. Hypotéza byla taková, že potenciální vzestup poměru ve prospěch pixelů nesoucích informaci bude znamenat vyšší pravděpodobnost, že se v zorném poli nachází překážka, již se bude nutné vyhnout.

Implementace této části v Matlabu spočívala v sečtení všech nulových a nenulových elementů zvláště pro každý jednotlivý snímek ve videu a výpočet tohoto poměru pro vizualizaci histogramu i tohoto poměru.

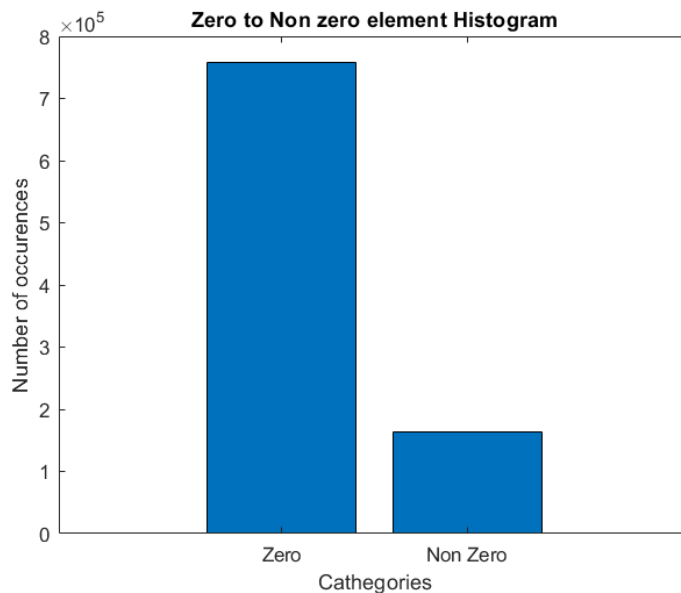
Klíčové části kódu jsou zobrazeny níže:

```
numZeros = sum(blanc(:) == 0);           % Sum all zeros
numNonZeros = sum(blanc(:) ~= 0);       % Sum all non zero elements
ratio(i) = numZeros / numNonZeros;      % Ratio

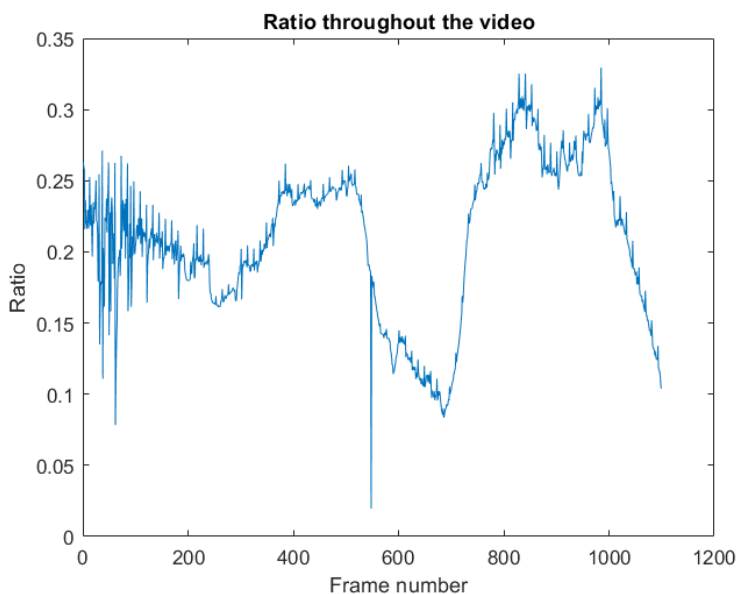
figure;
    bar([numNonZeros, numZeros]);
    title('Zero to Non zero element Histogram');
    set(gca, 'XTickLabel', {'Zero', 'Non Zero'});

figure;
    plot(1:1:i-1, ratio);
    title('Ratio throughout the video');
```

Tento způsob analýzy byl aplikován na několik dostupných videí z různých prostředí (Trackmania, vinice, dron, ...). Za všechny je uvedena například analýza videa z dronu v lese a jeden histogram (Obrázek 6.11) a vývoj poměru pixelů (Obrázek 6.12).



Obrázek 6.11: Příklad histogramu z videa z dronu



Obrázek 6.12: Poměr pixelů v průběhu videa z dronu

Je nutno mít na paměti, že histogram ukazuje pouze jeden jistý poměr v určitém okamžiku, kdežto poměr vykreslený jako křivka ukazuje všechny tyto histogramy v průběhu celého videa. Pro explicitní doplnění informace z kódu výše o tom, jakou informaci nese histogram/poměr:

- *Poměr blízký se 0 znamená drtivou převahu pixelů bez informace a poměr blízký se 1 by znamenal drtivou převahu ve prospěch pixelů s informací. Presentovaný poměr kolem 0.2 znamená, že zhruba 20% obrazu bylo detekováno jako potenciální překážka. Tento poměr se může lehce zvyšovat nebo snižovat v závislosti na různých faktorech.*

Na grafu poměru (Obrázek 6.12) je možno pozorovat jistý vývoj (stejně tak jako na všech dalších videích), nicméně tuto analýzu se nepodařilo dát do souvislosti s překážkami. Dochází sice k jistým výkyvům, avšak nepodařilo se prokázat, že ty jsou způsobeny překážkami v zorném poli. Jako více pravděpodobné ovlivňující faktory byly shledány vlivy od prostředí (kdy bohatější prostředí souvisí s větším množstvím detekovaných objektů, tudíž s vyšším poměrem), nebo také světelnosti (kdy vyšší světelnost umožňuje detekovat více hran a tudíž také zvyšuje poměr).

Pakliže se vezme v potaz velice explorativní povaha této práce, je samozřejmostí, že ne všechny úvahy vedou k vylepšení cíle a různým krokům vpřed jako byla binární analýza. Nicméně je nutné prezentovat i cesty, jenž byly později shledány slepými bez markantnějšího vlivu na výstup.

# 7 Hloubka obrazu

Dosavadní praktická část práce šla směrem od prvotních komplexních zobrazení v podobě vektorů, v kapitole 5.3, k jednodušší práci s výstupy, což přineslo výsledky v podobě kvalitního vidění z perspektivy robota v kapitole 6.2.

Nyní tato kapitola 7 nabere opět opačný trend, kdy vidění robota je bráno jako nalezený pevný základ, na kterém se bude toto řešení vyvíjet a zvyšovat jeho hodnota ale i komplexnost.

Jelikož zatím dosažené řešení je černobílé využívající pouze dvou odstínů, z pohledu robota i pozorovatele se jedná o pouhé 2D řešení, kterému chybí hloubka. A právě hloubka nebo také různé rychlosti objektů v obraze budou předmětem dalšího zkoumání.

## 7.1 Potenciálová mapa

Potenciálové pole v kontextu této práce je chápáno jako schopnost překážky ovlivňovat do jisté míry své okolí. Tato vlastnost se bude dát libovolně škálovat pro dosažení různých vlastností, jenž budou nabízet své specifické benefity pro popis prostředí v zorném poli.

Po nastavení žádaných parametrů a po zpracování takového obrazu/video je výsledkem *Potenciálová mapa*. Základem této potenciálové mapy je binární obraz, na nějž byly aplikovány dodatečné metody pro vytažení nebo zvýraznění dalších informací.

Dosavadní binární zpracování a zejména zobrazení bylo provedeno s šedým pozadím (to znamená  $uint8 = 150$ ) a černými překážkami ( $uint8 = 0$ ). Pro lepší vizuální výsledek potenciálových map byl vytvořen negativ tohoto zpracování, což znamená černý podklad ( $uint8 = 0$ ) a bílé překážky ( $uint8 = 255$ ). Důvod je ten, že potenciálové mapy již nejsou omezeny pouze na dva odstíny, nýbrž budou využívat celé černobílé spektrum od 0 do 255, kdy toto škálování nabízí na černém podkladu nejlepší kontrast a viditelnost.

Následný výpočet a zobrazení potenciálové mapy je proveden pomocí využití funkce *bwdist()*, jenž každému pixelu přiřadí hodnotu, která odpovídá Euklidovské vzdálenosti k nejbližšímu nenulovému prvku matice (= pixelu). Tato funkce vytvoří takzvanou transformační masku, která dále zakomponovává škálovací koeficient  $K$ , jenž určuje jakou *silou* ovlivňují překážky své okolí, kdy  $K = 1$  znamená minimálně a s rostoucím  $K$  roste tento vliv na okolí. Poslední částí je aplikace této masky a vytvoření výsledné potenciálové mapy. Tato mapa je vypočítaná opět pro každý jednotlivý snímek a tyto snímky za sebou vrátí zpracované video s potenciály.

Základní kameny tohoto provedení jsou zobrazeny v kódu níže.

```

%% Video and Optical flow calc.
frame = readFrame(videoObj);
flow = estimateFlow(opticFlow, rgb2gray(frame));

%% Inversion - Black and White img.
blanc = uint8(ones(1080,1920)) * 255;           % White image
ZeroIndeces = find(flow.Magnitude == 0);       % Black empty space
blanc(ZeroIndeces) = 0;                        % Remaining white = Obstacles

%% Potential Field
% Low impact K
K = 3;                                         % Scaling factor K - small
DistTr = bwdist(blanc) / K;                   % Distance to neearest non 0
PotentialField = uint8(255./(1 + DistTr));     % Scaled Potential Map 1

% High impact L
L = 50;                                       % Scaling factor L - big
DistTr2 = bwdist(blanc) / L;                 % Distance to neearest non 0
PotentialField2 = uint8(255./(1 + DistTr2)); % Scaled Potential Map 2

```

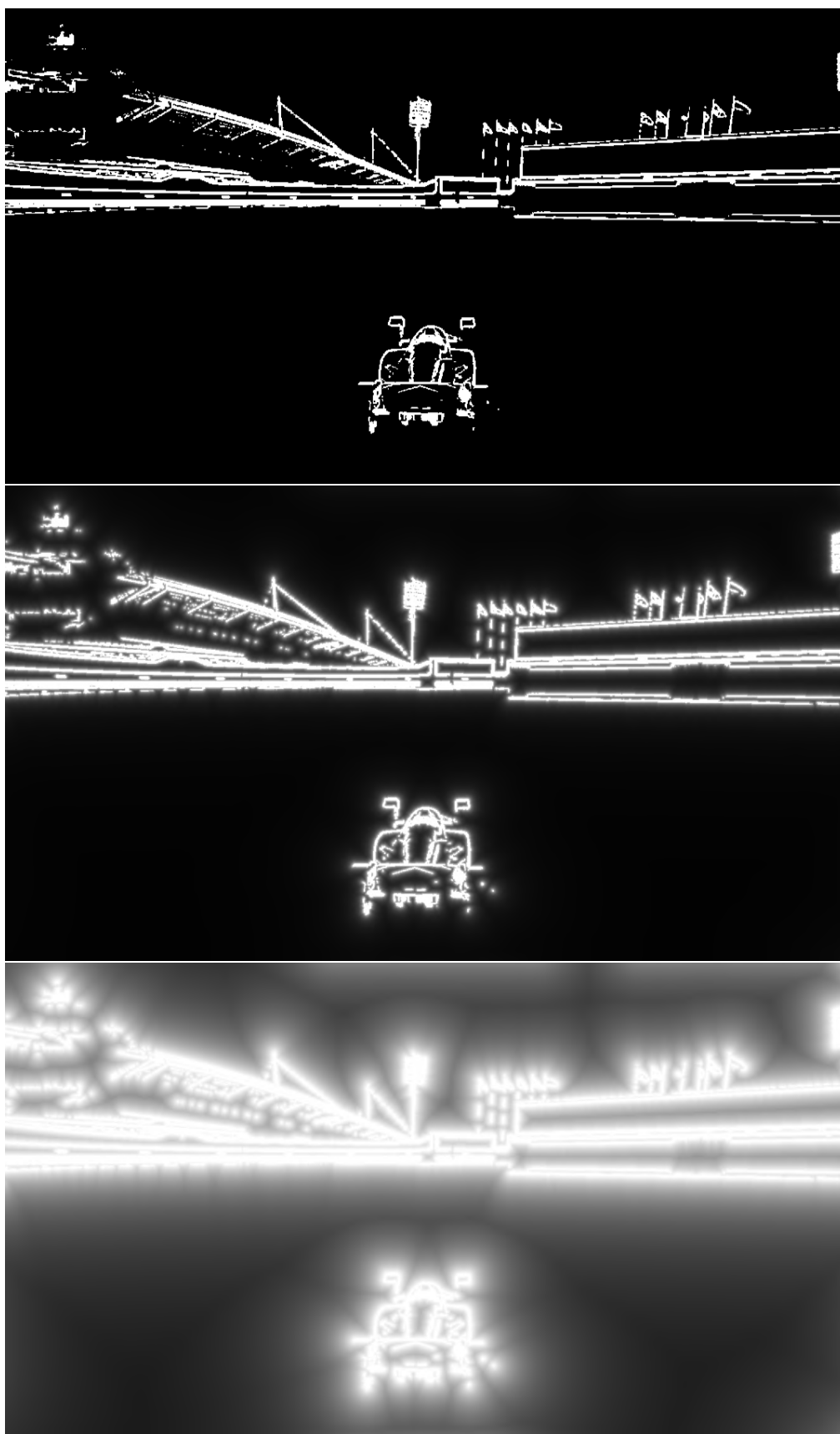
### 7.1.1 Analýza potenciálových map

V této podkapitole budou prezentovány výsledky analýzy videí stojících na základě binárního vidění obohacené o potenciálové mapy.

Předmětem zkoumání bude zejména vliv škálovacího faktoru  $K$ , případně  $L$ . Oba koeficienty hrají totožnou roli. Rozdělení je pouze z notačního hlediska, kdy při odkazování na faktor  $K$  je myšlen malý vliv/přesah překážek do volného okolí a pro faktor  $L$  je tento přesah vyšší. Rozdíl je zhruba o jeden řád, kdy  $K \in \langle 1, 10 \rangle$  a  $L \in \langle 10, 100 \rangle$

Prvním předmětem zkoumání bude testovací prostředí Trackmanie. Následující ukázky budou z reálného prostředí z lesa a z vinic. Všechny tyto ukázky budou pro pro trojici obrázků pod sebou umožňující přímé porovnání vlivů:

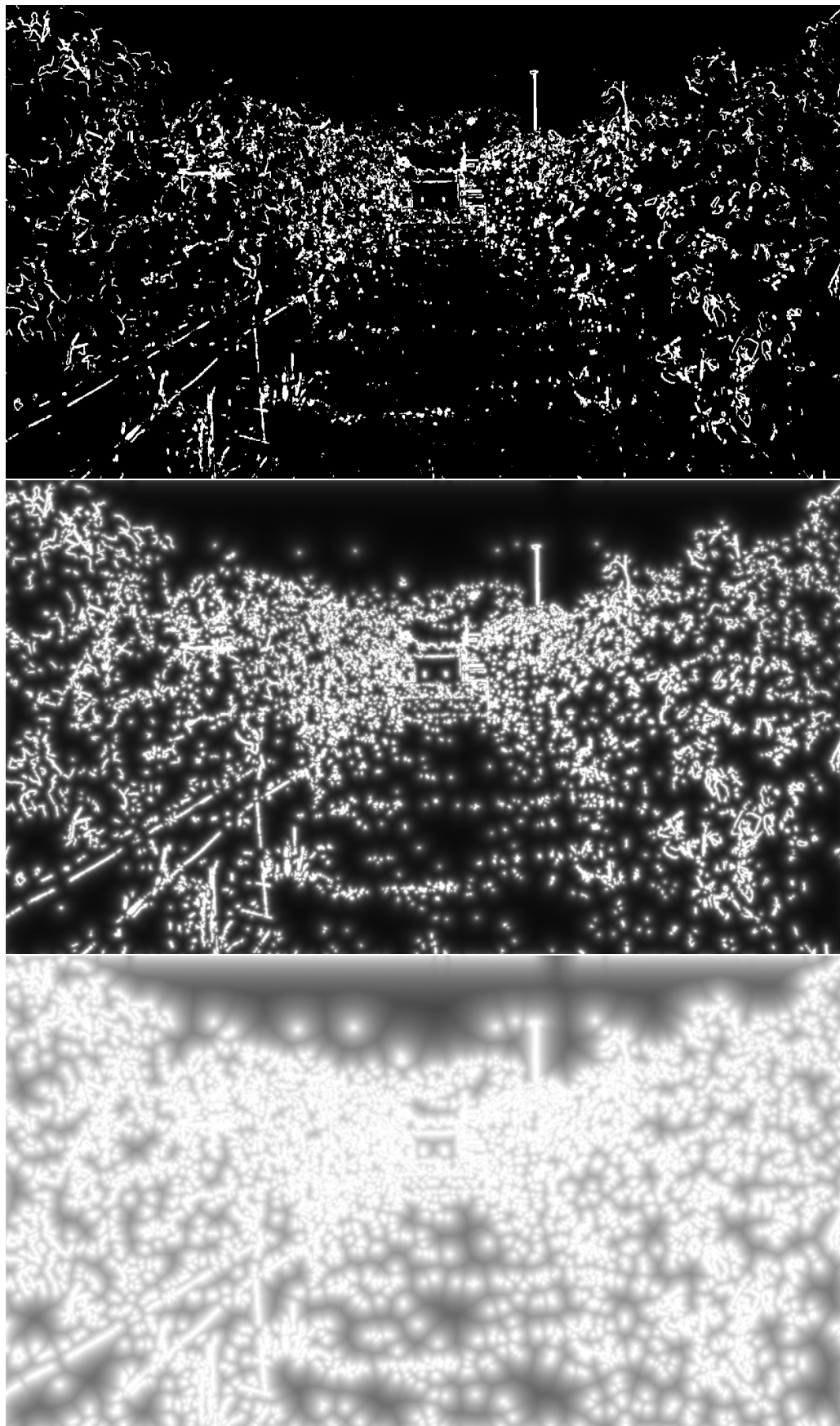
- Originální video pouze na binární bázi s inverzí.
- Potenciálová mapa s nízkým koeficientem ovlivnění  $K$ .
- Potenciálová mapa s vysokým koeficientem ovlivnění  $L$ .



Obrázek 7.1: Porovnání potenciálů v Trackmanii s originálem



Obrázek 7.2: Porovnání potenciálů z dronu v přírodě s originálem



Obrázek 7.3: Porovnání potenciálů z vinic s originálem



První viditelný vliv je již na originálním obrázku. Tím, že binární analýza obsahuje inverzi a škálování pro celý rozsah černobílého obrazu, pixely s hodnotou  $uint8 \in <0, 255>$ , jsou hrany a detaily o to viditelnější a kontrastnější než na původní binární analýze.

Ve druhém obrazu, ze tří porovnávaných, je implementován malý koeficient ovlivnění od překážek  $K$ . Tato malá úprava má za následek dva hlavní jevy:

1. Již při malé hodnotě koeficientu ovlivnění dochází k jemnému rozmazání obrazu. Příčina je ta, že přilehlé pixely k detekovaným překážkám také nabývají nenulové (= nečerné) hodnoty.
2. Toto ovlivnění malého okolí způsobuje zrobustnění, zvýraznění a spojení hran, které vytvoří obraz s *vytaženými* překážkami.

Při analýze sady třetích obrázků s koeficientem vysokého ovlivnění  $L$  dochází k jasnému zviditelnění potenciálových map, kdy detekované překážky a objekty značnou měrou zasahují a ovlivňují své okolí. Tento efekt je nejlépe vidět v Trackmanii, díky již dříve zvolenému neutrálnímu podloží, jak objekty doslova vyzařují svůj potenciálový vliv do okolí. Na videích z venkovního prostředí je tento efekt také pozorovatelný, nicméně kvůli většímu množství překážek v celém zorném poli již nemusí být efekt tak dobře zaznamenaný.

Příležitost, kterou tato analýza přináší, je možnost vyhledání, detekce a lokalizace černých (= tzn. prázdných) míst bez překážek, kudy by potenciálně mohl vést úhybný manévr.

Tato metoda potenciálových map dodává obrazu jistou hloubku, tak jak bylo cílem této kapitoly, nicméně využívá pouze dalšího matematického zpracování již získaného binárního obrazu. A právě proto další část této kapitoly se znova zaměří na informaci přímo obsaženou ve výpočtu optického toku.

## 7.2 RGB rozlišení

Dosavadní práce přinesla již kvalitní *vidění* robota a popisu zorného pole kamery za využití optického toku. Nicméně stále velká část informace z výpočtu a polárního výstupu: (*Velikost*, *Úhel*) nebyla využita, a tohoto se ujme tato kapitola. Ta jednak poskytne obrazu hloubku, ale také využije další sadu informací přímo obsažené ve výpočtu optického toku, přesněji v polárním výstupu z *Velikosti*.

Binární analýza staví jen na informaci je/není optický tok, kdežto při pohledu na *Velikost* v kombinaci s teorií se dá využít i hodnota tohoto toku. Jak je již známo z teorie, objekty blíže pozorovateli a objekty dále od středu obrazu se pohybují vyšší optickou rychlostí. Této znalosti by se potenciálně dalo využít k detekci klíčových překážek, které jsou například v kolizní trajektorii.

Myšlenka tohoto zpracování spočívá v rozdělení všech detekovaných pixelů na 3 části do tercílů a také na barevné oddělení rychlostních segmentů na malou, střední a velkou optickou rychlost. Přiřazení v této práci využívá intuitivního RGB spektra:

- Červená barva pro malé rychlosti v obraze.
- Zelená barva pro střední tercíl rychlostí.
- Modrá pro nejvyšší třetinu rychlostí.

Rozdělení bylo uděláno do tercilů, tak aby obraz byl procentuálně vyvážený, to znamená třetina pro každou barvu. Při navázání na absolutní hodnoty by tyto poměry byly nevyvážené, procentuální rozdělení zabrání dominanci jedné barvy. Tento přístup dodá výslednému obrazu jednak hloubku (neboli rychlostní rozšíření) ale zároveň ho přesune z černobílého do barevného spektra, což dále pomůže optickému vnímání.

Zpracování a implementace klíčových částí v Matlabu vypadá následovně:

```
%% Quantiles
NonZeroElements = flow.Magnitude(flow.Magnitude ~= 0);
Q_33 = quantile(NonZeroElements, 0.33);
Q_67 = quantile(NonZeroElements, 0.67);

%% RGB
R_idx = find(flow.Magnitude > 0 & flow.Magnitude <= Q_33);
G_idx = find(flow.Magnitude > Q_33 & flow.Magnitude <= Q_67);
B_idx = find(flow.Magnitude > Q_67);
R = uint8(zeros(720, 1280));      % MATRIX SIZE!
G = uint8(zeros(720, 1280));      % MATRIX SIZE!
B = uint8(zeros(720, 1280));      % MATRIX SIZE!
R(R_idx) = 255;
G(G_idx) = 255;
B(B_idx) = 255;
RGB = cat(3, R, G, B);

%% Histogram = Checking the thirds
binEdges = [0, 0.00000000001, Q_33, Q_67, 10000];
histCounts = histcounts(flow.Magnitude, binEdges);
```

Tato implementace spočívá nejdříve v nalezení hraničních hodnot mezi kvantily, značené jako:  $Q_{33}$  a  $Q_{67}$ , které rozdělí momentální snímek na 3 třetiny podle rychlostí.

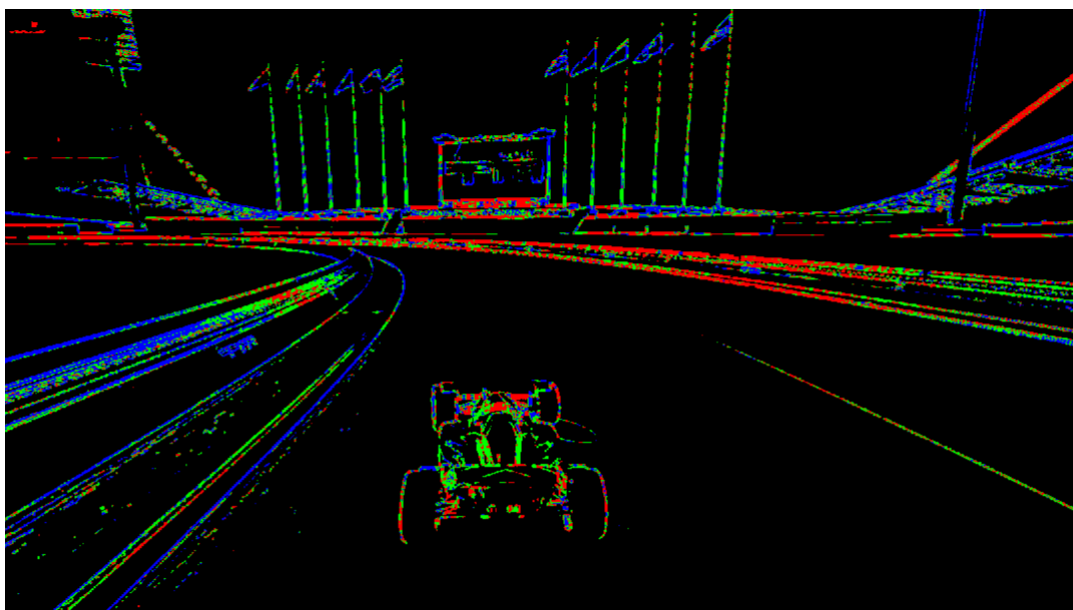
Dále je nutno přiřadit správné barvy odpovídajícím rychlostem, proto se naleznou a v celé matici přiřadí indexy těchto pixelů k žádané barvě. Následuje vytvoření černých matic pro každou barvu jako podklad, na který se následně přenesou barevná maska každé barvy a zvýrazní se na maximum jako  $uint8 = 255$ .

Nyní existují 3 separátní matice obsahující dedikované rychlostní spektrum dle odpovídající barvy. Příkazem  $cat()$  dojde ke spojení těchto tří matic a vytvoření výsledného RGB obrázku.

Histogram, jako poslední část, slouží k ověření správného rozdělení do třetin a obsahuje tyto 4 koše:

1. Nulový koš pro součet všech pixelů bez informace (= nulový optický tok)
2. Koš pro červenou barvu, spodní třetina (= nejnižší rychlosti)
3. Třetí koš pro zelenou barvu (= střední rychlosti)
4. Poslední koš pro modrou barvu (= nejvyšší rychlosti)

První ukázka výsledku tohoto zpracování z Trackmanie je vidět na Obrázku 7.4. Výsledek je opět stavěný na binárním základu, jenže tentokrát jsou navíc segmenty v obraze barevné a každá barva odpovídá své optické rychlosti.



Obrázek 7.4: RGB zpracování videa z Trackmanie

Koš	0	R	G	B
# pixelů	1869692	67288	69330	67290

Obrázek 7.5: Konkrétní poměr barevného rozdělení v obraze

Obrázek 7.5 zase demonstruje rozdělení, nebo také potvrzuje vybalancování, pixelů v obraze. Dominantní je v tomto případě černá barva, což znamená nulový optický tok, a proto nulový koš obsahuje nejvyšší počet těchto pixelů, okolo 1,87 milionů.

Nicméně poměry mezi barevnými tercily jsou vyrovnané na třetiny (s lehkou odchylkou jelikož přesný poměr je 33:34:33, zhruba 70 tisíc pixelů pro každou barvu), což přináší balanc a zabraňuje dominanci.

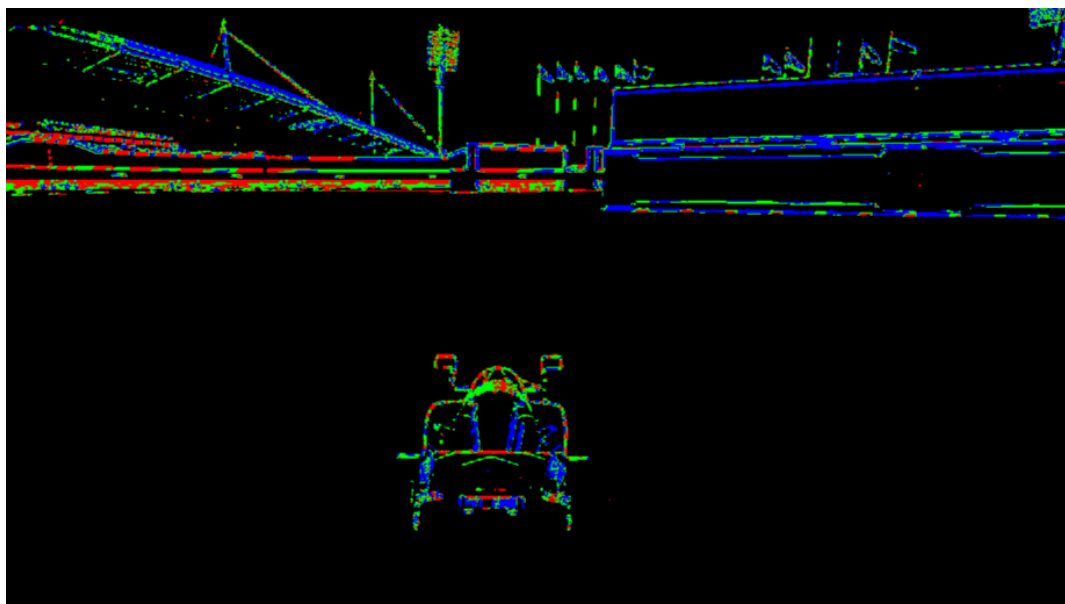
### 7.2.1 Přínos RGB zpracování

RGB způsob zpracování přináší do obrazu jak hloubku, tak hlavně rozlišení překážek. Tento efekt je demonstrován při porovnání Obrázku 7.6 a Obrázku 7.7. Na původním

snímku lze jasně rozeznat bližší černou překážku napravo a vzdálenější struktury na středu a nalevo. RGB snímek dokáže barevně odlišit tyto struktury, to znamená, bližší a opticky rychlejší zeď napravo, je správně označena dominantně modrou barvou. Naopak vzdálenější struktury, které by se na základě teorie měly pohybovat opticky pomaleji vůči pozorovateli, jsou označeny dominantně červenou barvou.



Obrázek 7.6: Originální video z Trackmanie



Obrázek 7.7: RGB zpracování Trackmanie

Je ovšem také samozřejmostí uznat nedostatky této metody, kdy například v pozadí lze pozorovat některé struktury označeny modře i přesto, že by měly být k pozorovateli více statické právě kvůli jejich vzdálenosti.

Důvody pro tyto nedokonalosti se nabízejí dva: Prvním je možný okamžitý pohyb

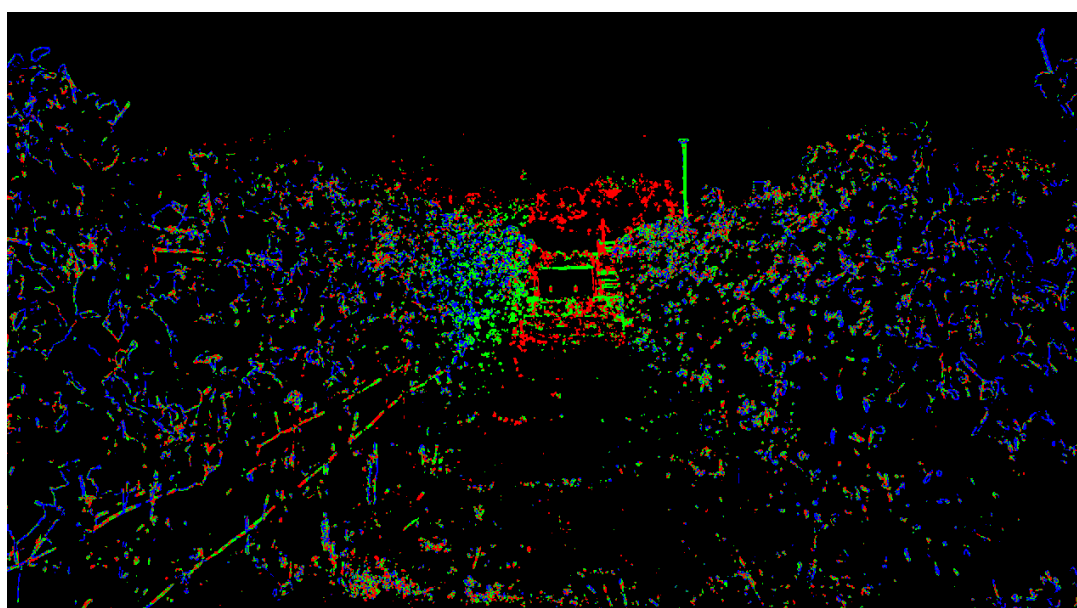
kamery a scény zároveň, jelikož se jedná pouze o výsek videa a tak se může stát, že v tomto okamžiku dochází i k relativnímu pohybu celé scenerie.

Druhým důvodem může být zvýšená citlivost struktur, které se nachází v neutrálním prostředí. Například vlajky na modrém nebi mohou být snáze a lépe detekovatelné s větší přesností, která jim potenciálně připisuje vyšší optickou rychlost.

Toto bylo shrnutí pozorování a zpracování ve virtuálním světě Trackmanie. Klíčové je opět chování a aplikovatelnost v reálném prostředí. Následující dvě dvojice snímků (vždy nejdříve originál a poté porovnání se zpracovaným obrazem) budou ukazovat reálné výsledky z venkovního prostředí.



Obrázek 7.8: Originální video z vinic

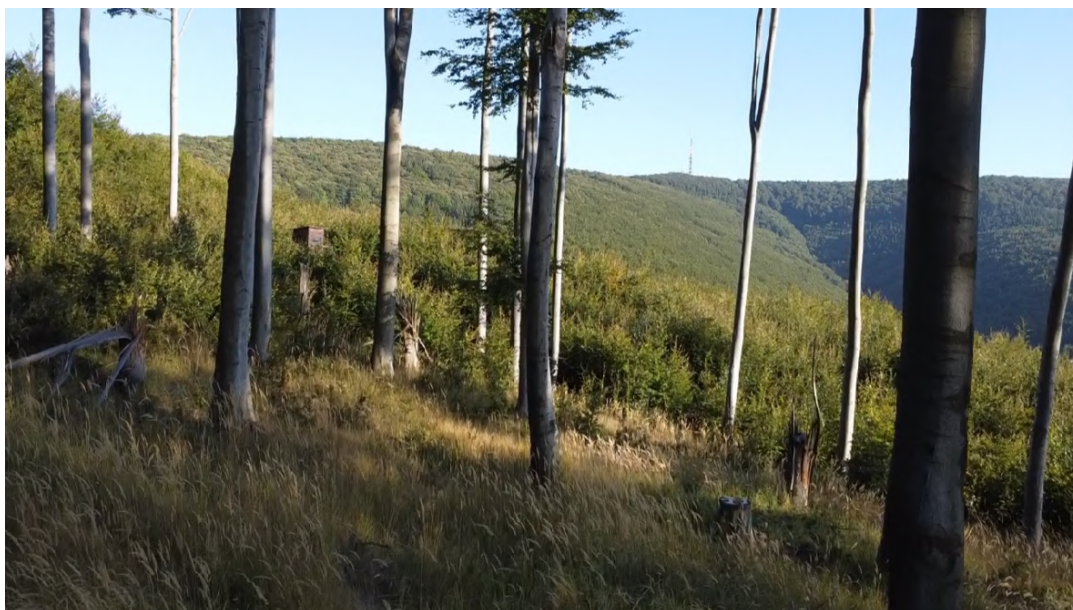


Obrázek 7.9: RGB zpracování vinic

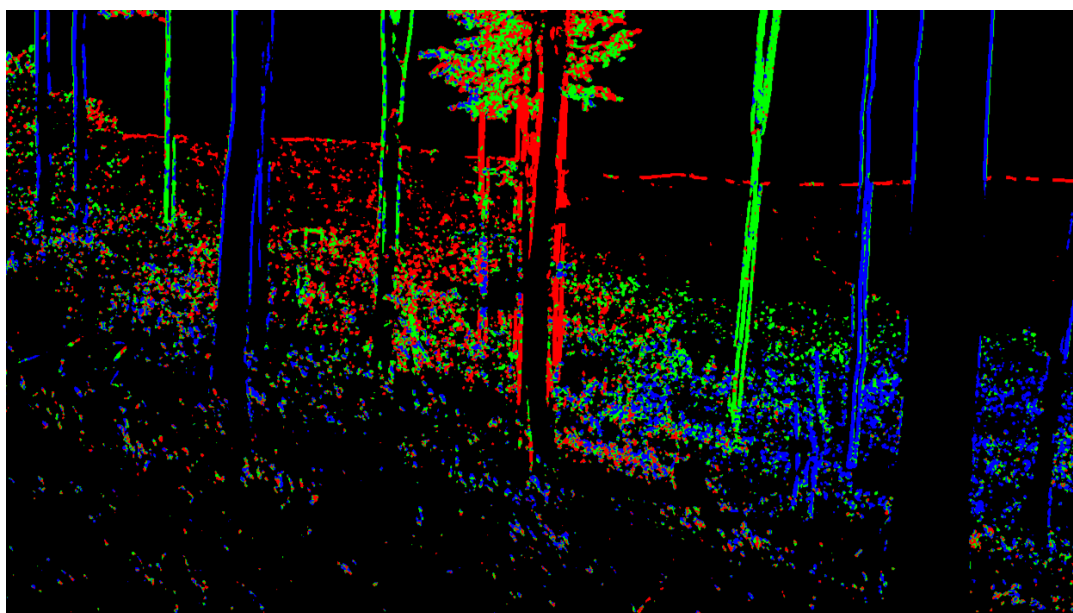
První dvojice výše je z videa z vinic (Obrázek 7.8 a Obrázek 7.9). Toto prostředí je typické svým spojitým charakterem bez výraznějších samostatných struktur.

Teorie o optickém toku říká, že struktury blíže a na krajích se opticky pohybují rychleji, čemuž odpovídá jejich zbarvení do modra. Naopak směrem na střed a dále od pozorovatele dochází k pozvolnému přechodu do zelené, střední rychlosti, až červené, znamenající nejmenší rychlost optického pohybu.

Druhá dvojice níže (Obrázek 7.10 a Obrázek 7.11) také úspěšně potvrzuje tuto teorii v pozadí, tentokrát ovšem na diskrétním prostředí stromů.



Obrázek 7.10: Originální video z dronu



Obrázek 7.11: RGB zpracování videa z dronu

Prostřední strom, jako potencionálně kritická překážka, se nepohybuje opticky vlevo ani vpravo, právě proto, že je uprostřed, a je tedy správně vyhodnocen červenou barvou. Naopak stromy lehce nalevo a napravo jsou zelené jak se směrem do stran optický pohyb zrychluje, což znamená vyšší rychlosti. A na okrajích lze očekávat v obraze relativní nejrychlejší pohyby a tím pádem i stromy označené modrou barvou.

Za zmínku stojí i jeden speciální případ stromu na Obrázku 7.11. Na středu je jeden výrazný červený strom, kousek nalevo je další zelený a vedle něj nalevo je další modrý. Dle teorie by se dalo očekávat, že za modrým dále na kraj již bude jen další modrá. Není tomu tak. Následuje zase jeden zelený strom a až poté zbytek modrých. A právě tento zelený strom je předmětem diskuze.

Zdánlivě tento nesoulad narušuje teorii v pozadí, nýbrž není tomu tak, a naopak ukazuje další důležitý faktor a tím je kýžená *hloubka* obrazu. V originálním zobrazení na Obrázku 7.10 je možno tento strom lokalizovat jako bílý strom mnohem dále než zbytek stromů. A přesně toto je důvod jeho označení zelenou, střední barvou. Ano, optický tok se zrychluje více a více tím, jak je překážka blíže k okrajům zorného pole. Nicméně, proti tomuto aktu hraje roli to, že optický tok se naopak zpomaluje s tím, jak je předmět opticky dále od pozorovatele. Toto jsou dva klíčové vlivy, které v tomto způsobu zpracování mohou i leckdy působit proti sobě a vytvořit přesně takovou situaci jakou je možno pozorovat na Obrázku 7.11.

### 7.2.2 Efekt clustrování

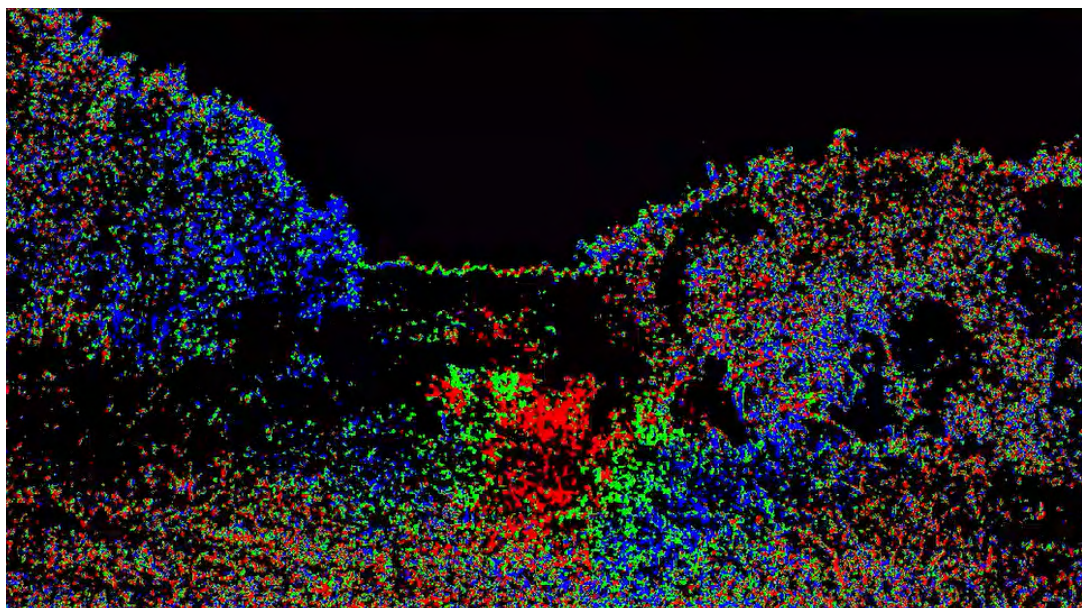
Za zmínku stojí ještě jeden efekt, který je možno pozorovat při tomto způsobu zpracování. Tento efekt je pro potřeby této práce nazván jako efekt *Clustrování*. Je pozorovatelný již na Obrázku 7.9 a jedná se o efekt sovisející s pojmem z teoretické části, a to konkrétně s *bodem expanze* nebo v originále z literatury také jako *FoE - Focus of Expansion*, známý jako zřídlový bod, z něž vycházejí vektory optického toku.

Pro pozorování tohoto efektu je vhodné spojitě prostředí, bez výraznějších struktur. Daný obraz (nejlépe video) je nutno vnímat s odstupem jako jeden celek a zaměřit se na viditelné clustry, v tomto případě nejlépe červený, jelikož ten vždy drží pospolu.

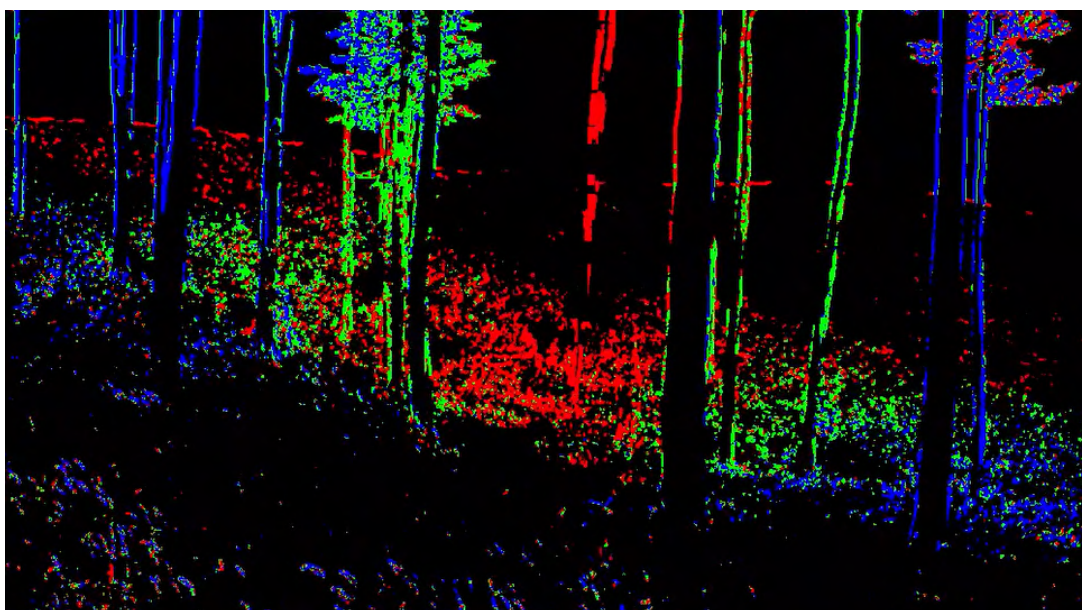
Tento červený cluster značící nejmenší rychlosti v obraze by mohl ukazovat, kde se ve videu může nacházet *FoE*. Červený cluster bývá poté *obalen* zeleným clustrem, jenž následně přechází do modrého.

Pohyb tohoto clustru lze také navázat s tím, jakým směrem se zrovna robot/kamera pohybuje a tak je možno pozorovat jeho plynulé přesouvání po obraze. Pokud je pohyb velmi rychlý, zejména rotačního charakteru, může se také stát, že naprosto zmizí z obrazu, jelikož v tomto případě by se *FoE* nacházelo také mimo obraz.

Názornou ukázkou tohoto jevu lze vidět na Obrázku 7.12, kde se cluster nachází zhruba uprostřed obrazu, kolem něj je viditelná vyšší koncentrace zelené, jenž přechází do modré. Na druhém příkladu na Obrázku 7.13 z dronu, lze pozorovat podobné chování. Pro plnou představu a vizualizaci tohoto efektu je nutno shlédnout video z příloh k tomuto tématu určené.



Obrázek 7.12: Efekt Clustrování v poli



Obrázek 7.13: Efekt Clustrování u dronu v lese



### 7.2.3 Vliv světla

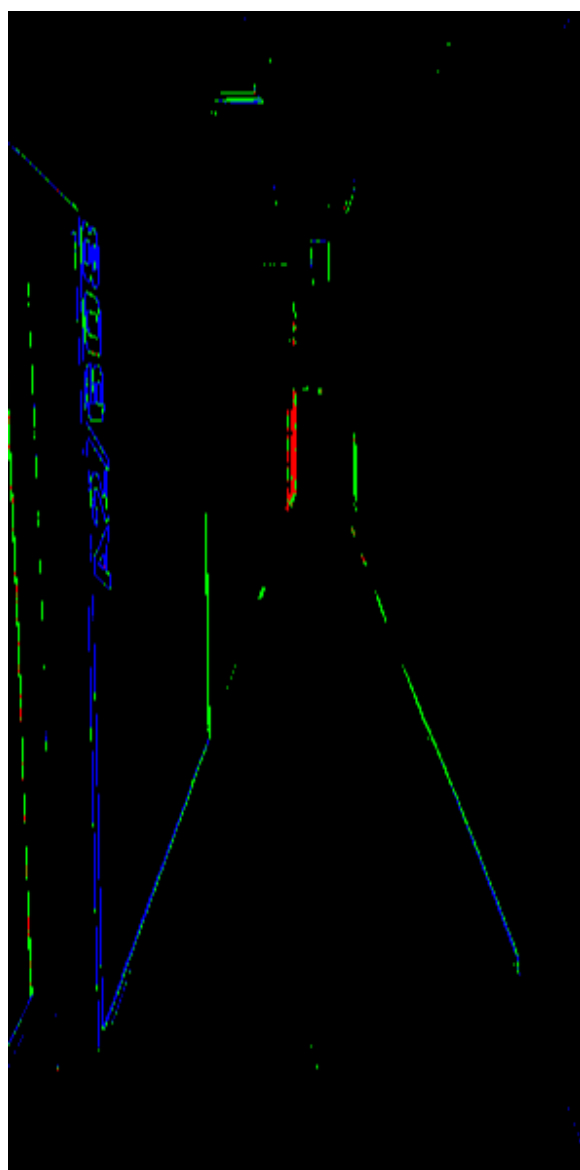
Již z názvu této metody: *Optický tok*, lze poznat, že se jedná o metodu pracující ve viditelném spektru elektromagnetického záření. Nutnou součástí této práce tak musí být i prozkoumání problematiky spojené s tímto omezením, a tu nejlépe vystihuje následující otázka:

- *Jak ovlivňuje intenzita osvětlení scény zachycení detailů a kvalitu zpracování?*

Pro tento experiment bylo vybráno prostředí, kde je možno jednoduše nasimulovat změnu osvětlení. Tímto vhodným kandidátem byla chodba budovy, kde jednou bylo světlo vypnuto (Obrázek 7.14 - originální video a Obrázek 7.15 - RGB analýza) a jednou zapnuto (Obrázek 7.16 - originál se světlem a Obrázek 7.17 - RGB analýza).



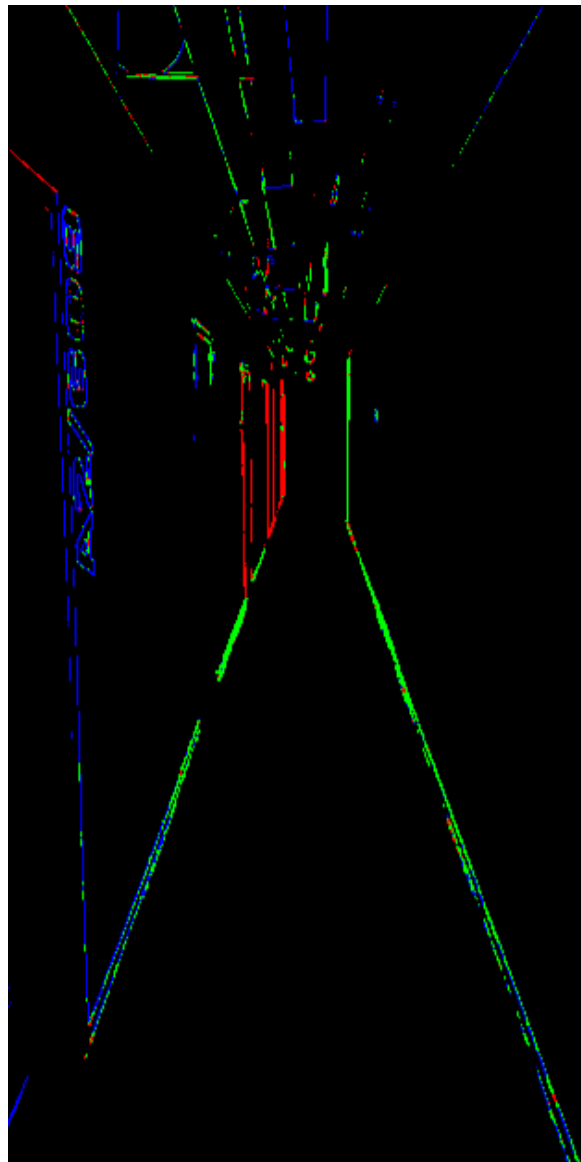
Obrázek 7.14: Chodba bez světla



Obrázek 7.15: RGB chodba bez světla



Obrázek 7.16: Chodba se světlem



Obrázek 7.17: RGB chodba se světlem

Jak je možno pozorovat při porovnání obrázků s RGB zpracováním, role intenzity osvětlení je významná. Samozřejmě tento efekt byl očekávaný, pakliže by se vzal do extrému a udělal se rozbor černého videa v absolutní tmě, ani tento rozbor by nepřinesl zázrak a vrátil opět černé video.

Toto je možná slabina této metody, kdy například taková GPS není vázána na denní dobu, zde by mohlo docházet k problémům při implementaci za zhoršených vizuálních podmínek, tzn. například noc, déšť, atp.

# 8 Úhlová informace

Pro kompletaci a úplné využití všech informací z optického toku je nutno se také věnovat dosud opomíjenému *úhlu*. Zvolené polární zpracování ve formě (*Velikost, Úhel*) má tu výhodu, že tento výstup dává dva odlišné parametry, se kterými je možno relativně nezávisle pracovat a extrahovat odlišné funkcionality.

To je hlavní důvod a výhoda proč byl tento přístup upřednostněn pro tuto práci, nežli pracovat s kartézským výstupem ve formě ( $V_x, V_y$ ), jenž sice obsahuje tu stejnou celkovou rychlost, jenže pro následnou práci dávají *pouze* ten stejný typ informace pro dva různé směry v totožné podobě.

V polárním zpracování může *úhel*, jako výstup z Matlabu, nabývat hodnoty mezi  $(-\pi, \pi)$ . Pro jasnou definici jsou stanoveny tyto 4 hodnoty na základě Obrázku 5.2.

1. Hodnota  $\acute{U}hel = 0$  znamená směřování vektoru úhlu v souladu s kladným směřováním osy  $x$ .
2. Hodnota  $\acute{U}hel = \pi \Leftrightarrow \acute{U}hel = -\pi$  znamená směřování vektoru úhlu do protisměru osy  $x$ .
3. Hodnota  $\acute{U}hel = \pi/2$  znamená směřování vektoru úhlu v souladu s kladným směřováním osy  $y$ .
4. Hodnota  $\acute{U}hel = -\pi/2$  znamená směřování vektoru úhlu do protisměru osy  $y$ .

## 8.1 Definice hypotézy

Pokud se tedy spojí poznatky z teorie s očekávaným výstupem zpracování obrazu mělo by tedy dojít k následujícím třem možnostem v obraze:

1. Při přímém pohybu vpřed se scéna *rozbaluje* a odvíjí od  $FoE$ , které se nachází zhruba uprostřed obrazu, což znamená, že by mělo dojít k odlišení částí v obrazu, které se opticky pohybují vlevo, od těch, které se opticky pohybují vpravo
2. Při rotaci vlevo se celé scéna pohybuje vpravo, to znamená, že by mělo převažovat dominantní zobrazení pohybu vpravo
3. Naopak při rotaci vpravo se celá scéna pohybuje vlevo a mělo by dojít k jasné obrazové dominanci pohybu vlevo.

Předtím, než bude možné ukázat, zda-li je tato hypotéza správná a potvrzuje tak teorii, je nutné ukázat a vysvětlit implementaci v kódu, k čemuž slouží následující kapitola 8.2.

## 8.2 Implementace

Výňatek klíčových pasáží z kódu je zobrazen níže a ukazuje důležité kroky při práci s úhlovou informací ve videu.

```

%% Base matrix
blanc = uint8(zeros(720, 1280));
ZeroIndeces = find(flow.Magnitude == 0);
NonZeroIndeces = find(flow.Magnitude ~= 0);

%% Angles
K = 81.15;          % Scaling coefficient to uint8 (0 - 255)
AbsoluteAngles = uint8(abs(flow.Orientation) * K); % Angles adjustment

Direction = AbsoluteAngles(NonZeroIndeces) > 255/2; % Condition
FacingLeft = find(Direction == 1); % Check condition
FacingRight = find(Direction == 0); % Check condition

LeftIndeces = NonZeroIndeces(FacingLeft); % Left = Red
RightIndeces = NonZeroIndeces(FacingRight); % Right = Green

%% RGB
R_idx = LeftIndeces; % Assign index left
G_idx = RightIndeces; % Assign index right
R = uint8(zeros(720, 1280)); % MATRIX SIZE!
G = uint8(zeros(720, 1280)); % MATRIX SIZE!
B = uint8(zeros(720, 1280)); % MATRIX SIZE!
R(R_idx) = 255;
G(G_idx) = 255;
RGB = cat(3, R, G, B); % Only RG valid, B = 0

```

Nejdříve je nutno opět vytvořit podkladovou matici. Následně metoda také pracuje pouze s pixely, jež mají nějakou informaci, ovšem je nutno zde dávat pozor, jelikož to, zda-li pixel obsahuje informaci je nutno soudit dle *Velikosti*, jestli ta je nenulová. Důvod je ten, že pokud je  $\text{Úhel} = 0$ , tak to není jako u *Velikosti*, že pixel neobsahuje žádnou využitelnou informaci. Ba naopak, v tomto specifickém případě  $\text{Úhel} = 0$  je kýžená informace o směru, která nesmí být ztracena.

V další části kódu je nutno se zamyslet nad budoucí aplikací, ta je pro robot v poli, který se má potenciálně vyhnout překážce. Takovýto robot má možnosti jen dvě: vlevo nebo vpravo. Možnost nahoru nadskočit nebo se podhrabat zde není uvažována. Přesně proto je možné tuto úlohu zjednodušit a zajímat se pouze o průmět úhlu do osy  $x$ . V

takovém případě je možné vzít absolutní hodnotu úhlu a pouze zjistit zda je nebo není větší než  $\pi/2$ .

Nachází se zde ještě jedna konstanta  $K = 81,15$ . Ta se stará o to aby byl později využit celý rozsah  $uint8 \in \langle 0, 255 \rangle$ . Tímto krokem se otázka rozdělení pohybu mění v otázku zda-li je daný úhel větší nebo menší než  $255/2 = 127,5$ .

Dané pixely se roztrídí, zda-li označují pohyb vlevo nebo vpravo a dále přichází na řadu opět barevné zobrazení. Použije se stejná metoda jako výše v RGB přístupu, nicméně tentokrát budou pro dva směry stačit pouze dvě barvy.

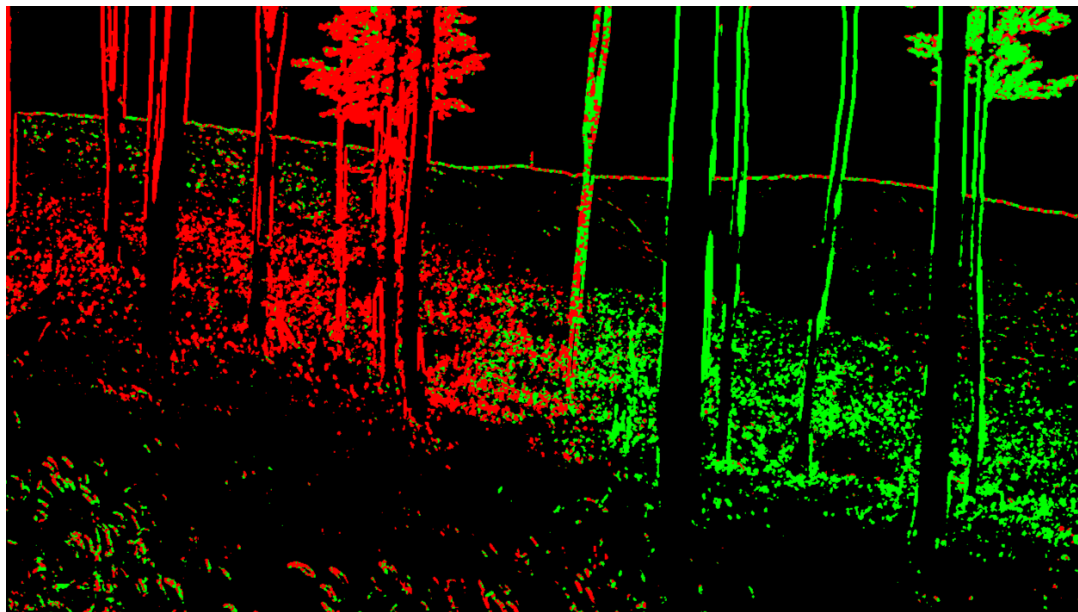
- Červená barva značící pohyb vlevo.
- Zelená barva značící pohyb vpravo.

Poté se vytvoří tyto barevné masky (modrá zůstane nulová) a následně se spojí a vytvoří jeden barevný obraz.

### 8.3 Ověření

Z výše uvedeného kódu a hypotézy v kapitole 8.1 je tedy teoreticky možno aby nastaly tři různé situace.

Při přímém pohybu vpřed by mělo dojít k rozdělení obrazu vesměs na dvě stejné poloviny, kdy ta levá se bude pohybovat opticky vlevo a bude tedy označena červeně. Druhá, pravá polovina, by měla opticky směřovat vpravo a tím pádem mít zelené zabarvení. To, že tomu takto opravdu je, lze pozorovat na následujícím Obrázku 8.1 z přímého pohybu robota v lese.



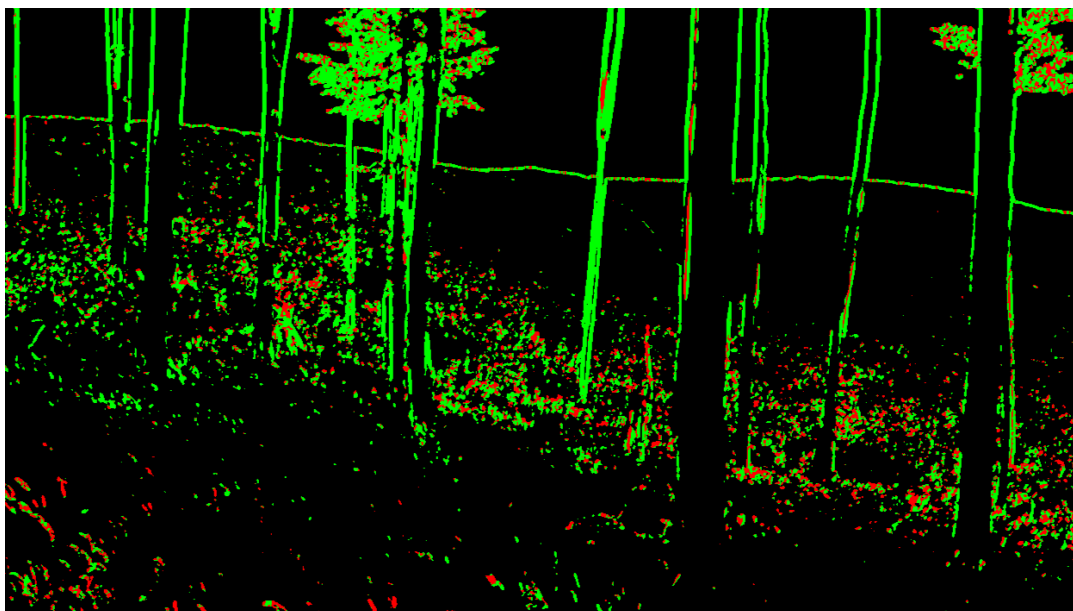
Obrázek 8.1: Úhlová analýza pohybu vpřed

Je také možno si povšimnout zajímavého úkazu stromu přesně uprostřed, který se nachází v kolizním směru a je tedy označen zhruba půl na půl červenou a zelenou barvou.

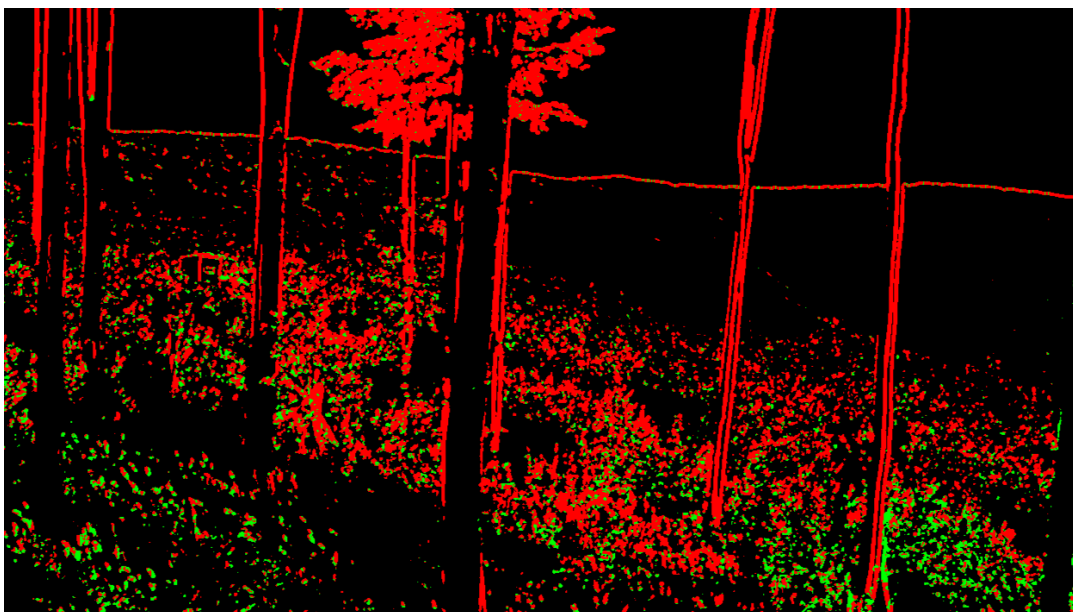
Samozřejmě metoda není naprosto bezchybná a stoprocentní a pro hůře rozeznatelné předměty jako křoví a trávy v dále je možno pozorovat nahodilě pixely označené špatnou

barvou. Nicméně pro dominantní překážky jako stromy je detekce téměř bezchybná.

Další dvojice příkladů bude demonstrovat rotaci. Na Obrázku 8.2 je vidět celožabarové zabarvení do zelené, což znamená, že celý obraz se pohybuje vpravo. Naopak Obrázek 8.3 ukazuje rotaci kamery vpravo a zabarvení obrazu do červené, což znamená pohyb celé scény vlevo.

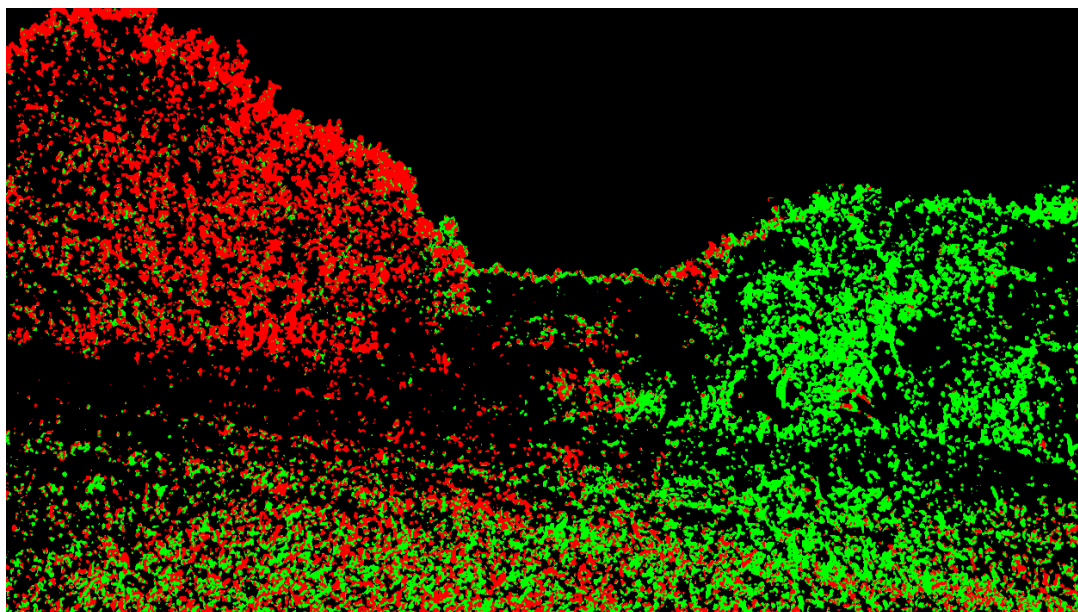


Obrázek 8.2: Úhlová analýza při rotaci vlevo



Obrázek 8.3: Úhlová analýza při rotaci vpravo

Pro kontrolu lze přidat ještě analýzu z jiného videa, například se lze podívat na původní video z obyčejné louky.



Obrázek 8.4: Úhlová analýza pohybu vpřed na videu z louky

Opět lze pozorovat rozdělení zhruba půl na půl. Ovšem Obrázek 8.4 pochází z videa z louky. Toto video je specifické v tom, že rotační pohyb a dopředný pohyb jsou zhruba stejně rychlé, proto lze na zpracovaném videu (z obrázku toto nelze poznat) pozorovat *přelévání* těchto barev, kdy rozdělení místy není půl na půl, nýbrž například dvě třetiny pro jednu a jedna třetina pro druhou barvu. Toto souvisí s tím jak se mísí a sčítají dva různé pohyby, dopředný + rotace.

Těmito čtyřmi příklady jsou všechny 3 body hypotézy z kapitoly 8.1 potvrzeny. Zpracování úhlové informace funguje tak jak bylo předpokládáno na základě teorie a zvoleného postupu. Samozřejmě, pro plné přesvědčení je nutno shlédnout videa.

# 9 Rekapitulace metod

Na závěr praktické části je zde shrnutí a ukázka jak se práce vyvíjela a také přehledně zrekapitulovány dosažené výsledky nalezených metody. Dále jsou zde také vyzdvíženy v čem byly přínosem, kde mají své silné stránky a kde naopak zaostávaly, což bylo následnou příčinou dalšího bádání. V následujících dvou podkapitolách budou zvýrazněny dvě větve tohoto přehledu.

1. Podkapitola 9.1 ukáže a popíše schéma vývoje tak jak se práce odvíjela a vylepšení plynuly z předchozích bádání a spojovaly dosažené poznatky.
2. Podkapitola 9.2 přehledně zesumarizuje výsledky a vlastnosti klíčových metod zkoumání.

## 9.1 Evoluce práce

Tato diplomová práce byla velmi explorativního charakteru. Počáteční cíl byl schválně postaven velice široce a to ještě bez jasné vize jak k němu dojít. Práce se odvíjela v mnoha iteracích jdoucích po sobě, kdy bylo dosaženo parciálního výsledku a poté se zpětně hodnotila jeho využitelnost a upravovala cesta dalšího bádání.

*High level* schéma pracovního postupu, návazností a vývoje práce je vyobrazené na Obrázku 9.1 níže.

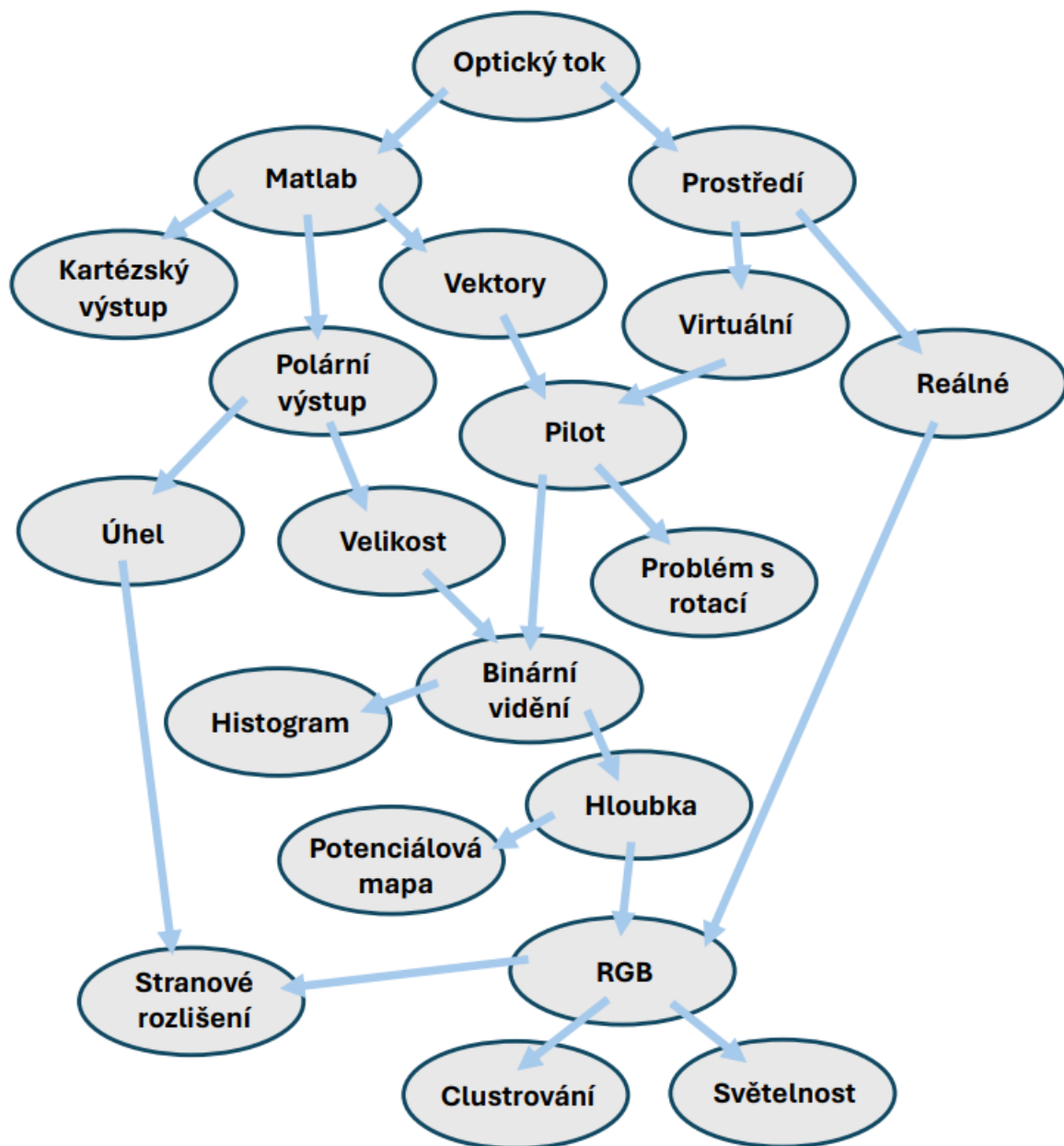
Na počátku nahoře stála hlavní myšlenka analýzy optického toku, ten se analyzoval v prostředí Matlabu, z čehož vzešlo několik možností výstupu. Paralelně k tomuto se prozkoumávala větev potenciální aplikace a vývojového prostředí (virtuální a reálné).

Poté ve spojení virtuálního prostředí a specifického výstupu vznikly pilotní analýzy, ty ukázaly na velké nedostatky a problémy, které se záhy staly hnacím motorem pro myšlenku binárního vidění.

Následně na bázi binárního vidění bylo provedeno několik dalších bádání a vylepšení, z nichž nejsignifikantnější bylo barevné a rychlostní rozšíření pojmenováno jako RGB analýza. Tato analýza se spojila s reálným (venkovním) prostředím a umožnila dokonce využití i starších, a dříve problematických výstupů, jako například úhlové informace.

Závěrem byl prostor ještě na specifické pozorování některých speciálních příkladů jako například vyzdvihnutí efektu *clustrování* ve spojitosti s *FoE* a nebo taky prověření jak světelnost ovlivňuje kvalitu dosažených výsledků.





Obrázek 9.1: Schéma vývoje zkoumání metod na základě optického toku

## 9.2 Hodnocení metod

Jako poslední přichází na řadu přehledné shrnutí hlavních použitých metod. Toto srovnání níže ukazuje metody společně s jejich silnými stránkami a nedostatky.

Tabulka 9.1 slouží k přehlednému výčtu kvalitativních parametrů jednotlivých metod. Leckdy to, co bylo negativem metody předchozí, se objevuje jako silná stránka metody následující, jelikož potenciální problém byl hlavním předmětem následného zkoumání.

Tabulka 9.1: Hodnocení metod

Metoda	Silné stránky a pozitiva	Negativa a nedostatky
<b>Optický tok s šipkami</b>	Vizualizace zpracování videa Představa o principu fungování a chování Jednoduchá parametrizace	Těžko ověřitelná správnost Zobrazení pouze omezeného počtu vektorů Pomalá vizualizace vyššího množství vektorů Zkreslená rotací
<b>Binární analýza</b>	Ukazuje vidění robota Nezatížena na rotaci Informace získaná pouze z optického toku Funguje v testovacím i venkovním prostředí Rychlé zpracování videa	Nemá hloubku Náročná na světelné podmínky
<b>Potenciálové pole</b>	Dodává představu o hloubce obrazu Možnost kombinovat s dalšími metodami Spojuje překážky Vhodná pro ideální prostředí	Rozmazává obraz Není ideální venku
<b>RGB rozšíření</b>	Stejně kvality jako binární analýza velikosti Není zatížena rotací Dodává obrazu hloubku Rozlišuje rychlosti v obraze Funguje v testovacím i reálném prostředí	Jedna rychlost pro více možností Náročná na světelné podmínky
<b>Úhlová analýza</b>	Jednoduché ověření správnosti fungování Využívá informaci o úhlu	Specifické chování při rotaci

# 10 Závěr

Tato diplomová práce se zabývala návrhem a implementací metody určené pro navigaci mobilních robotů ve volném venkovním prostředí. Všechny tři hlavní cíle práce: specifikace prostředí, ve kterém se robot pohybuje (kapitola 2), výběr a popis metody vyhýbání se překážkám (kapitola 3) a návrh implementace této metody (kapitoly 4 až 8), byly úspěšně adresovány.

Teoretická část byla věnována zejména popisu venkovního prostředí tzv. *Field robotics*. Dále je zahrnut i popis virtuálního prostředí Trackmanie, jež byla využita pro prvotní experimenty, které mely ověřit vhodnost metody pro další zpracování ve složitějších podmínkách reálného prostředí.

Jako vybraná metoda pro úkol vyhýbání se překážkám byl zvolen Optický tok. Práce obsahuje matematické odvození, ukázkou řešení a aplikovatelnost metody, spolu s jasně definovanými limity použitelnosti. Závěrem praktické části je teoretický návrh zařazení optického toku do heterogenního navigačního systému jako doplněk k současně využívaným technologiím GPS a Lidaru.

Praktická část se poté zaměřuje na detailní popis vývoje zpracování obrazu pomocí optického toku. Jako první je prozkoumáno zpracování obrazu v prostředí *Matlabu* za použití *Computer vision toolboxu*. Rozebrány a popsány jsou matematické a obrazové výstupy, jež jsou završeny pilotní analýzou obrazu na videích z virtuálního a také reálného venkovního prostředí.

Práce poté postupuje směrem ke zjednodušení komplexnosti a zobrazení *vidění robota*, čehož výsledkem je černobílá *Binární analýza obrazu*. Princip *binárního vidění robota* je poté páteří pro následující kroky, které zase zpětně představují další úroveň komplexity, jako je přidání hloubky do obrazu. Pro hloubkovou interpretaci obrazu byly použity dvě metody: Potenciálové mapy a RGB analýza, jež barevně na třetiny segreguje části obrazu s malou, střední a vysokou optickou rychlostí.

Jako poslední je využita informace o úhlu, která do obrazu přidává stranové rozlišení a potvrzení o okamžitém směřování/rotaci robota. Závěrem jsou ve vývojovém schématu na Obrázku 9.1 ukázány jednotlivé kroky a propojení mezi jednotlivými iteracemi, načež navazuje přehledné shrnutí všech prozkoumaných metod s jejich silnými stránkami a nedostatky v Tabulce 9.1.

Veškeré výsledky využívají pouze obrazu z jedné kamery čímž umožňují použití této metody na širokém spektru zařízení. To znamená, že jediná nutná podmínka implementace je osazení robota kamerovým systémem. Směřování návazné práce by se mohlo udávat několika směry: za prvé je nyní otevřená cesta pro zamýšlení se nad ovládním a zejména řídicí logikou na základě poskytnutých výstupů. Druhá možnost je jít ve vývoji dále a adresovat identifikované nedostatky jednotlivých metod. Poslední možnost je samotná implementace a testování na reálném mobilním robotovi.

# Bibliografie

- [1] SOLEM, Jan Erik. *Programming Computer Vision with Python: Tools and algorithms for analyzing images*. "O'Reilly Media, Inc.", 2012.
- [2] BALÁTĚ, Jaroslav. *Technické prostředky automatického řízení*. Státní nakladatelství technické literatury, 1986.
- [3] FOUNTAS, Spyros; MYLONAS, Nikos; MALOUNAS, Ioannis; RODIAS, Efthymios; HELLMANN SANTOS, Christoph; PEKKERIET, Erik. Agricultural robotics for field operations. *Sensors*. 2020, roč. 20, č. 9, s. 2672.
- [4] GONZALEZ-DE-SANTOS, Pablo; FERNÁNDEZ, Roemi; SEPÚLVEDA, Delia; NAVAS, Eduardo; EMMI, Luis; ARMADA, Manuel. Field robots for intelligent farms—inhering features from industry. *Agronomy*. 2020, roč. 10, č. 11, s. 1638.
- [5] THORPE, Chuck; DURRANT-WHYTE, Hugh. Field robots. In: *Robotics Research: The Tenth International Symposium*. Springer, 2003, s. 329–340.
- [6] XU, Guochang; XU, Yan. *GPS*. Springer, 2007.
- [7] RAHMAN, Mohammed Ziaur. Beyond trilateration: GPS positioning geometry and analytical accuracy. *Global Navigation Satellite Systems: Signal, Theory and Applications*. 2012, s. 241–256.
- [8] RAHIMAN, Wan; ZAINAL, Zafariq. An overview of development GPS navigation for autonomous car. In: *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2013, s. 1112–1118.
- [9] PURKIS, Samuel J; BROCK, John C. LiDAR overview. In: *Coral Reef Remote Sensing: A Guide for Mapping, Monitoring and Management*. Springer, 2013, s. 115–143.

- [10] MALAVAZI, Flavio BP; GUYONNEAU, Remy; FASQUEL, Jean-Baptiste; LAGRANGE, Sebastien; MERCIER, Franck. LiDAR-only based navigation algorithm for an autonomous agricultural robot. *Computers and electronics in agriculture*. 2018, roč. 154, s. 71–79.
- [11] BELKIN, Ilya; ABRAMENKO, Alexander; YUDIN, Dmitry. Real-time lidar-based localization of mobile ground robot. *Procedia Computer Science*. 2021, roč. 186, s. 440–448.
- [12] DERNONCOURT, Franck. TrackMania is NP-complete. *arXiv preprint arXiv:1411.5765*. 2014.
- [13] NEINDERS, LJ. *Improving Trackmania Reinforcement Learning Performance: A Comparison of Sophy and Trackmania AI*. 2023. B.S. thesis. University of Twente.
- [14] KRIEMLER, Pius. Distributed Setup for Trackmania Reinforcement Learning. 2023.
- [15] BEAUCHEMIN, Steven S.; BARRON, John L. The computation of optical flow. *ACM computing surveys (CSUR)*. 1995, roč. 27, č. 3, s. 433–466.
- [16] SOUHILA, Kahlouche; KARIM, Achour. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*. 2007, roč. 4, č. 1, s. 2.
- [17] BURTON, Andrew; RADFORD, John. *Thinking in perspective: critical essays in the study of thought processes*. Taylor & Francis, 2022.
- [18] GIBSON, James J. The perception of the visual world. 1950.
- [19] ROYDEN, Constance S; MOORE, Kathleen D. Use of speed cues in the detection of moving objects by moving observers. *Vision research*. 2012, roč. 59, s. 17–24.
- [20] GAUR, Vedant. Lucas-Kanade Optical Flow Machine Learning Implementations. *Journal of Student Research*. 2022, roč. 11, č. 3.
- [21] HORN, Berthold KP; SCHUNCK, Brian G. Determining optical flow. *Artificial intelligence*. 1981, roč. 17, č. 1-3, s. 185–203.
- [22] O'DONOVAN, Peter. Optical flow: Techniques and applications. *International Journal of Computer Vision*. 2005, roč. 1, s. 26.
- [23] SHARMIN, Nusrat; BRAD, Remus. Optimal filter estimation for Lucas-Kanade optical flow. *Sensors*. 2012, roč. 12, č. 9, s. 12694–12709.

- [24] FAJMON, Břetislav; FEKT, UMAT. 3 Metoda nejmenších čtvercu 3 Metoda nejmenších čtvercu. [B.r.].
- [25] AGARWAL, Anshuman; GUPTA, Shivam; SINGH, Dushyant Kumar. Review of optical flow technique for moving object detection. In: *2016 2nd international conference on contemporary computing and informatics (IC3I)*. IEEE, 2016, s. 409–413.
- [26] ASLANI, Sepehr; MAHDAVI-NASAB, Homayoun. Optical flow based moving object detection and tracking for traffic surveillance. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*. 2013, roč. 7, č. 9, s. 1252–1256.
- [27] YE, Jin; SUN, Bingzhen; CHU, Xiaoli; ZHAN, Jianming; BAO, Qiang; CAI, Jianxiang. A novel diversified attribute group decision-making method over multi-source heterogeneous fuzzy decision systems with its application to gout diagnosis. *IEEE Transactions on Fuzzy Systems*. 2022.
- [28] REID, Sam. A review of heterogeneous ensemble methods. *Department of Computer Science, University of Colorado at Boulder*. 2007.
- [29] TSOUMAKAS, Grigorios; KATAKIS, Ioannis; VLAHAVAS, Ioannis. Effective voting of heterogeneous classifiers. In: *European conference on machine learning*. Springer, 2004, s. 465–476.
- [30] DUFFY, Nigel; HELMBOLD, David. Boosting methods for regression. *Machine Learning*. 2002, roč. 47, s. 153–200.
- [31] BREIMAN, Leo. Bagging predictors. *Machine learning*. 1996, roč. 24, s. 123–140.
- [32] DŽEROSKI, Saso; ŽENKO, Bernard. Is combining classifiers with stacking better than selecting the best one? *Machine learning*. 2004, roč. 54, s. 255–273.

# Seznam obrázků

- 2.1 Editorový mód se základní tratí
- 3.1 Hledaný bod optického toku
- 3.2 Znamá rovnice přímky
- 3.3 Rozložení na normálovou a tečnou složku
- 3.4 Skutečný pohyb celého obrazce
- 3.5 Zaměření na pohyb pouhé části obrazce
- 3.6 Snímek nebe a špatné podmíněné okolí
- 3.7 Pestrobarevný snímek s dobrým podmíněním
- 4.1 Heterogenní rozhodovací systém - volba hlasováním
- 4.2 Heterogenní rozhodovací systém - volba váhováním
- 5.1 Výstup výpočtu optického toku
- 5.2 Standardní zápis bodu  $(-3, 2)$  v kartézském systému
- 5.3 Standardní zápis bodu  $(2, \pi/4)$  v polárním systému
- 5.4 Ukázka z pilotního zpracování prvního analyzovaného videa
- 5.5 Ukázka z prvního zpracovaného videa v Trackmanii
- 5.6 První iterace vylepšeného prostředí v Trackmanii
- 5.7 Druhá iterace prostředí v Trackmanii
- 5.8 Přímočarý pohyb vpřed
- 5.9 Optický tok v průběhu rotace
- 6.1 Optický tok na černém pozadí
- 6.2 Optický tok na černém pozadí i s menšími *Velikostmi* toku
- 6.3 Binární zobrazení *Velikosti* optického toku v Trackmanii
- 6.4 Binární zobrazení *Velikosti* optického toku při zatáčení
- 6.5 Originál video z pole

- 6.6 Video extrahované z výpočtu optického toku
- 6.7 Originál video z lesa pořízené dronem
- 6.8 Video z dronu po extrakci obrazu z optického toku
- 6.9 Originál video z vinic
- 6.10 Extrahované video z vinic
- 6.11 Příklad histogramu z videa z dronu
- 6.12 Poměr pixelů v průběhu videa z dronu
- 7.1 Porovnání potenciálů v Trackmanii s originálem
- 7.2 Porovnání potenciálů z dronu v přírodě s originálem
- 7.3 Porovnání potenciálů z vinic s originálem
- 7.4 RGB zpracování videa z Trackmanie
- 7.5 Konkrétní poměr barevného rozdělení v obraze
- 7.6 Originální video z Trackmanie
- 7.7 RGB zpracování Trackmanie
- 7.8 Originální video z vinic
- 7.9 RGB zpracování vinic
- 7.10 Originální video z z dronu
- 7.11 RGB zpracování videa z dronu
- 7.12 Efekt Clustrování v poli
- 7.13 Efekt Clustrování u dronu v lese
- 7.14 Chodba bez světla
- 7.15 RGB chodba bez světla
- 7.16 Chodba se světlem
- 7.17 RGB chodba se světlem
- 8.1 Úhlová analýza pohybu vpřed
- 8.2 Úhlová analýza při rotaci vlevo
- 8.3 Úhlová analýza při rotaci vpravo
- 8.4 Úhlová analýza pohybu vpřed na videu z louky
- 9.1 Schéma vývoje zkoumání metod na základě optického toku



# Přílohy

- A1** m-file: OpticalFlow 1 - Sipky
- A2** m-file: OpticalFlow 2 - Binarni videni
- A3** m-file: OpticalFlow 3 - Potencialove mapy
- A4** m-file: OpticalFlow 4 - RGB
- A5** m-file: OpticalFlow 5 - Vliv svetla
- A6** m-file: OpticalFlow 6 - Uhly
- A7** Video Zpracovane - Dron - RGB - Kratke
- A8** Video Zpracovane - Dron - Uhel - Kratke
- A9** Video Zpracovane - Pole - Binarni videni - Kratke
- A10** Video Zpracovane - Trackmania - Bezna trat - Potencialove mapy - Kratke