



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**DETEKCE DOPRAVNÍCH PŘESTUPKŮ NA KŘIŽOVAT-
KÁCH**

TRAFFIC VIOLATION DETECTION ON CROSSROADS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MIROSLAV KARPÍŠEK

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAKUB ŠPAÑHEL

BRNO 2019

Zadání bakalářské práce



20822

Student: **Karpíšek Miroslav**
Program: Informační technologie
Název: **Detekce dopravních přestupků na křižovatkách**
Traffic Violation Detection on Crossroads
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy zpracování obrazu. Zaměřte se zejména na problematiku detekce statických i dynamických objektů v obraze.
2. Dohleďte dostupné metody na automatickou kalibraci kamery.
3. Vyberte vhodnou metodu a navrhnete možný způsob řešení daného problému.
4. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
5. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
6. Vytvořte stručný plakát a video prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Špaňhel Jakub, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 6. listopadu 2018

Abstrakt

Předložená bakalářská práce se zabývá navržením vhodného postupu pro detekci jízdy na červenou. V rámci teoretické části práce jsou popsány aktuální postupy využívané pro zpracování obrazu. Praktická část práce se zabývá vlastním návrhem programu pro automatickou detekci jízdnic pruhů, sledování vozidel a určení aktuálního stavu semaforu. V práci jsou též diskutovány výsledky získané experimentováním s vytvořeným programem a diskutovány další možnosti jeho dalšího zlepšení.

Abstract

This bachelor thesis presents procedure for the detection of red-light violation. In the theoretical part of the thesis, the current solution approaches used in image processing are described. The practical part focuses on creation of program for automatic traffic corridors detection, vehicle tracking and the current traffic light state detection. The results obtained by experimenting with the proposed procedure and the possibilities of its further improvement are also discussed.

Klíčová slova

detekce stavu semaforu, sledování vozidel, jízda na červenou, detekce objektů, jízdnic pruhy, OpenCV, úběžník, automatická kalibrace, Kalmanův filtr, problém přiřazení, RANSAC, HSV, Houghova transformace, detekce hran, sledování objektů

Keywords

traffic light state detection, car tracking, red-light violation, traffic corridors, object detection, OpenCV, vanishing point, automatic calibration, Kalman filter, assignment problem, RANSAC, HSV, Hough transform, edge detection, object tracking

Citace

KARPÍŠEK, Miroslav. *Detekce dopravních přestupků na křižovatkách*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jakub Špaňhel

Detekce dopravních přestupků na křižovatkách

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jakuba Špaňhela. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Miroslav Karpíšek

15. května 2019

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Jakubu Špaňhelovi za vedení a rady, které mi poskytl během řešení této práce.

Obsah

1	Úvod	2
2	Použité přístupy	3
2.1	Segmentace obrazu	3
2.2	Filtrování obrazu	5
2.3	Operace nad binárním obrazem	7
3	Přehled současných způsobů řešení	9
3.1	Kalibrace kamery	9
3.2	Detekce objektů	12
3.3	Sledování pohybu objektů	15
4	Detekce jízdy na červenou	17
4.1	Získání stavu semaforu	17
4.2	Sledování pohybu automobilů v obraze	20
4.3	Kalibrace kamery	23
4.4	Segmentace oblastí vozovky	26
4.5	Detekce jízdy na červenou	28
5	Implementace a vyhodnocení	29
5.1	Využití knihovny	29
5.2	Návrh programu	29
5.3	Úspěšnost použitých metod	32
5.4	Možnost dalšího vývoje	36
6	Závěr	37
	Literatura	38
A	Obsah DVD	42

Kapitola 1

Úvod

Z výroční zprávy ministerstva dopravy České republiky [26] lze vyčíst, že nerespektování pokynu „Stůj!“ se řadí mezi časté přestupky (tvoří téměř 2.5% veškerých evidovaných dopravních přestupků). Zároveň se řadí mezi nejčastější přestupky, které jsou ohodnoceny více jak třemi trestnými body. Pouze za rok 2018 bylo těchto přestupků na území ČR evidováno necelých 9000, což odpovídá průměrně 25 přestupkům za den. Z toho důvodu jsem se ve své práci zaměřil právě na detekci jízdy na červenou jakožto represivním opatřením vůči delikventům.

Nasazením kamer lze dosáhnout snížení počtu jízd na červenou na dané křižovatce na polovinu a díky tomu také snížit vysokou nehodovost způsobenou právě nerespektováním světelného značení až o 30%. Vyplývá to z práce [29], která shrnuje různé studie v oblasti zkoumání vlivu nasazení dopravních kamer na redukcii nehodovosti na křižovatkách.

V rámci mé práce je čtenář nejprve v kapitole 2 seznámen se základními metodami týkající se segmentace obrazu, filtrování a práce s binárním obrazem používanými v oblasti počítačového vidění. Následuje kapitola 3 sloužící jako přehled současných řešení podproblémů souvisejících s detekcí jízdy na červenou. Jedná se o postupy pro kalibraci scény, detekci objektů a jejich sledování. V následující kapitole 4 se zabývám vlastním návrhem řešení dílčích úkonů potřebných pro úspěšné provedení detekce jízdy na červenou za využití metod představených v předešlých kapitolách. Na závěr 5 je představen vlastní způsob implementace navržených postupů a zhodnocení kvality navrženého postupu, kde jsou též diskutována možná zlepšení.

Kapitola 2

Použité přístupy

Při použití metod pro zpracování obrazu je třeba mít na paměti, že záznam je sekvencí po sobě jdoucích snímků. Jednotlivé snímky jsou reprezentovány jako matice hodnot a proto mohou být výpočty prováděné v oblasti zpracování obrazu často časově náročné, jelikož je třeba provést operace nad veškerými pixely na snímku, kde doba zpracování roste exponenciálně vůči změně velikosti snímku. Také je třeba brát v úvahu, že některé postupy vyžadují výpočty o více iteracích. Důraz na dobu výpočtu se stává kritickým parametrem pro metody v oblasti zpracování obrazu, kde je třeba, aby jednotlivé snímky byly vyhodnocovány dostatečně rychle pro vytvoření aplikace fungující v reálném čase.

Následující kapitola představuje výběr metod pro práci s videozáznamem a strojovým získáváním informací z něj. Jejím cílem je zasvětit čtenáře do problematiky, která je pro pochopení této práce nezbytná – nejedná se tedy o výčet veškerých pojmů, který přesahuje obsah této práce. V rámci této kapitoly je čtenář nejprve seznámen se způsoby segmentace obrazu. Následně je čtenáři představeno použití filtrů jednak pro vyhlazení obrazu, ale také jejich využití při detekci hran. Závěrem obsahuje tato kapitola výběr metod pro práci s binárním obrazem.

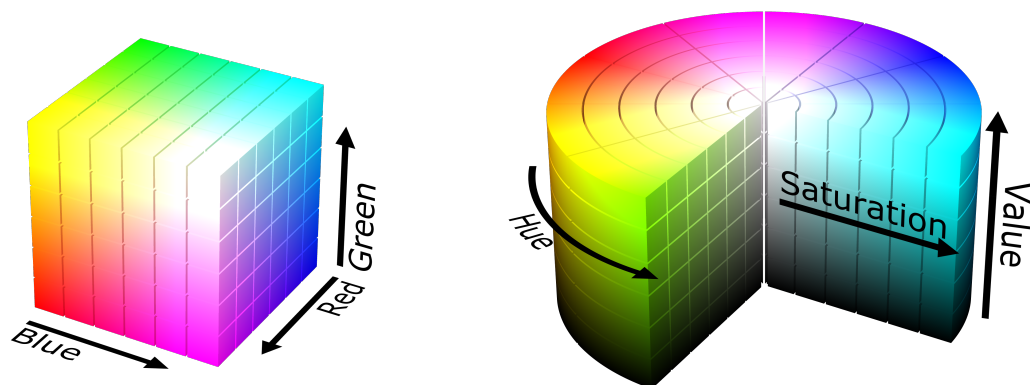
2.1 Segmentace obrazu

Jednotlivé části obrazu lze rozdělit do skupin pomocí různých metod. V rámci této sekce je nejprve popsána metoda která umožňuje v binárním obraze nebo obraze ve stupních šedi jednotlivé pixely přiřadit do skupin pomocí porovnání jejich hodnot. Následně je popsán postup pro segmentaci obrazu na základě barvy, kdy je vhodnější využít odlišný barevný model a až v rámci něj provést porovnání jednotlivých hodnot. Dále je vysvětlen způsob rozdělení obrazu na pixely příslušící popředí a pozadí. Závěrem sekce je popsán záplavový algoritmus, který nalezne své využití při vyplnění určité ohraničené oblasti v obraze.

Prahování

Někdy je třeba po mnoha výpočetních krocích provést rozhodnutí nad každým konkrétním pixelem v obraze, kde každý z nich je porovnán se zadanou prahovou hodnotou a na základě porovnání je buď jeho hodnota změněna nebo ponechána. Existuje více typů prahování, které se liší svým výstupem [2].

Metoda prahování je často metodou již vestavěnou do jiných metod buď za účelem úpravy vstupu nebo výstupu, jelikož umožňuje filtraci mezních hodnot v obraze. Prahování



Obrázek 2.1: Možná reprezentace jednotlivých modelů. Vlevo model RGB, pracující se třemi barevnými složkami (červená, zelená, modrá). Vpravo model HSV využívající pro popis barvy hodnoty odstínu, sytosti barvy a hodnotou jasu. Zdroje [32], [31].

také umožňuje vytvořit binární obrazy na základě příslušnosti daných pixelů do specifikovaných mezí.

Segmentování barevného spektra

Při zpracování obrazu je někdy třeba pracovat pouze s pixely, jejichž hodnota odpovídá určitému rozsahu barevného spektra. Pro záznam barev bývá nejčastěji využíván aditivní barevný model RGB (*Red, Green, Blue*). Který pro každou základní barevnou složku (červená, zelená, modrá) uchovává hodnotu její přítomnosti, kde 0 představuje že daná složka není na daném pixelu přítomna a hodnota 255 (pokud uvažujeme reprezentaci barvy jednoho pixelu pomocí 3 bytů) představuje její plnou přítomnost. V případě, že jsou se tyto hodnoty rovny je výsledný pixel v odstínech šedi. Tento model je výhodný pro zobrazování snímků na obrazovkách, avšak pro filtrování obsahu podle barvy vhodný není. Většinou je totiž žádoucí pouze určitá výšeč barevného spektra, která není pomocí reprezentace barvy modelem RGB dobře specifikovatelná.

Řešením je reprezentace barev pomocí jiného barevného modelu. V rámci této práce je při filtraci obrazu na základě barvy využít barevný model HSV [34] (*Hue, Saturation, Value*), jehož reprezentace je znázorněna na obrázku 2.1. Tento model poskytuje možnost filtrovat barvu na základě odstínu, který je definován pomocí hodnoty *Hue* a tak lze jednoduše definovat výšeč barevného spektra stanovením prahu této hodnoty.

Detekce pozadí na snímcích

V některých metodách je třeba oddělit pozadí scény od popředí. Pomocí detekovaného popředí lze nalézt v obraze pohybující se objekty viz sekce 3.2. V ideálním případě, kdy existuje přesný snímek pozadí, jej lze porovnat s aktuálním snímkem a v pixelech kde je absolutní hodnota rozdílu vyšší než nula, lze s určitostí tvrdit, že daný pixel připadá popředí. Většinou však takový snímek neexistuje a je za tímto účelem třeba využít metod pro extrakci pozadí ze záznamu. Pro použití následujících metod platí předpoklad, že je záznam pořízen ze statické kamery.

Pro modelování pozadí lze využít metodu představenou v práci [18], která využívá kombinace normálních rozdílů pro každý modelovaný pixel pozadí. Váha těchto kombinací

poté určuje dobu, po jakou jednotlivé pixely zůstaly v obraze beze změny. Výsledná barva pixelu je určena právě touto vahou a její hodnota definuje staticnost daného pixelu.

Alternativou, která poskytuje vyšší odolnost vůči změnám osvětlení, je metoda popsaná v práci [14], která přiřazuje novějším snímkům vyšší váhy, což umožňuje rychlejší adaptaci modelovaného pozadí.

Záplavové algoritmy

Segmentace scény může být provedena také pomocí záplavového algoritmu. Ten najde využití například při výplni určitého objektu ohraničeného hranami. Pro provedení této metody je třeba určit výchozí bod. Následně je třeba stanovit hodnotu prahu, která určuje hranici, za kterou již šíření hodnoty pixelu nemá být vedeno. Jedná se o iterační algoritmus, který postupně šíří zadanou hodnotu z výchozího bodu, dokud je mu tak umožněno definovanými hranami. Je možné také specifikovat, zda-li algoritmus šíří pixely ve čtyřech směrech nebo zda-li využívá i diagonálních sousedů [2].

2.2 Filtrování obrazu

Filtry lze využít pro rozdílné úkony. Jednak umožňují vyhlazení nepřesností v obraze, v takovém případě se jedná o nízkofrekvenční filtry které zbavují obraz strmých přechodů. Naopak vysokofrekvenčních filtrů je použito v případě, kdy je třeba detekovat strmé přechody v obraze. Jedná se zpravidla o hrany, jejichž extrakce je klíčová pro využití dále popsaných metod 2.3 pro detekci přímek nebo kontur v obraze. Filtry se zpravidla na obraz aplikují pomocí konvoluce, kdy je na základě konvolučního jádra vypočtena výsledná hodnota pro každý pixel [2].

Nízkofrekvenční filtry

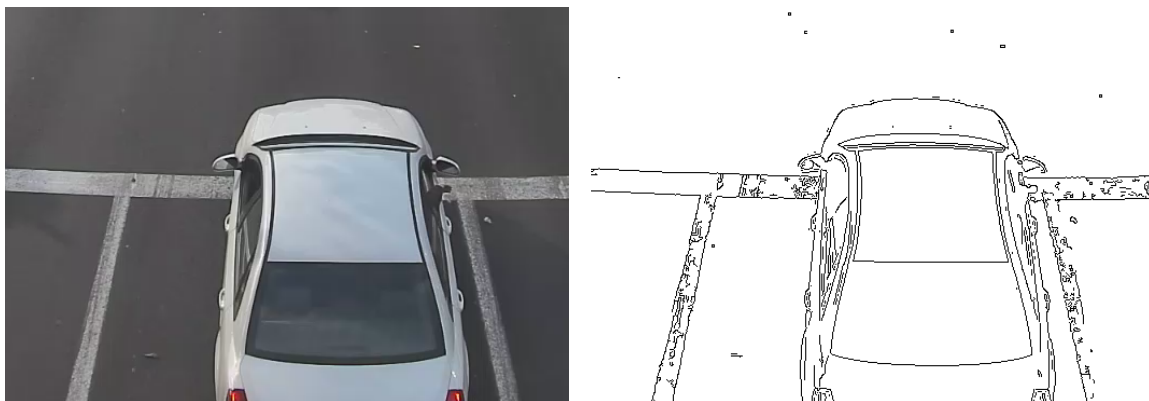
Při pořizování záznamu dochází k mnoha nepřesnostem. Tyto nepřesnosti mohou mít různý charakter původu. Nepřesnosti vznikají již při vytváření samotných snímků jako tomu je například u radiální a tangenciální distorze obrazu, nebo též mohou pocházet z fyzikálních vlastností částicového proudění světla před jejich zachycením [17]. Avšak mohou vznikat také při kompresi, při které se určitá část nosné informace ztratí.

Nízkofrekvenční filtry, jinak také nazývané dolní propustě, potlačují vysokofrekvenční aspekty obrazu. Díky tomu je lze využít pro zbavení obrazu šumu nebo přílišných detailů které mohou vnášet chyby do pozdějších výpočtů. Takové využití je viditelné na obrázku 2.2b, kde je nízkofrekvenčního filtru využito pro vyhlazení obrazu před provedení detekce hran. Filtrovaný obraz pomocí nízkofrekvenčního filtru se pozorovateli jeví bez strmých hran a přechodů. Mezi filtry dolní propustě řadíme například lineární Gaussův filtr.

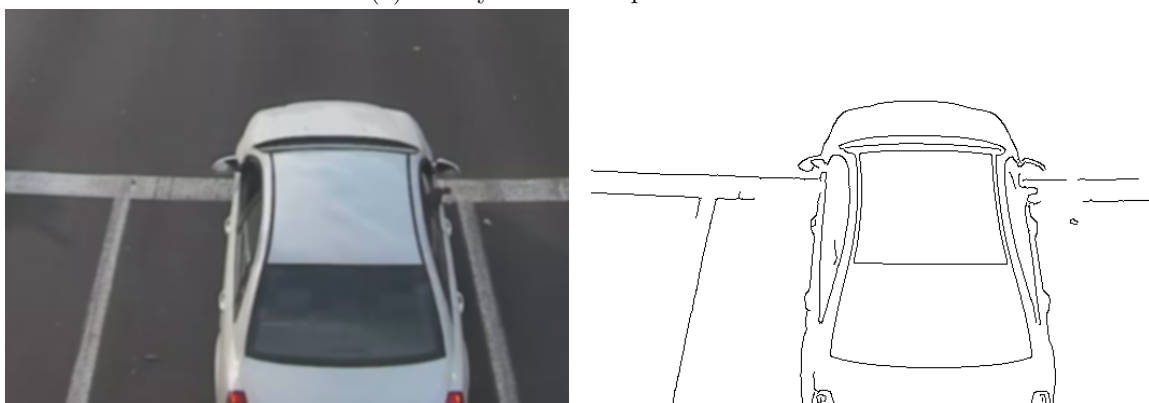
Vysokofrekvenční filtry

V mnoha případech je třeba v obraze nalézt objekty ve snímané scéně. Objekty jsou ve scéně ohraničeny hranami, tedy místy se strmou změnou hodnot a proto detekce hran je důležitou součástí zpracování obrazu. V předešlé sekci byl ukázáno využití filtrů dolní propustě pro vyhlazení obrazu. Při detekci hran je však třeba opačného efektu, tedy zesílení ostrých přechodů v obraze a potlačení nízkofrekvenčních oblastí.

Pro detekci hran lze využít například Sobelův filtr [12]. Jedná se o lineární filtr, jehož jádrem je čtvercová matice o velikosti 3x3. Jelikož se jedná o takzvaný směrový filtr, který



(a) Bez vyhlazení vstupního obrazu.



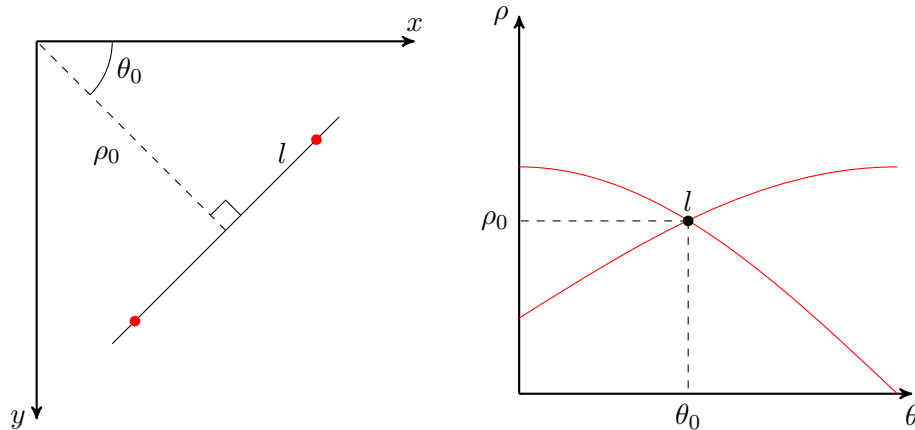
(b) Použití Gaussova filtru před detekováním hran.

Obrázek 2.2: Obrázky obsahují výstupy Cannyho algoritmu pro detekci hran. Pro porovnání ukázka výstupu před a po použití Gaussova filtru pro redukci šumu ve výsledném obraze (detekované hrany jsou barevně invertovány).

je schopen provádět zesílení vysokofrekvenčního obsahu vždy buď pouze po vertikálních nebo horizontálních souřadnicích, je třeba pro nalezení hran ve snímku vždy výstupy pro oba směry sečíst a analýzu provádět až na takto vytvořeném snímku. Pro získání binárního obrazu hran je třeba výsledný snímek podrobit prahování [22].

S binárním výstupem Sobelova filtru však nelze provádět přesné výpočty v oblasti rozpoznávání objektů nebo hledání přímek v obraze. Důvodem je nemožnost definovat tloušťku nalezených hran na výstupu, stejně tak nelze efektivně nastavit vhodný práh tak, aby obraz obsahoval všechny důležité hrany, avšak zároveň redukoval počet těch nepotřebných, které mohou při následném výpočtu vyústit v chybné výsledky.

Pro částečnou eliminaci zmíněných problémů lze využít Cannyho algoritmus [3], jehož základem je již zmíněný Sobelův filtr. Ten umožňuje definovat prahové hodnoty pro získání kompromisu mezi nalezením potřebných kontur a odstraněním nepotřebných [22]. Před detekcí hran je možné využít nízkofrekvenčních filtrů pro zbavení obrazu šumu a získání pouze výrazných hran v obraze. Porovnání detekce hran s využitím vyhlazeného snímku a původního je viditelné na obrázcích 2.2.



Obrázek 2.3: Vlevo původní přímka v obraze. Vpravo její reprezentace v Houghově prostoru. Inspirováno ze [2].

2.3 Operace nad binárním obrazem

Jak již bylo zmíněno v předešlé sekci 2.2. Detekce hran je důležitou součástí zpracování obrazu, jelikož umožňuje lépe analyzovat snímanou scénu. Samotné binární výstupy metod pro detekci hran však pro další zkoumání objektů ve scéně příliš užitečné nejsou. Bylo by vhodné jednotlivé hrany lépe popsat a určit jejich vzájemnou příslušnost – například jaké detekované pixely společně tvoří v obraze přímky nebo popisují určité kontury. Pro jejich popis by se též hodilo využití efektivnější interpretace, kdy by detekované přímky byly popsány rovnicí a kontury objektů pomocí posloupnosti bodů. Řešením se zabývají dále představené metody.

Nalezení přímek

Ve světě vytvořeném člověkem se lze často setkat s objekty, které obsahují přímé linie, pomocí nichž lze takové objekty jednoduše rozpoznat či charakterizovat. Také z tohoto důvodu je schopnost nalezení přímek v obraze jednou z elementárních metod, používanou v mnoha dalších algoritmech.

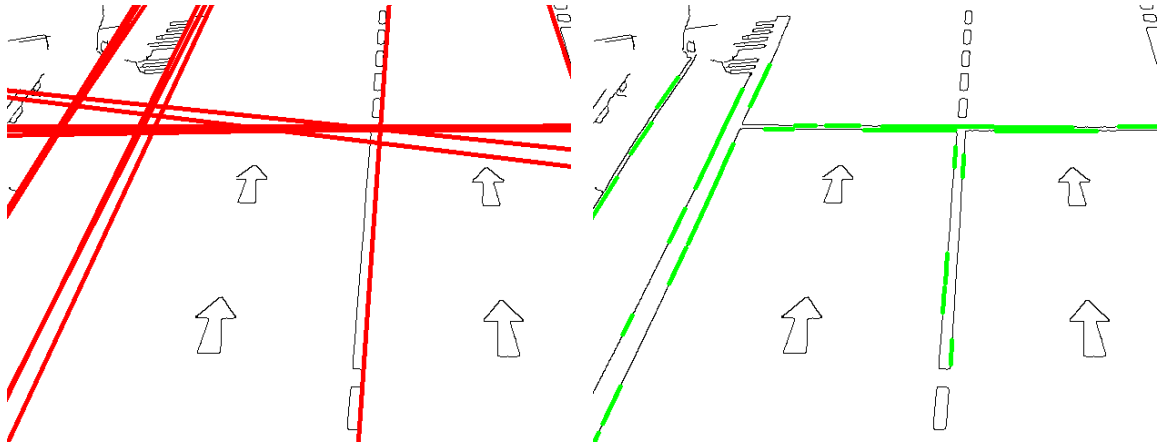
Algoritmus umožňující detekci přímek na základě nalezených hran je Houghova transformace (*Hough transform*) [11]. Přestože byl primárně vyvinut pro detekci rovných čar, tak jej lze s úpravou využít i pro detekci jiných jednoduchých obrazových struktur, například kružnic. Výstupem Houghovy transformace pro detekci přímek je její parametrický popis, popsán rovnicí 2.1:

$$\rho = x \cos \theta + y \sin \theta \quad (2.1)$$

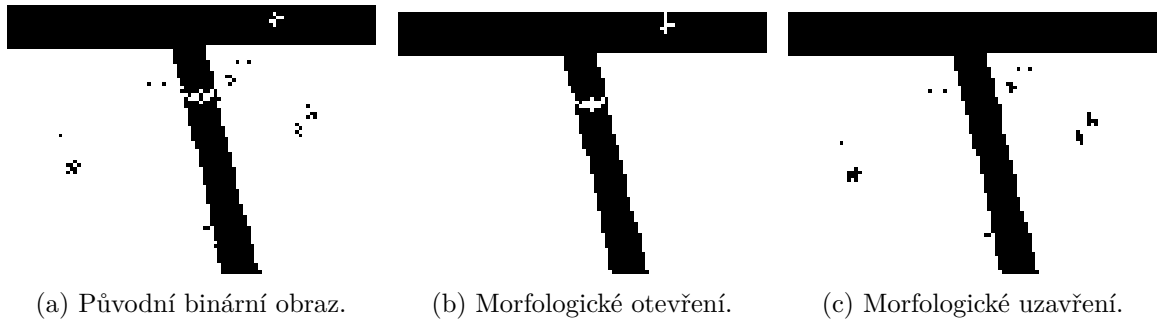
parametr ρ je délka kolmice na popisovanou přímku která prochází počátkem souřadnicového systému a θ je úhel jež tato kolmice svírá s osou x [22]. Graficky je tento popis znázorněn na obrázku 2.3.

Nevýhodou Houghovy transformace je možnost chybné vícenásobné detekce téže linie. Jelikož je výstupem přímka, nelze tak jejich nalezení omezit pouze na určitý úsek a přímka tedy protíná celý obraz.

Zmíněné problémy se snaží vyřešit pravděpodobnostní Houghova transformace (*probabilistic Hough transform*) [21] [20], která umožňuje detekovat jednotlivé segmenty čar. Po-



Obrázek 2.4: Vlevo jsou hrany nalezeny pomocí *Hough transform*, vpravo je detekce provedena pomocí *Probabilistic Hough transform* (detekované hrany jsou barevně invertovány).



(a) Původní binární obraz.

(b) Morfologické otevření.

(c) Morfologické uzavření.

Obrázek 2.5: Ukázka využití morfologických operací pro zbavení obrazu šumu a zacelení detekovaných objektů. *Obrázky jsou pro lepší viditelnost barevně invertovány.*

rovnání obou metod je viditelné na obrázku 2.4. Nalezené přímky lze využít pro detekování úběžníků v obraze. Tento postup je dále popsán v sekci 3.1.

Morfologické operace

Mezi morfologické operace [41] se řadí dilatace, eroze a jejich následné kombinace. Lze je využít při redukci šumu v binárních obrazech, přes izolaci určitých elementů nebo také k slučování sousedících objektů. Morfologické operace lze též využít pro zaplnění děr v binárních obrazech.

Dilatace využívá konvoluce s jádrem zadané velikosti, ze kterého je vybráno lokální maximum pro daný zkoumaný pixel. Eroze funguje na stejném principu ovšem z daného jádra je vybráno lokální minimum [2].

Provedení nejprve eroze s následnou dilatací se nazývá morfologickým otevřením, které slouží pro odstranění osamocených malých objektů – často se jedná o nepotřebný šum. Naproti tomu morfologické uzavření, které je tvořeno nejprve dilatací a až následnou erozí, umožňuje zbavit obraz děr uvnitř objektů. Využití morfologických lze vidět na obrázcích 2.5.

Kapitola 3

Přehled současných způsobů řešení

V rámci této kapitoly jsou popsány některé z existujících způsobů řešení jednotlivých podproblémů týkajících se detekce dopravních přestupků na křižovatkách.

V praxi používanou, avšak nejnákladnější metodou, je vytvoření systému, který má externí přístup k aktuálnímu stavu semaforu. Takový systém obdrží signál o změně stavu a dále je připojen k hardwarovému detektoru, který je schopný detekovat, pokud vozidlo překročí specifikovanou linii. Takto vytvořený systém poté zaručeně plní svoji funkci, avšak jak bylo zmíněno, jeho plošné nasazení by vyžadovalo nemalé finanční prostředky a pro každou instalaci by bylo třeba lidské práce s jeho kalibrací. Cílem mé práce je navrhnout automatické řešení, které je schopno ze záznamu bez lidského zásahu provádět detekci jízdy na červenou. Z tohoto důvodu v přehledu aktuálních řešení vynechám ta, která jsou využita v systémech vyžadují ruční kalibraci a tedy zároveň ta, která potřebují k funkčnosti více hardwarových zařízení kromě kamery.

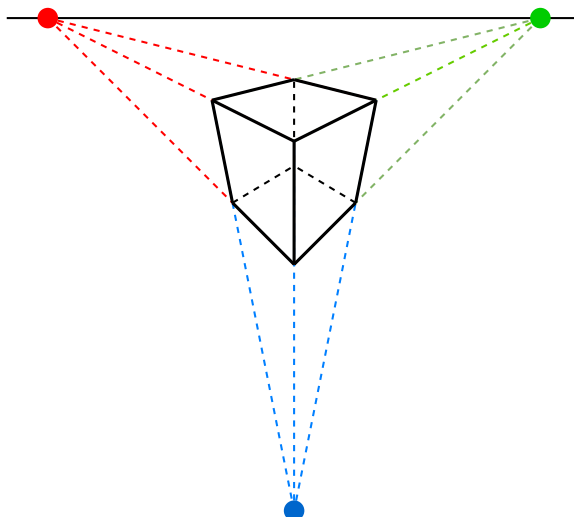
Nejprve bude čtenář seznámen s metodami používanými pro kalibraci kamery ve scéně, jež využívají nalezení ortogonálních úběžníků. Následně budou představeny postupy pro detekování objektů v obraze. Na závěr budou čtenáři popsány způsoby využití pro sledování pohybu objektů ve scéně.

3.1 Kalibrace kamery

Jelikož kamera, stejně jako lidské oko, zachycuje informace z reálného světa pouze na dvou-dimenzionální průmětnu, dochází k jevu, kdy se pozorovateli vzdálenější objekty jeví menší než ty pozorovateli blíže. Díky tomu se rovnoběžné přímky v prostoru na zachyceném snímku všechny protínají v jednom bodě známém jako úběžník (*vanishing point*) viz 3.1.

Zmíněné vlastnosti úběžníku lze využít pro kalibraci kamery ve snímané scéně jak je tomu navrženo v práci [9]. Nalezením tří dominantních úběžníků lze získat informaci o pozici jednotlivých struktur dvoudimenzionálního obrazu ve třídimeznionálním reálném prostoru. Podmínkou pro využití této metody je, aby snímaná scéna spadala do takzvaného *Manhattan world* (lidmi vytvořené struktury často obsahují pravidelné tvary, kde si jejich hrany jsou navzájem buď kolmé nebo rovnoběžné).

Pro nalezení úběžníků je třeba vyhledat průsečky přímek v obraze, které jsou si v reálném světě paralelní. Jelikož se v úběžníku setkávají veškeré rovnoběžné přímky ze snímané scény, bylo by dostačující najít dvě ideální přímky a nalézt jejich průsečík. Tato úloha však není lehce řešitelná, jelikož je třeba počítat s nepřesnostmi záznamu, způsobené například horizontální a vertikální distorzí.



Obrázek 3.1: Na obrázku lze vidět náčrt perspektivní projekce kvádrů. Hrany kvádrů, které jsou vůči sobě v třídímném prostoru paralelní, se v perspektivním promítání protínají v úběžníku. Černá čára, protínající dva úběžníky poté představuje horizont.

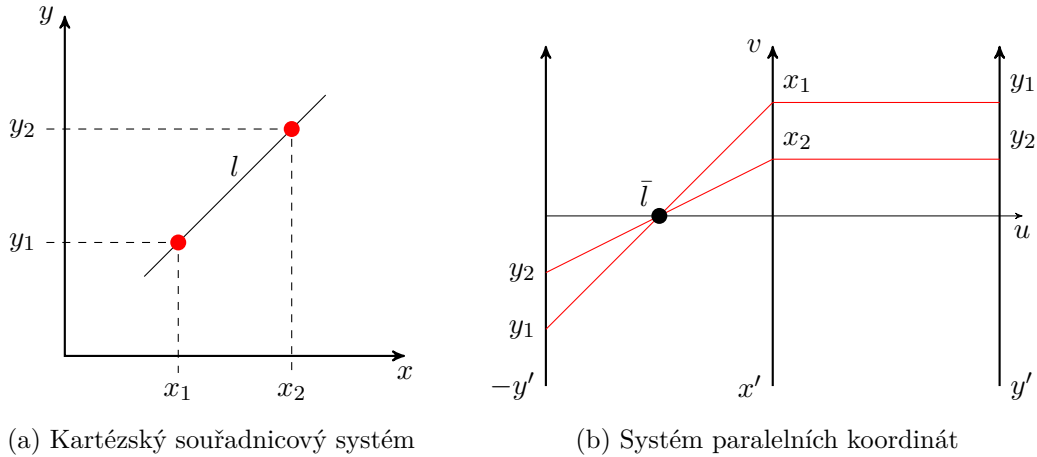
Pro detekci úběžníků lze využít existenci pravidelné mříže *Manhattan world*, definující tři ortogonální směry, které popisují snímaný prostor [4]. Tuto mříž lze nalézt v industriálních a městských oblastech, kde působením člověka vzniká velké množství pravidelných tvarů. Pozdějším výzkumem [5] bylo zjištěno, že postupy navržené pro detekci úběžníků v industriálních a městských oblastech lze využít i v oblastech venkovských, kde pravidelná mříž *Manhattan world* nemusí být na první pohled pozorovateli patrná.

Kalibrací kamery pomocí tří ortogonálních úběžníků a segmentací scény se zabývá práce [8], která pro kalibraci kamery využívá tří ortogonálních úběžníků. První úběžník je nalezen pomocí analýzy pohybu detekovaných automobilů. Pro detekci druhého úběžníku je využito detekovaných hran automobilů. V případě, kdy automobil míří směrem k prvnímu úběžníku, směřuje většina takto detekovaných hran k prvnímu nebo druhému úběžníku. Třetí ortogonální úběžník je vypočten pomocí již nalezených a předpokladu, že hlavní bod se nachází uprostřed obrazu. Tatáž práce se též zabývá postupem segmentace a klasifikace jízdních pruhů pomocí záznamu pohybu vozidel využitím trojrozměrného popisu vozidla ve scéně.

Dvojitá Houghova transformace

Při hledání úběžníku je třeba akumulovat určité množství přímek a nalézt jejich ideální průsečík. Přímky lze, jak již bylo zmíněno v předchozí sekci 2.3, v obraze nalézt pomocí algoritmu Houghovy transformace.

Využitím dvojitě Houghovy transformace lze dle metody zmíněné v práci [37] dosáhnout nalezení úběžníků. Jednotlivé body z kartézského souřadnicového systému se nejprve první transformací mapují na křivky v parametrickém prostoru $\rho - \theta$. Pro body, které vytvářejí ve vstupním obraze přímku, platí, že v parametrickém $\rho - \theta$ prostoru se jejich křivky protínají v konkrétním bodě. Pro nalezení úběžníků je třeba provést druhou transformaci, kde vstupem jsou body nalezené po první Houghově transformaci. Výsledné nalezené průsečíky křivek jsou hledané úběžníky. Ve zmíněné práci je též navrženo využití třetí Houghovy transformace pro nalezení kolineárních úběžníků.



Obrázek 3.2: Znázornění přímky l v kartézském souřadnicovém systému a její reprezentace \bar{l} v prostoru paralelních souřadnic. Inspirováno ze [7].

Alternativní postup využitím reprezentace přímek v paralelním prostoru

Alternativní způsob k nalezení úběžníku je představen v práci [7], kde se pro popis nalezených přímek v obraze využívá systém paralelních souřadnic. Jednotlivé body přímky jsou z kartézského souřadnicového systému převedeny do systému paralelních souřadnic (viz obrázek 3.2). Bod v kartézských souřadnicích se mapuje na polynom n -tého stupně, kde n odpovídá počtu dimenzí prostoru v němž je převáděná přímka definována. Přímka $l : y = mx + b$ odpovídá bodu v systému paralelních souřadnic $\bar{l} : (d, b, 1 - m)_{\mathbb{P}^2}$, pro který platí, že d představuje vzdálenost mezi osami x' a y' . Pro přímky, pro které platí $0 \leq m < \infty$, je autory práce navrženo využití T (*twisted*) prostoru, jelikož reprezentace takových přímek se nenachází v systému paralelních souřadnic v intervalu $(0, d)$. T prostor obsahuje osy x' a $-y'$. Kombinací T prostoru s původním S (*straight*) prostorem lze reprezentovat přímky s libovolným parametrem m (viz obrázek 3.2).

Aproximace úběžníku prostoru paralelních souřadnic

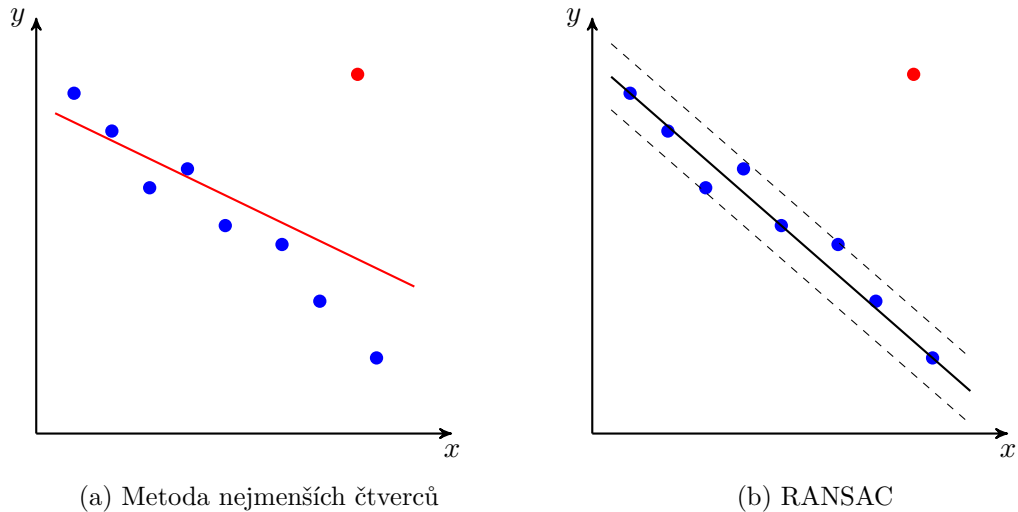
Tatáž práce se věnuje problematice nalezení úběžníku. Jelikož úběžník je ve vstupním obraze bodem, kde se sbíhají rovnoběžné přímky z reálného světa. Tak v paralelním souřadnicovém systému, který je využit ve zmiňované práci, je reprezentován polynomem druhého stupně protínajícími jednotlivé body na intervalu $(-\infty, \infty)$. Proložení polynomu skrze tyto body lze získat souřadnice úběžníku.

Namísto aproximace polynomu na základě bodů v jednotlivých intervalech $(-\infty, 0)$ a $(0, \infty)$ lze využít lineární transformace uvedené v práci [7] pro převod bodů mezi S a T prostorem;

$$\forall l : \bar{l}_S = (u, v, 1)_{\mathbb{P}^2} \Rightarrow \bar{l}_T = (u, v, \frac{d - 2u}{d})_{\mathbb{P}^2}, \quad (3.1)$$

$$\forall l : \bar{l}_T = (u, v, 1)_{\mathbb{P}^2} \Rightarrow \bar{l}_S = (u, v, \frac{d + 2u}{d})_{\mathbb{P}^2}. \quad (3.2)$$

Následnou aproximaci přímky ze zadaných bodů lze řešit mnoha způsoby. Ne všechny jsou však pro řešení tohoto problému vhodné, jelikož je během výpočtu třeba počítat s exis-



Obrázek 3.3: Porovnání metody nejmenších čtverců s metodou RANSAC pro aproximaci přímky ze zadaných bodů. Metoda nejmenších čtverců je náchylná vůči chybným bodům, které výrazně ovlivňují výsledek.

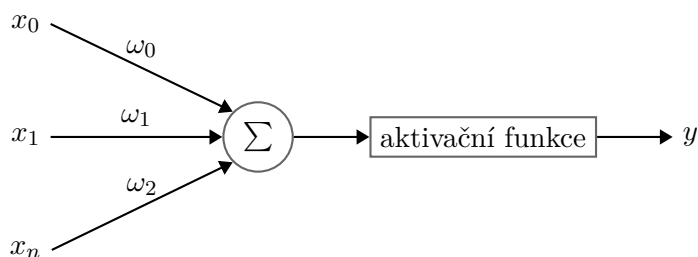
tenci chyb které nesmí narušit výsledek. Tento problém je například patrný při využití metody nejmenších čtverců, jelikož vzdálené chybné body mají při rozhodování větší váhu než-li ty, jež jsou blízko ideální aproximaci.

Za tímto účelem bylo v práci [24] pro detekci průsečíků bodů v prostoru paralelních souřadnic využito metody RANSAC (*Random sample consensus*) [13], která se s vysokým počtem chybně detekovaných přímek umí vyrovnat (jednotlivé přímky jsou v pralalelním souřadnicovém systému reprezentovány body). Nejprve se zvolí náhodný pár bodů, vytvoří se mezi nimi přímka a poté je zkoumána vzdálenost všech ostatních bodů od takto vytvořené přímky. Ohodnocení dané přímky se poté rovná počtu bodů, jejichž vzdálenost od přímky byla menší než zadaný práh. Výstupem metody je takto vytvořená přímka s nejvyšším ohodnocením. Porovnání metody nejmenších čtverců s algoritmem RANSAC lze vidět na obrázku 3.3.

3.2 Detekce objektů

Dříve popsané metody pro modelování pozadí scény v sekci 2.1 na snímcích ze stacionární kamery lze využít pro detekci objektů v obraze. Pokud totiž provedeme negaci pozadí na daném snímku získáme tak popředí, které využitím morfologických operací a filtrů může zbavit nedokonalostí a následně na takto upraveném snímku detekovat kontury – okraje objektu popsané pomocí sekvence bodů. Nalezenými konturami v popředí jsou pohybující se objekty v obraze. Tato metoda je výpočetně nenáročná v porovnání s detekcí objektů pomocí neuronových sítí a pro některé aplikace je dostačující. Využití této metody bylo například navrženo pro detekci osob v práci [14], kde sloužila pro sledování návštěvníků v místnosti galerie. Nebo v práci [38], kde byla nasazená pro získání statistických údajů o projíždějících automobilech.

Tato metoda ovšem má svá omezení. Zejména se jedná o nemožnost detekovat statické objekty v obraze. Které ovšem patří do popředí, jedná se například o auta stojící z důvodu povelu „Stůj!“. Dále je tato metoda náchylná na okluzi jednotlivých kontur. Při překryvu



Obrázek 3.4: Schéma perceptronu; x_0, x_1, \dots, x_n představují vstupy, $\omega_0, \omega_1, \dots, \omega_n$ určují váhy daných přechodů. Po průchodu aktivační funkcí je výstupem y .

dvou snímaných objektů ve scéně dochází zpravidla ke sloučení kontur v jednu a tak získání chybných informací.

Konvoluční neuronové sítě

Alternativou a s rozvojem výpočetní techniky stále častěji používanou metodou pro detekci objektů v obraze je využití strojového učení. Cílem strojového učení je přeměna dat v informace, kdy po určité době učení je stroj sám schopen odpovědět na dotazy ohledně dat [2]. V posledních letech dochází ke strmému vývoji strojového učení v oblasti neuronových sítí, které jsou inspirovány biologickými neuronovými sítěmi. Neuronová síť se skládá z mnoha propojených výpočetních bloků, kde každou takový blok lze považovat za umělý neuron (viz obrázek 3.4) a jejich propojení jakožto ekvivalent axonů.

Typická struktura neuronové sítě popsaná v práci [27] obsahuje vstupní vrstvu, která slouží pro vkládání vstupních dat – často ve formě multidimenzionálních vektorů. Dále obsahuje skryté vrstvy, kde jednotlivé výstupy vyšších vrstev slouží jako vstupy vrstev nižších a také výstupní vrstvu. V téže práci je také provedeno členění na základní paradigmatu používaná pro trénování neuronové sítě určené pro zpracování obrazu:

- **učení s učitelem (*supervised learning*)** - trénovací data obsahují značky, které slouží jako cíle. Cílem tohoto typu trénování je redukce chyby vůči značkám. Tato chyba je počítána pomocí *loss funkce* na základě jejího výstupu je poté prováděna zpětná propagace (*back propagation*), díky níž dochází k úpravě vah jednotlivých propojení mezi neurony.
- **učení bez učitele (*unsupervised learning*)** - tento způsob neobsahuje ve fázi trénování žádné značky. Hodnota úspěchu predikce je poté počítána na základě hodnotící funkce. Jedná se o méně častěji používaný postup při detekci a klasifikaci v rámci zpracování obrazu.

Konvoluční neuronové sítě se odlišují tím, že jsou schopny výrazně redukovat počet potřebných parametrů v rámci sítě vložím konvolučních vrstev skládajících se z trénovatelných konvolučních jader. Díky tomu umožňují vytvoření výrazně větších a komplexnějších modelů. Hlavní výhodou konvolučních neuronových sítí je zbavení detekovaných vlastností prostorové závislosti. Toho se využívá právě například při detekci automobilů, kde by úspěšnost detekce neměla být závislá na poloze automobilu v obraze. Dalším důležitým aspektem je schopnost extrahovat vlastnosti různé úrovně abstrakce v závislosti na hloubce dané vrstvy v rámci neuronové sítě. Ve vyšších vrstvách dochází k detekci hran, se zvyšující hloubkou vzrůstá komplexita zkoumaných vlastností od jednoduchých tvarů až po vysoce komplexní tvary v nejnižších vrstvách [1].

Využití neuronové sítě pro detekci objektů umožňuje kromě detekce i objekty klasifikovat a tak odlišit například automobil od chodce, což v případě předešlé metody využívající detekci pozadí možné není. Detekce pomocí neuronové sítě poskytuje po správném natrénování více odolnosti vůči změnám osvětlení nebo okluzi. Správné natrénování však závisí na výběru vhodných trénovacích dat. Při trénování neuronové sítě je třeba zvolit vhodná trénovací data – jedná se o data dostatečně diverzifikovaná a o dostatečné velikosti v poměru k vytvořené neuronové síti. Neuronové sítě s vyšším počtem parametrů mají také vyšší tendenci se přeučit, jako tomu je vysvětleno v práci [27].

Pro detekci a klasifikaci objektů v obraze jsou vytvořeny modely, jejichž výstupem jsou souřadnice určující obdélník okolo nalezeného objektu společně s hodnotou určující třídu, do které detekovaný objekt spadá a ohodnocení kvality dané detekce. Jedním z využívaných typů modelů sloužících pro detekci objektů v obraze jsou modely spadající pod klasifikaci Faster R-CNN [28]. Tyto modely dosahují vysoké přesnosti [16], avšak jejich model je příliš komplexní. Alternativou jsou modely spadající do kategorie SSD (*single shot detector*) [25] jejichž hlavním rysem je využití pouze jednoho dopředného průchodu neuronovou sítí.

Detekce semaforu

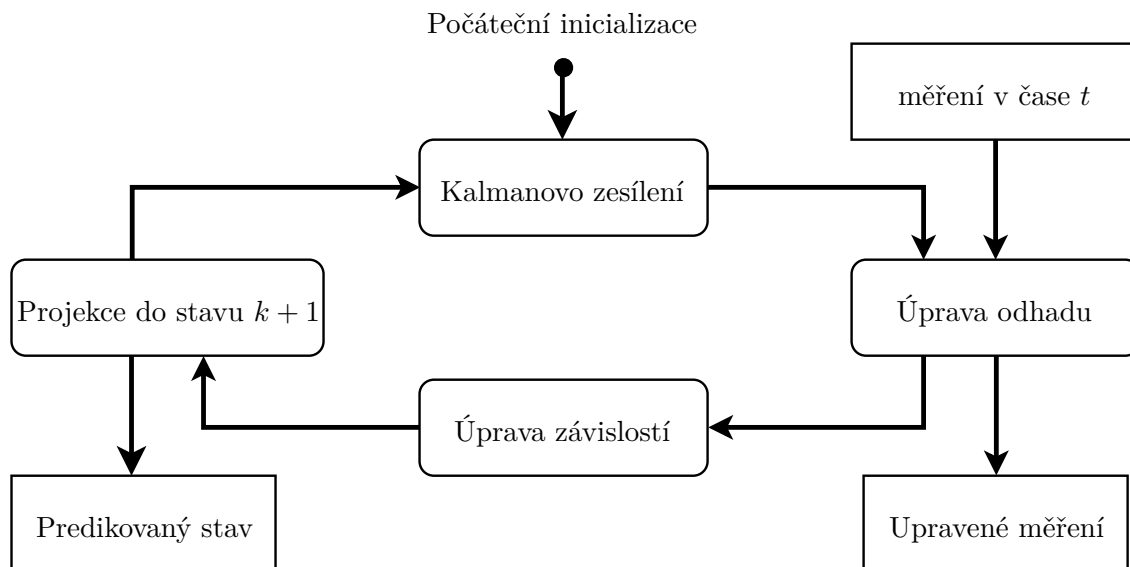
S rozvojem systémů pro autonomní řízení vozidel se detekcí semaforů zabývá mnoho prací, například [33], která využívá pro detekci pozice metody modelování pozadí a pro získání stavu využívá statistického zkoumání přítomnosti určitých barev v zájmových regionech. Jejich postup dovoluje detekovat i obtížnější světelné signály, které umožňují pouze odbočení. Rozdílný způsob navržený v práci [30] využívá pro detekování pozice semaforu ve scéně barevnou segmentaci obrazu a následně pomocí konvoluční neuronové sítě určuje jeho aktuální stav.

Zmíněné práce však pracují se záznamy z jedoucího automobilu, který je v určitých aspektech odlišný od problémů s detekcí semaforu ze stacionární kamery. Zejména se jedná o možnost opravy detekce z následujícího snímku, který je vždy pořízen z odlišného úhlu než ten předešlý a tak některé metody mají možnost korekce vytvořené detekce na základě porovnání více po sobě jdoucích snímků. Stejně tak při pořizování záznamů z jedoucího vozidla se semafor v častých případech nachází na kontrastním pozadí, což zjednodušuje jeho detekování například pomocí tvaru.

Detekce automobilů

Často používanou metodou pro detekci vozidel v záznamu ze stacionární kamery je využití modelování pozadí jako tomu například je v práci [38]. Zvolený způsob řešení však může mít problémy s detekováním automobilů, které se ve snímku překrývají a také s detekcí stínů. Obdobného postupu je využito ve článku [40] pro získávání statistických informací o dopravě pomocí záznamů pořízených z dronu. Navržený způsob je navíc odolný vůči pohybujícímu se pozadí. Též v práci [35] je využito pro detekci automobilů adaptivního modelu pro určení pozadí v kombinaci s metodami umožňující odstranění stínů.

Alternativou pro detekci vozidel je využití neuronových sítí. Takové metody většinou poskytují vyšší odolnost vůči nepřesnostem způsobených stíny. Takový postup je například zvolen v práci [15], kde je využit Fast R-CNN model neuronové sítě.



Obrázek 3.5: Zjednodušené schéma fungování Kalmanova filtru. Převzato z [36].

3.3 Sledování pohybu objektů

Jednotlivé objekty na snímku lze detekovat pomocí metod zmíněných v předešlé sekci 3.2. Sama detekce však popisuje závislosti objektů mezi jednotlivými snímky, a tak je třeba jednotlivé detekované objekty přiřadit tentýž objektům v po sobě jdoucích snímcích.

Cílem sledování objektů, je získání informace o jejich pohybu. Hlavními dílčími úkoly sledování jsou [2]:

- **identifikace objektu** - schopnost nalézt tentýž objekt z předešlého snímku na snímku následujícím a schopnost přidat sledovaný objekt který doposud sledovaný nebyl. Důležitými aspekty jsou také schopnosti detekce ztráty sledování objektu a vypořádání se s okluzí objektu.
- **modelování pohybu** - jelikož prováděná měření detekce nejsou přesná, je zapotřebí využití modelovacích technik, které na základě měření zatížených chybou dokáží tuto chybu snížit a získat tak přesnější informaci o aktuální pozici objektu.

V této sekci bude nejprve popsán algoritmus pro modelování pohybu, následně bude zmíněn algoritmus pro získání optického toku a na závěr bude popsán problém přiřazení naměřených a odhadnutých pozic a jeho řešení pomocí Khun-Munkresovy metody.

Predikce následující pozice modelováním pohybu

Pro vyřešení problému modelování pohybu lze použít Kalmanova filtru [19], který má hned dvě využití. První z nich je, že na základě předešlých měření a popisu modelovaného systému dokáže predikovat stav následující, čímž umožňuje snažší identifikaci téhož objektu pomocí porovnání predikované a naměřené pozice, postup identifikace je více popsán v následující sekci. Druhým využitím je schopnost minimalizovat chyby provedených měření právě na základě srovnání s vlastní predikcí.

Kalmanův filtr je model reálného systému. Využívá poznatků o modelovaném systému, které jsou známy a na jejich základě vytváří predikce dle dříve naměřených hodnot. Pokud

by bylo chování systému dokonale popsáno, nebylo by třeba korekčních měření. Takový model ovšem není reálný, proto je třeba stavy tohoto modelovaného systému upravovat na základě provedených měření [23].

Pro sestavení Kalmanova filtru je třeba znát matematické vztahy popisující reálný systém. V případě popisu pohybu automobilu v obraze využijeme pozice objektu v obraze a hodnotu udávající změnu jeho pozice mezi snímky. Nejprve tedy Kalmanův filtr provede svoji predikci, kdy vytvoří odhad následujícího stavu. Tento odhadnutý stav je poté porovnán s naměřenou hodnotou a je provedena jak úprava naměřených hodnot, tak úprava vnitřních stavů Kalmanova filtru, což ovlivní následující predikce [2]. Zjednodušené schéma popisující fungování Kalmanova filtru je vidět na obrázku 3.5.

Čím přesněji bude matematický popis odpovídat chování popisovaného systému, tím lépe bude Kalmanův filtr provádět svoji predikci. Pro jednotlivá měření lze specifikovat velikost chyby jakou jsou tato měření zatížena, čímž je možnost upravovat váhy vzájemného vztahu mezi predikovaným stavem a provedeným měřením. Modelovaný systém může být jak uzavřený, tak otevřený – v takovém případě je třeba při modelování definovat interakci systému s okolím.

Optický tok

Pro sledování pohybu lze také využít algoritmus pro získání optického toku, navržen v [6]. Ten popisuje změnu polohy zkoumaných pixelů na základě nalezení vlastností okolí daného pixelu z předešlého snímku na snímku následujícím. Tento pohyb může být způsoben jak pohybujícími se objekty ve snímané scéně tak ale také pohybem kamery [2].

Analýzu optického toku lze využít při sledování objektů za účelem získání informace o změně jejich pozice mezi snímky. Využití tohoto postupu je dále popsáno v sekci 4.2.

Řešení problému přiřazení odhadovaných a naměřených pozic

Většinou je třeba v obraze sledovat více objektů najednou. Bylo by také naivní tvrdit, že pořadí detekovaných objektů v předešlém snímku by odpovídalo pořadí detekovaných objektů ve snímku následujícím. Nelze také předpokládat, že po využití metody pro predikci následujících pozic by byly získané výsledky naprosto přesné. Proto je třeba využít algoritmus, který je schopen přiřadit nově detekované pozice objektů k jejich predikovaným pozicím s co možno nejmenším rozdílem jednotlivých vzájemně přiřazených hodnot. Jedná se o takzvaný *assignment problem* (problém ideálního přiřazení) a Khun-Munkresova metoda (také známá jako *hungarian method*) jej řeší.

Vstupem Khun-Munkresovy metody je matice. Sloupce reprezentují predikované pozice a řádky představují nově naměřené pozice. Sestrojená matice obsahuje vzdálenosti jednotlivých bodů. Na základě této matice je Khun-Munkresova metoda schopna přiřadit predikované body k nově naměřeným. Součet rozdílů poloh k sobě přiřazených bodů je nejmenším možným pro zadanou matici. Podrobnější popis fungování Khun-Munkresovy metody je popsán v práci [39].

Kapitola 4

Detekce jízdy na červenou

Jak bylo zmíněno v předešlé kapitole 3, aktuální časté způsoby řešení vyžadují buď vysoké nároky na hardware nebo lidskou anotaci scény v případech, kdy nebylo při pořizování záznamů uvažováno nad jeho využitím pro detekci jízdy na červenou. Z tohoto důvodu neobsahují anotace potřebné pro automatickou kalibraci. V následující kapitole se zabývám návrhem řešení detekce jízdy na červenou na základě automatické kalibrace kamery bez nutnosti interakce s uživatelem. Předpokládá se, že se jedná o záznam ze statické kamery, která je umístěnou nad úrovní vozovky a ve směru jízdy zkoumaných pruhů.

V rámci této kapitoly je nejprve popsán způsob získávání aktuálního stavu semaforu 4.1. Následně je popsán postup detekování automobilů v obraze 4.2 a využití snímání jejich pohybu pro automatickou kalibraci kamery 4.3. V další sekci 4.4 je popsán způsob aproximace jízdnic pruhů a místa pro zastavení za využití detekovaného pohybu automobilů a jejich chování v závislosti na detekovaném stavu semaforu a s využitím kalibrované kamery. V poslední sekci 4.5 je čtenář seznámen se způsobem samotné detekce přestupků s využitím výše získaných informací o snímané scéně.

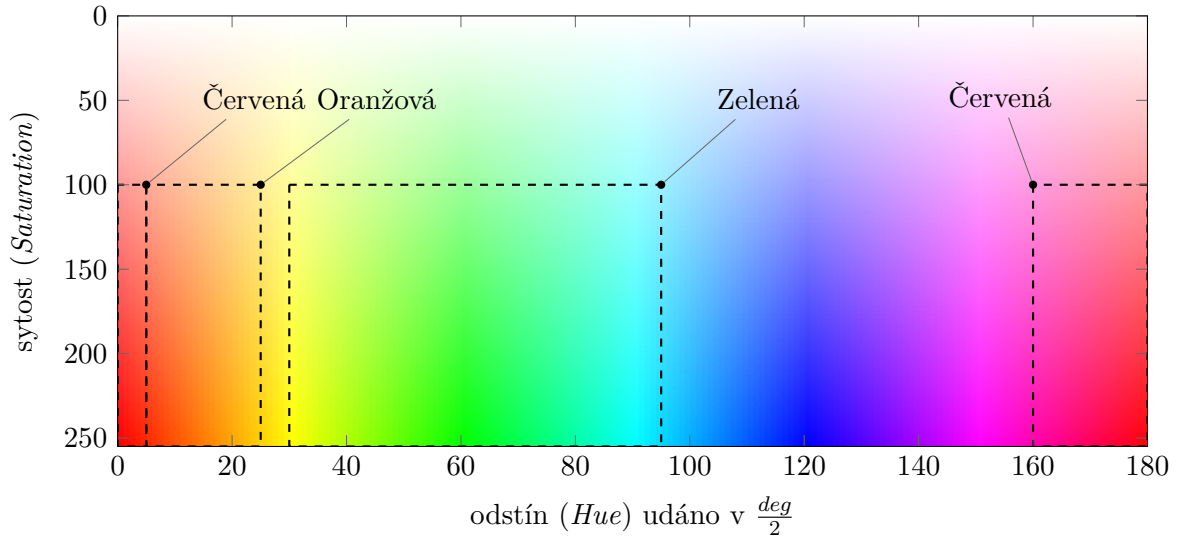
Postupy popsané v následujících sekcích využívají metod zpracování obrazu popsanych v teoretické části a poznatků získaných studiem aktuálních způsobů řešení popsanych v předešlé kapitole 3. Jednotlivé podkapitoly jsou navrženy jako samostatné vzájemně komunikující moduly, díky čemuž je umožněno jejich paralelní zpracování, které bylo následně využité i při implementaci 5.

4.1 Získání stavu semaforu

Umění rozpoznat aktuální stav semaforu je stěžejním prvkem navrhovaného systému. Na určení správného stavu závisí později navrhované metody pro aproximaci místa pro zastavení 4.4. Stejně tak je jeho znalost využita při detekování druhého úběžníku 4.3, tehdy je znalost aktuálního stavu využita pro filtraci totožných nebo velmi podobných snímků. Obsahem této sekce je návrh postupu pro detekci pozice semaforu a následné extrakce jeho stavu.

Lokalizace semaforu ve scéně

Pro lokalizaci umístění semaforů ve scéně je využito konvoluční neuronové sítě. Semafor je statickým objektem ve snímané scéně, tudíž se jeho poloha, velikost ani barva výrazně nemění a je tak třeba, aby použitý model byl dostatečně robustní pro provedení úspěšné detekce. Na rozdíl od detekce automobilů je při detekování semaforu kladen vyšší důraz



Obrázek 4.1: Na obrázku jsou vyznačeny jednotlivé výseče barevného spektra HSV barevného modelu příslušící jednotlivým barvám semaforu. V rámci obrázku není využito hodnoty jasu (*Value*), během detekce stavu se používá rozsah hodnot jasu 10 – 255. Hodnota odstínu je dělena dvěma aby její hodnota byla popsitelná pomocí jednoho bytu.

na přesnost než na rychlost provedení. Důvodem je fakt, že při správné detekci semaforu je třeba ji provést pouze při inicializaci scény. Z tohoto důvodu je pro detekci semaforů využito rozsáhlejší konvoluční neuronové sítě Faster R-CNN. Tato síť je dle statistik [16] při porovnání s SSD modelem, využitým pro detekci automobilů, je tento model výrazně pomalejší avšak dosahuje až dvakrát vyšší přesnosti detekce na témže testovacím datasetu.

Získání stavu

Jakmile je určena přesná pozice semaforu ve scéně, je možné v jednotlivých snímcích získat jeho aktuální stav. Během tohoto procesu je na každém snímku extrahována oblast, která je určena rozměry nalezeného semaforu. Před samotným určením aktuálního stavu je pro každý pixel na snímku nalezeno maximum mezi aktuálním a předešlým snímkem. Díky tomu dochází k odstranění drobných nepřesností způsobených například nevhodně zvolenou vzorkovací frekvencí pořízeného záznamu. Dále je takto sestavený snímek vyhlazen za pomoci využití Gaussova filtru.

Na takto upraveném obrazu je provedeno určení stavu na základě počtu pixelů příslušícím jednotlivým úsekům barevného spektra. Jak již bylo zmíněno v sekci 2.1, tak pro filtraci určité šíře barevného spektra je výhodnější využít HSV barevného modelu, jelikož umožňuje snazší specifikaci šíře zvolené výseče barevného spektra pro filtraci barev. Na obrázku 4.1 lze vidět zvolené jednotlivé barevné úseky pro detekci barev na semaforu, tedy červené, oranžové a zelené. Ukázky filtrace barev na semaforu jsou k vidění na obrázku 4.2.

Po získání počtu pixelů příslušícím k jednotlivým barvám lze v některých případech okamžitě určit aktuální stav. Jedná se o stavy semaforu, ve kterých svítí pouze jedna jeho část a počty pixelů jedné zkoumané barvy tak přesahují zvolenou prahovou hodnotu relativní četnosti. Pro detekci změny stavu, při kterém na semaforu svítí více barev najednou, čímž dochází k nepřesnostem při měření, je třeba využít znalosti chování semaforu. Nepřesnosti jsou zejména způsobeny blízkostí si oranžové a červené barvy v barevném spek-



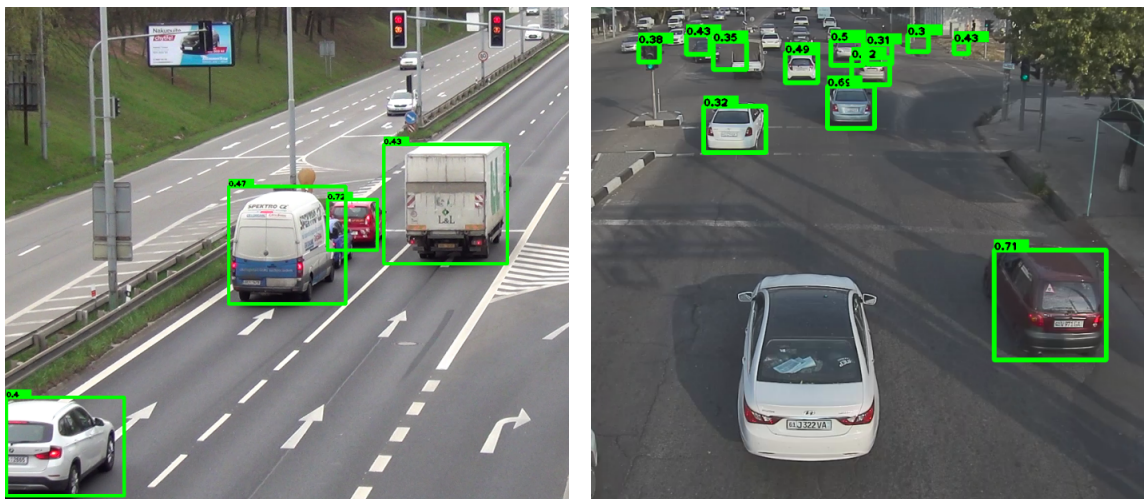
(a) zelená barva lze v barevném spektru bez problémů odlišit.

(b) Ukázka chyb vzniklých blízkostí si oranžové a červené barvy v barevném spektru.

Obrázek 4.2: Ukázka filtrace barevného spektra za účelem detekce stavu semaforu. Vždy zleva: původní obrázek a následně obrázky pro pixely příslušící červené, oranžové a zelené barvě. Lze si všimnout chyb při filtraci vzniklých blízkostí si oranžové a červené barvy v barevném spektru.

tru a nemožnosti určení přesné univerzální hranice jejich oddělení. Jejich hodnota je totiž často deformována chybou při pořizování záznamu často způsobenou nevhodným osvětlením scény. Pro eliminaci zmíněných nepřesností při detekci stavu je využito znalosti chování semaforu na jehož základě a na základě výpočtu relativní četnosti změněných pixelů (vůči počtu pixelů připadajících jednotlivým barvám naměřených pro daný stav) je rozhodnuto, zda došlo ke změně stavu, či nikoliv. Zkoumáním výrazné změny počtu pixelů příslušící jednotlivým barvám je také umožněno určovat stav semaforu ve scéně, kdy jedna jeho část je skryta, případně kdy se část semaforu nachází zcela mimo snímek.

Oba zmíněné postupy pro určení stavu semaforu lze spolu kombinovat, první zmíněný slouží v celém procesu pro počáteční určení stavu a dále jako případný korektor nastalé chybné změny stavu. Ke korekci dochází v případě, kdy relativní četnost pixelů určité barvy přesáhne zvolený práh.



Obrázek 4.3: Vlevo případ, kdy jsou všechna zkoumaná vozidla detekována, vpravo ne-nalezení vozidla na snímku, přestože na předešlém snímku vozidlo nalezeno bylo. Čísla u jednotlivých obdélníků reprezentují ohodnocení dané detekce.

4.2 Sledování pohybu automobilů v obraze

Schopnost sledovat pohyb vozidel se skládá z více dílčích úkonů. Kromě detekce vozidel v jednotlivých snímcích je třeba, aby byl systém schopný k sobě přiřadit objekty nalezené na dvou po sobě jdoucích snímcích. Díky tomu je umožněna analýza jejich pohybu.

V rámci následující sekce bude nejprve popsán způsob detekování automobilů na jednotlivých snímcích. Poté bude popsán způsob pro jejich přiřazení napříč snímky a problémy které je při jejich párování třeba řešit. Závěrem bude navrženo využití metody využívající optického toku pro získání dodatečných informací o pohybu v obraze.

Detekce automobilů

Pro nalezení automobilů ve snímku jsem zvolil, stejně jako pro detekci semaforu, metodu využívající konvoluční neuronové sítě. Konkrétně se jedná o model typu SSD který, jak již bylo zmíněno dříve, je výrazně rychlejší než model využitý pro detekci semaforů, zároveň ovšem stále poskytuje dostatečnou přesnost detekce. Vozidla se v rámci záznamu pohybují, čímž se na jednotlivých snímcích ocitají vždy v mírně odlišných pozicích s odlišnou velikostí a osvětlením. Díky tomu je možné nalezení nedetekovaných objektů v předešlém snímku na snímcích následujících. Snímání automobilů z různých úhlů při různém osvětlení však má za následek také jev, že některá vozidla nemusí být na všech snímcích detekována, jak lze vidět na obrázku 4.3. Je tedy třeba aby metody použité pro sledování vozidel byly schopny se s tímto nedostatkem vyrovnat.

Sledování automobilů

Nalezením automobilů v obraze je sice získána pozice jednotlivých vozidel, která ovšem neposkytuje žádné informace o jejich chování mezi jednotlivými snímky. Pro možnost analyzovat jejich pohyb je třeba, aby byl systém schopný přiřadit vozidla z předešlého snímku k totožným vozidlům ve snímku následujícím. Za tímto účelem je využito Khun-Munkresovi metody řešící problém přiřazení na základě vzdálenosti pozic detekovaných vozidel na pře-

dešlém a aktuálním snímku. S tím ovšem souvisí následující problémy s přiřazením jednotlivých vozidel:

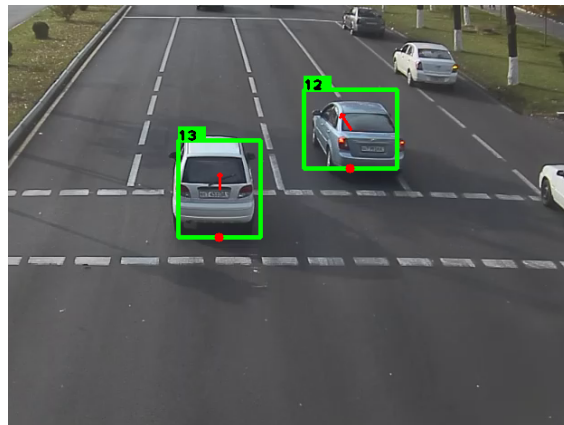
- **Pohyb automobilů** - jednotlivá vozidla mohou v případě nízké frekvence pořizování snímků mít i na dvou po sobě jdoucích snímcích výrazně odlišnou polohu.
- **Nepřesnost detekce** - pozice nalezených vozidel jsou zatíženy nepřesností měření.
- **Nedetekování vozidel** - jak již bylo zmíněno v předešlé části, detektor může na některých snímcích detekovat pouze určitou část z veškerých vozidel, která se na snímku nachází.
- **Problém přiřazení** - na následujícím snímku mohou být nově detekována vozidla, která se na předchozím snímku nenacházela, nebo mohou v následujícím snímku opustit snímanou scénu.

Pohyb automobilů Pro predikci pozice automobilu na následujícím snímku je využito Kalmanova filtru, ten využívá matematického popisu chování pohybu vozidel v obraze na jehož základě a na základě předešlých měření provede predikci vozidla v následujícím snímku. Úroveň přesnosti takové predikce závisí na přesnosti popisu chování. Ukázka predikování následujícího stavu je viditelná na obrázku [4.4b](#).

Nepřesnost detekce Kalmanova filtru lze využít i při řešení následujícího bodu, zde je jeho predikovaný stav využít pro porovnání s nově provedeným měřením. Na základě specifikace chyby měření a predikovaného stavu je provedena úprava naměřené pozice.

Nedetekování vozidel Kalmanův filtr by našel své využití i tomto bodě. Nutné je ovšem uvažovat o případech, kdy automobil není detekován i při více po sobě jdoucích snímcích. Je třeba mít na paměti, že modelování pohybu vozidel nebude nikdy dokonalé, tudíž s přibývajícím počtem predikcí bez provedení korekčního měření bude i vzrůstat chyba takto vytvořených predikcí. Tato chyba je například nejvíce patrná v případech, kdy je automobil naposledy detekován před zastavením. Tehdy je uchována informace o jeho aktuálním pohybu, která slouží při dalších predikcích, přestože na snímku tento automobil již zastavil. Takto vytvořené predikce se poté budou viditelně rozcházet s realitou a bude docházet k chybně detekovaným přestupkům.

Řešením je využití informací získaných analýzou optického toku v obraze. Je možné na základě pozic automobilů v předešlém obraze vytvořit zájmové pole obsahující pouze optický tok příslušící danému vozidlu mezi jednotlivými snímky viz obrázek [4.4a](#). Tato informace je poté předána Kalmanovu filtru jakožto další měření. Díky tomu jsou predikce prováděny vždy pouze pro následující snímek a nedochází tak ke zmíněným chybám. Nevýhodou tohoto postupu je náchylnost na libovolný pohyb v příslušné oblasti způsobený například chodcem přecházejícím před automobilem.



(a) Využití optického toku pro měření pohybu vozidel v obraze.

(b) Pomocí modelování pohybu je Kalmanův filtr schopen provádět predikci následující pozice.

Obrázek 4.4: Optického toku z oblastí na nichž se nachází vozidla lze využít pro měření pohybu vozidel v obraze.

Problém přiřazení Při provedení přiřazení automobilů z předešlého snímku na aktuálním snímku je třeba uvažovat nad faktem, že automobilů může být ve scéně reálně jiné množství, než tomu bylo na snímku předešlém. Tento problém je řešen v rámci Khun-Munkresovy metody, která je schopna provést přiřazení i v případech, kdy jejím vstupem není čtvercová matice, tedy v případech, kdy se liší počty přiřazovaných objektů v jednotlivých množinách. Khun-Munkresova metoda však není sama schopna odhalit případy, kdy počet automobilů v předešlé a zkoumané scéně je totožný (pouze pro názornost, jelikož tento problém může nastat i v případě, kdy počet vozidel je odlišný), avšak kdy se nejedná o tatáž vozidla – některá vozidla opustila scénu a nová ve scéně přibyla. Khun-Munkresova metoda v takovém případě nemá možnost, jak tento fakt detekovat a provede přiřazení pozic stejným způsobem, jakoby naměřené hodnoty připadali totožným vozidlům. Výsledkem je chybná identifikace vozidel.

Na řešení tohoto problému navrhuji využít znalosti velikosti detekovaných automobilů na předešlém snímku. Snímky v záznamu jsou perspektivní projekcí snímané scény, tudíž objekty, které jsou od pozorovatele více vzdálené se jeví jako menší oproti objektům stejné velikosti, které jsou pozorovateli blíže. Této znalosti lze využít pro určení hranice maximální vzdálenosti dvou přiřazovaných pozic, po kterou je systém ochoten tyto dvě pozice považovat za tentýž objekt. Pro větší objekty lze předpokládat, že se nachází blíže ke kameře, tudíž jejich rozdíl pozic na snímku bude vyšší než rozdíl u objektů, které jsou uživateli více vzdáleny. Velikost detekovaného objektu je tedy přímo úměrná tomu, jak velký rozdíl pozic je akceptovatelný pro jejich přiřazení. V případě, kdy je tato hranice pro danou dvojici pozic překročena, je do vstupní matice Khun-Munkresovy metody na dané místo vložena informace, která znemožňuje této metodě provést přiřazení dané dvojice pozic. Nově detekované pozice, které nebyly přiřazeny k žádným z pozic předešlých, lze považovat za nově detekovaná vozidla.



Obrázek 4.5: Ukázka pozice a úhlu kamery vůči snímané vozovce.

4.3 Kalibrace kamery

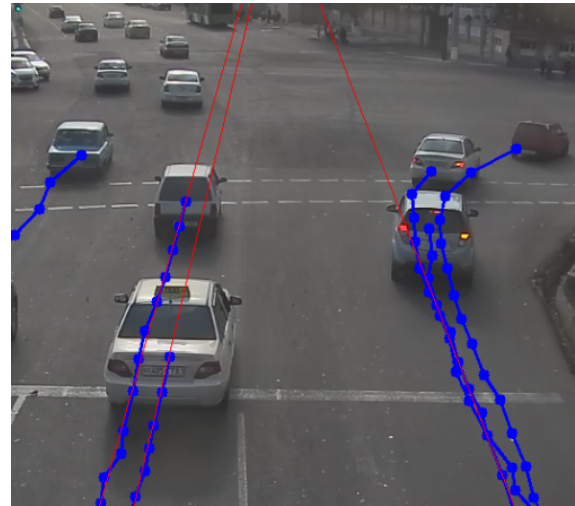
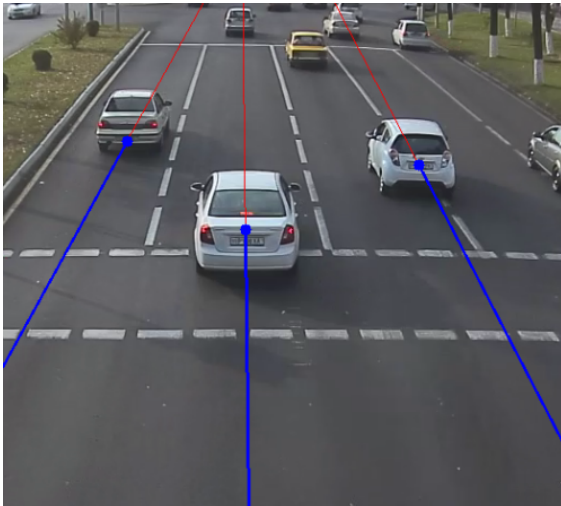
Pro možnost analyzovat scénu z pohledu jejího rozložení, tedy možnost detekovat jednotlivé jízdni pruhy a linie pro zastavení, je třeba nejprve získat bližší informace o pozici pozorovatele (kamery) vůči snímané scéně. Během této fáze jsou využita nalezená a sledovaná vozidla z předešlé části. Pro úspěšný výpočet se předpokládá, že kamera je umístěna v dostatečné výšce vůči vozovce a je zvolen vhodný úhel který snímá s vozovkou. Kalibrace kamery je provedena nalezením tří dominantních úběžníků v obraze, které definují 3 ortogonální směry ve sledované scéně. Kdy první nalezený směr určuje směr pohybu vozidel, druhý je kolmý vůči tomuto pohybu. Zároveň jsou oba nalezené směry paralelní vůči rovině definované povrchem vozovky. Zmíněné postupy kalibrace v této sekci byly představeny v práci [10]. Způsob detekce úběžníku na základě paralelních linií je popsán již dříve 3.1. V rámci následujících částí budou popsány postupy pro nalezení jednotlivých linií příslušících daným směrům v obraze.

Nalezení prvního úběžníku

Pro nalezení prvního úběžníku je využito sledování pohybu vozidel. Jelikož se jedná o záznam z kamery, která často bývá umístěna v těsné blízkosti křižovatky, kde v záběru před vjezdem do křižovatky není příliš manévrovacího prostoru je předpokládáno, že jednotlivá vozidla směřují svůj pohyb směrem k prvnímu úběžníku. Ukázka snímané scény je vidět na obrázku 4.5.

Navržená metoda pro získání prvního úběžníku využívá pro sestrojení přímky první a poslední pozice sledovaného vozidla. Tato metoda dosahuje obstojných výsledků na křižovatkách, kde většina vozidel pokračuje skrze křížení rovně, jak je vidět na obrázku 4.6a. Pomocí metody RANSAC se následně ze získaných trajektorií nalezne jejich aproximovaný průsečík, viz obrázek 4.7.

V rámci experimentů bylo zjištěno, že odbočující vozidla značně ovlivňují kvalitu zmíněné metody. Se zvyšujícím se počtem odbočujících vozidel výrazně stoupá riziko chybné detekce prvního úběžníku. Proto jsem na základě experimentů provedl úpravy vedoucí k metodě, která reprezentuje trajektorii vozidla pomocí více bodů. Dominantní směr vozidla, tedy ten jež by měl směřovat k prvnímu úběžníku, je následně vypočten též za využití metody RANSAC, která tuto přímku aproximuje z bodů tvořících trajektorii daného vozidla. Metoda RANSAC přikládá nalezené přímce určitou kvalitu na základě bodů, které se parti-



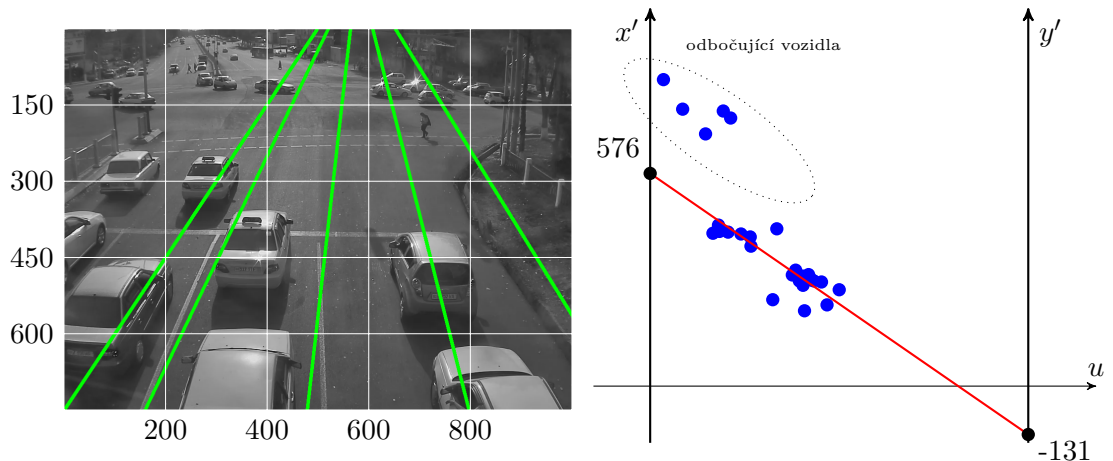
(a) Metoda využívající pouze startovní a poslední pozici.

(b) Metoda odolná vůči odbočujícím vozidlům využívající RANSAC pro aproximaci přímky.

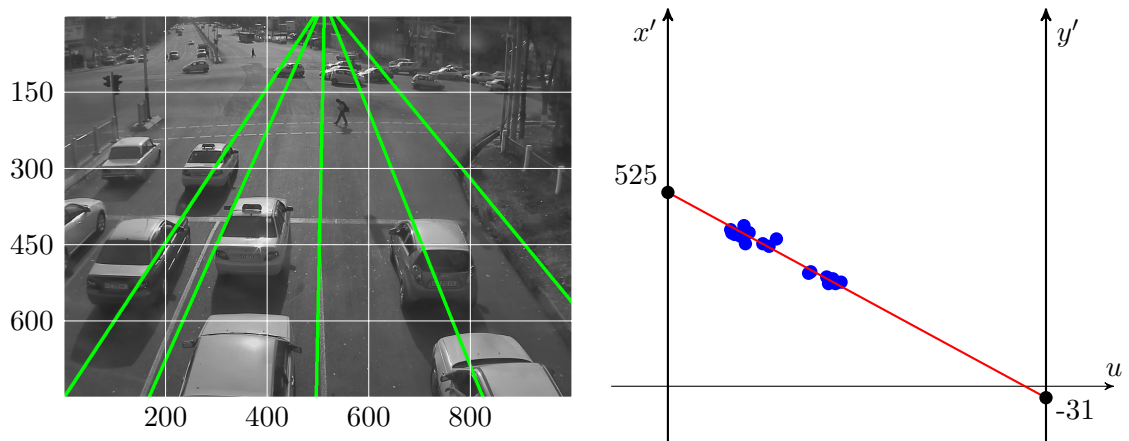
Obrázek 4.6: Porovnání metod pro získání přímek pro detekci prvního úběžníku z trajektorií vozidel.

cipovali na jejím zvolení. Kvalita nalezené přímky je následně určujícím parametrem, zda-li se má přímka využít pro detekci úběžníku. V případě, kdy není možné v pohybu automobilu nalézt dostatečně ohodnocenou přímku, se daný pohyb neúčastní v detekci úběžníku. Tento jev je patrný na obrázku 4.6b, kde z trajektorií některých vozidel nebyly sestrojeny červené přímky vedoucí k prvnímu úběžníku.

Porovnání kvality obou metod na křižovatce, kde je zmíněný problém s vysokým počtem odbočujících vozidel lze vidět na obrázcích 4.6.

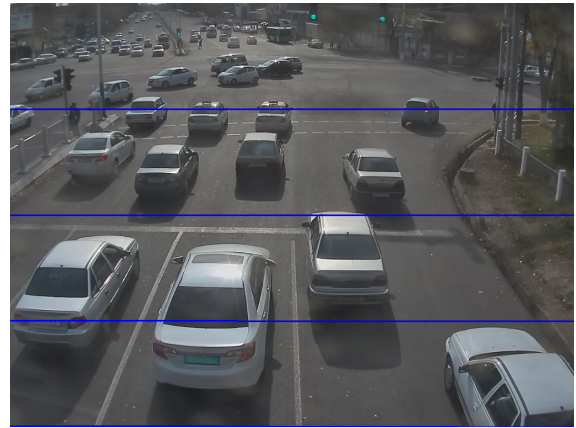
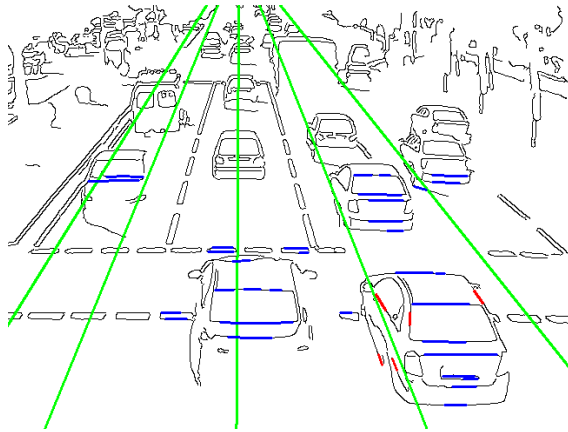


(a) Nalezený úběžník pomocí metody využívající pouze první a poslední pozici vozidla. Na obrázku vpravo lze vidět, že některé přímky jsou evidentně chybně detekované. Většinou se jedná o odbočující vozidla.



(b) Upravený algoritmus využívající více bodů k reprezentaci trajektorie.

Obrázek 4.7: Rozdílné způsoby sestrojení trajektorií vozidel a jejich využití pro nalezení prvního úběžníku. Vlevo vždy obrázek zachycující detekovaný úběžník na snímku (průsečík zelených čar). Vpravo detekovaný úběžník pomocí metody RANSAC v paralelním souřadnicovém systému.



(a) Pro hledání druhého úběžníku nejsou využity hrany automobilů směřující k prvnímu úběžníku (červené). Přímký jsou detekovány pouze na hranách automobilů a jejich blízkém okolí.

(b) Detekovaný druhý úběžník ve snímaných scénách křižovatek mívá druhý úběžník často x -souvadnicí blízko nekonečna. Modré čáry se protínají v druhém úběžníku.

Obrázek 4.8: Detekce druhého úběžníku pomocí akumulovaných hran automobilů které ne

Nalezení druhého úběžníku

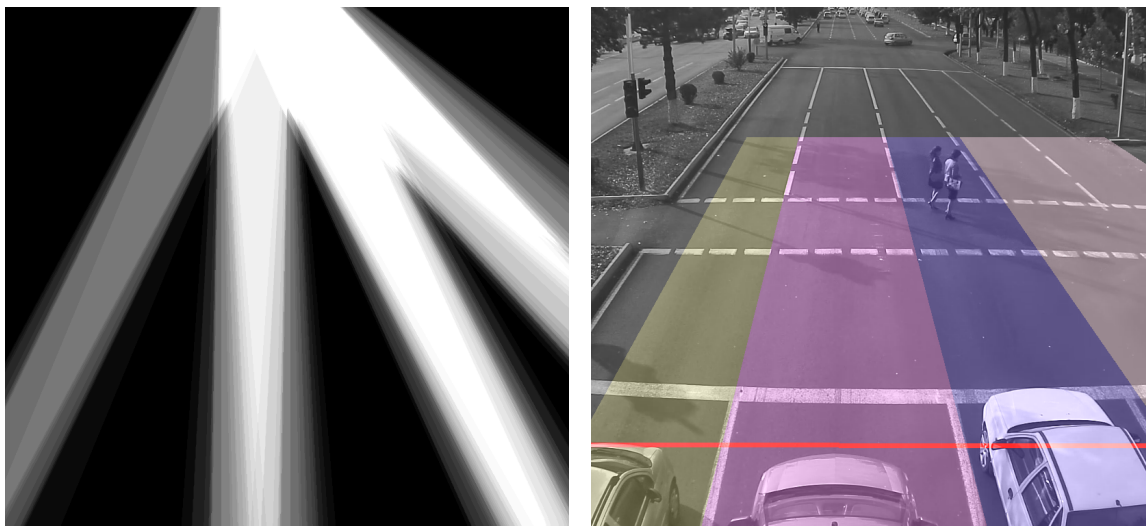
Pro nalezení druhého úběžníku je třeba ve scéně nalézt linie, které jsou reálně kolmé vůči směru jízdy. Nejvíce takovýchto přímek se nachází na hranách vozidel samotných. Pomocí znalosti velikosti vozidel a jejich pozice je možné detekovat hrany na snímku pouze v oblastech, na kterých se jednotlivá vozidla nachází. Extrahování přímek poté spočívá ve vyhlazení obrazu za využití nízkofrekvenčních filtrů. Následně je provedena detekce hran v obraze, jejíž výstup je předán Houghově transformaci pro nalezení přímek. Vozidla ovšem často obsahují také přímký směřující k prvnímu úběžníku, takové jsou pro výpočet odstraněny. Postup detekce druhého úběžníku je viditelný na obrázku 4.8. Po získání dostatečného počtu hran je provedena detekce druhého úběžníku za využití stejného algoritmu jako pro detekci prvního. Během detekce druhého úběžníku jsou využity i informace o aktuálním stavu semaforu. Většina automobilů během povelu „Stůj!“ totiž nemění svoji pozici, čímž by docházelo k vícenásobné detekci téže hrany. Z toho důvodu není detekce druhého úběžníku nad takovými snímky prováděna.

4.4 Segmentace oblastí vozovky

V rámci sekce zabývající se segmentací vozovky v obraze bude nejprve popsán způsob pro nalezení jízdních pruhů. Díky jejich znalosti je provedena aproximace místa pro zastavení, která je popsána závěrem této sekce. Navržené postupy využívají jak prvního tak druhého nalezeného úběžníku z předešlé sekce. Stejně tak využívají detekovaný pohyb automobilů ve scéně a znalost stavu semaforu. Detekované automobily jsou popsány pomocí obdélníků tudíž metody zde představené vždy budou obsahovat chybu při výpočtu.

Nalezení jízdních koridorů

Na základě získaných trajektorií automobilů v obraze je možné sestavit masku jejich pohybu. Po obdržení dostatečného počtu trajektorií je možno detekovat pomocí této masky jízdní pruhy. V rámci detekce se vyhází z předpokladu, že kamera je umístěna vhodně nad



(a) Maska jednotlivých pruhů získaná díky trajektorii vozidel.

(b) Aproximované jízdní pruhy a čára pro zastavení.

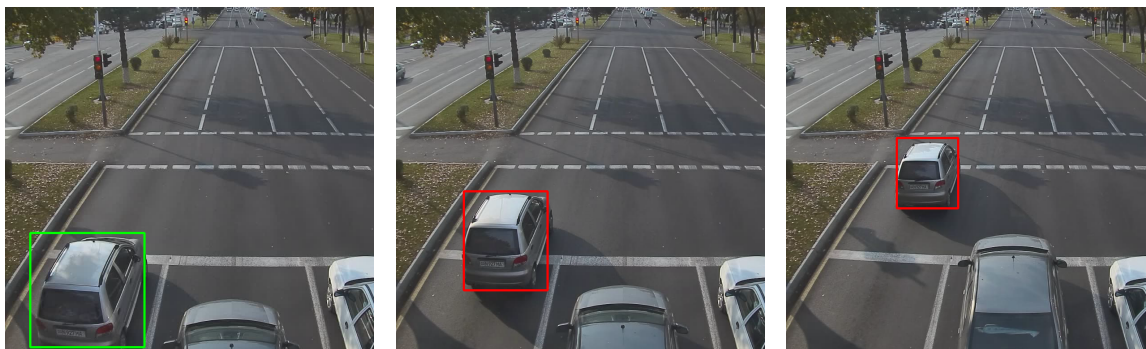
Obrázek 4.9: Ukázka využití trajektorií vozidel pro aproximaci jízdních pruhů. Linie pro zastavení (viditelná na obrázku vpravo) je aproximována metodou RANSAC ze získaných pozic vozidel při přechodu semaforu ze stavu „Pozor!“ na stav „Volno!“.

vozovkou a že první úběžník se tak nachází v horní polovině obrazu. Následně je získaná maska vyhlazena a zbavena šumu pomocí filtrů. Nad takto upravenou maskou je provedeno prahování na základě vhodně zvolené hodnoty, která umožňuje eliminaci chybně detekovaných trajektorií – například automobilů, které v rámci křižovatky nepokračovaly směrem k prvnímu úběžníku. Na takto upraveném binárním snímku jsou poté detekované jednotlivé jízdní koridory, při jejichž detekci je využito prvního úběžníku – přímky ohraničující jízdní koridory směřují k prvnímu úběžníku. Následně jsou oblasti mezi detekovanými pruhy jsou rovnoměrně vyplněny jelikož se předpokládá, že dané jízdní pruhy na sebe navazují. Masky získaných trajektorií a výsledné detekované jízdní pruhy jsou viditelné na obrázku 4.9b.

Aproximace místa pro zastavení

Detekce jízdních pruhů dále umožňuje aproximovat místo pro zastavení. K jeho odhadnutí je využito znalosti aktuálního stavu semaforu. Při změně ze stavu „Pozor!“ na stav „Volno!“ jsou získány pozice všech sledovaných bodů automobilů ve scéně. Ty jsou následně přiřazeny nalezeným jízdním pruham, díky čemuž je umožněno získat první automobil v řadě za čarou pro zastavení v každém jízdním pruhu. Tento postup se opakuje, dokud není získáno dostatečné množství detekovaných pozic. Po jejich obdržení je využito modifikovaného algoritmu RANSAC, kde každá přímka je proložena vždy druhým úběžníkem a jedním z detekovaných míst zastavení. Díky tomu je detekovaná čára pro zastavení vždy kolmá ke směru jízdy.

Je třeba podotknout, že navržená metoda funguje za předpokladu, že většina řidičů dodržuje pravidla místa pro zastavení. Úspěšnost aproximace místa pro zastavení je tedy přímo úměrná poměru řidičů, kteří dodržují místo pro zastavení.



Obrázek 4.10: Ukázka úspěšné detekce jízdy na červenou (během signálu „Pozor!“ se řidič má pouze připravit k jízdě).

4.5 Detekce jízdy na červenou

Jakmile jsou známy výstupy metod popsaných výše v této sekci, tak je možné provádět samotnou detekci přestupku jízdy na červenou. V každém zkoumaném snímku je provedena kontrola aktuálního stavu semaforu. V případě kdy stav semaforu je rozdílný od stavu „Volno!“, jsou pozice detekovaných automobilů porovnány vůči poloze čáry pro zastavení. V případě detekce jízdy na červenou je daný detekovaný automobil označen. Tyto údaje jsou dále využitelné pro statistické účely tak i pro analýzu získaného krátkého videa zachycující pouze konkrétní přestupek konkrétního oanotovaného vozidla. Na obrázcích 4.10 lze vidět posloupnou sekvenci snímku detekujících jízdu na červenou.

Kapitola 5

Implementace a vyhodnocení

Tato kapitola se zabývá představím vlastního postupu implementace navrženého způsobu řešení z předešlé kapitoly 4. Následně provedu zhodnocení dosažených výsledků. Závěrem této kapitoly navrhnou možnosti dalšího vývoje této práce, nebo možného navázání na tuto práci.

5.1 Využití knihovny

Příložené programy implementující návrh z předešlé kapitoly jsou napsány pomocí programovacího jazyka Python verze 3.7.1, tato verze byla zvolena z důvodu kompatibility s ostatními využitými knihovnami. Mimo standardních knihoven bylo v rámci práce využito knihovny `OpenCV`¹ verze 4.1, která poskytuje řadu metod využitých pro zpracování obrazu. Pro práci s maticemi bylo využito `numpy`² knihovny verze 1.16. Dále bylo využito knihovny `tensorflow`³ verze 1.13 pro implementaci neuronové sítě využitě pro detekování semaforů a automobilů v obraze. Pro výpočet Khun-Munkresova algoritmu řešících problém přiřazení bylo využito knihovny `munkres`⁴ verze 1.1.

5.2 Návrh programu

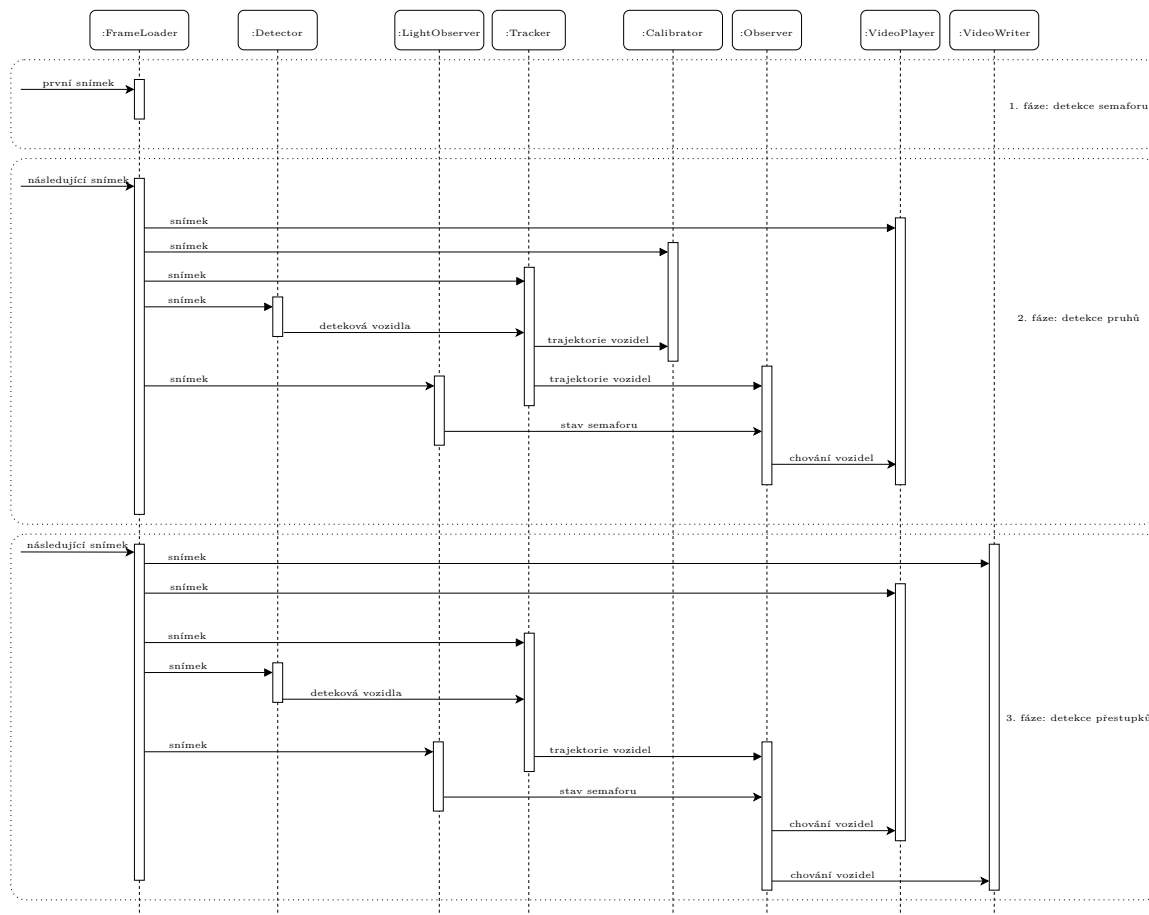
Jelikož metody využívané pro zpracování obrazu jsou výpočetně náročné, je program navržen jako sada paralelně zpracovávaných částí, které si navzájem předávají potřebné výsledky. Každá část je představována jednou třídou. Každá třída má pro svůj výpočet dedikované vlastní vlákno. Pro vytváření vláken je využito standardní knihovny `multithreading`. Vzájemná komunikace jednotlivých částí je zajištěna pomocí synchronních front implementovaných v knihovně `queue`. Dále je navržena třída `InputInfo` uchováující potřebné informace ohledně analyzovaného videa (jedná se například o jeho rozměry, nebo také detekované úběžníky), reference na její instanci se nachází ve většině implementovaných tříd. Běh programu se dělí na více fází, nejprve dochází ke kalibraci kamery – v rámci ní provádí navržený systém kalibraci implicitně automaticky, avšak poskytuje také možnost ručního zadání umístění semaforů a jízdních pruhů pro danou scénu. Následně již funguje systém

¹OpenCV - Open source computer vision, více viz <https://opencv.org/>

²NumPy - viz <https://www.numpy.org>

³TensorFlow - otevřený software pro strojové učení, viz <https://www.tensorflow.org/>

⁴munkres - viz <http://software.clapper.org/munkres/>



Obrázek 5.1: Zjednodušené schéma komunikace mezi jednotlivými částmi programu v rámci jednotlivých fází. Fáze číslo 2 se iterativně opakuje, dokud nejsou detekovány jízdní pruhy a místo pro zastavení. Fáze číslo 3 se opakuje, dokud lze načíst následující snímek. Každá fáze před svým výpočtem posouvá čtecí hlavu na začátek vstupního souboru.

ve fázi detekce, kdy dochází k detekci jízd na červenou. Následuje stručný popis chování jednotlivých tříd:

PipeBlock Jedná se o básovou třídu, která představuje jednu výpočetní část programu. Obsahuje metody umožňující vzájemnou komunikaci instancí tříd, které z ní dědí. Obsahuje též abstraktní metody pro specifické chování které má být provedeno před spuštěním samotného výpočtu `_before()`, během každého výpočetního kroku `_step()` a před ukončením výpočtu v dané části `_after()`. Dále je implementována třída `ThreadedPipeBlock`, která se liší tím, že metoda, která se stará o „nekonečný“ běh programu `_run()`, je umístěna do odlišného vlákna. Uchovává též informaci o aktuální módu zpracování, jelikož vzájemná interakce jednotlivých částí se liší při kalibrování scény a při následné detekci – kdy již některé části není třeba. Při vzájemné komunikaci je též tento uchovaný mód delegován příjemci. Při změně módu je invokována abstraktní metoda `_mode_changed()`. Všechny dále zmíněné třídy dědí z jedné ze zde zmíněných tříd, čímž je umožněna vzájemná komunikace jejich instancí, jak je patrné na obrázku 5.1.

FrameLoader Se stará o načtení jednotlivých snímků ze zadaného souboru. Jednotlivé snímky jsou poté předány ostatním částem pro další výpočet, viz obrázek 5.1. V rámci chování této třídy je také prováděno podvzorkování snímků v případě, že frekvence je příliš vysoká. Vyšší frekvence by měla dopad jednak na delší dobu zpracování celého vstupního videa, jednak na nemožnost detekování dostatečného pohybu mezi jednotlivými snímky pro využití optického toku pro zpřesnění sledování automobilů viz sekce 4.2. Při invokaci metody `_mode_changed()` dochází k posunu čtecí hlavy vstupního souboru zpět na jeho začátek tak, aby detekce přestupků byla prováděna nad celou dobou záznamu.

Detector Se stará o detekci automobilů na obdržných snímcích. Pro detekci automobilů je využito modulu `ObjectDetectionAPI`⁵ konkrétně modelu `SSD_mobilenet_v2` natrénovaného na COCO⁶ datasetu. V rámci této třídy je využito knihovny `tensorflow` pro načtení grafu a vah přechodů ze zmíněného modelu a následně využití načteného grafu pro provedení výpočtu nad konkrétním snímkem. Jelikož je využito neuronové sítě, která je natrénovaná na datech obsahujícím anotace pro větší množinu tříd, než-li je potřeba v této práci, tak je v rámci této třídy též provedena filtrace detekovaných tříd, které nejsou v rámci výpočtu dále třeba.

Tracker Stěžejní funkcí této části je identifikování jednotlivých vozidel napříč snímky. Pro každý krok výpočtu je zde prováděno porovnání detekovaných vozidel na daném snímku s polohou vozidel na snímku předešlém. Na základě navržených metod pro sledování automobilů 4.2 je následně provedeno jejich vzájemné přiřazení případně vytvoření nových instancí. Každý detekovaný automobil je reprezentován pomocí instance třídy `TrackedObject` která obsahuje Kalmanův filtr pro eliminaci chyby měření a též pro možnost aproximace pozice automobilu na následujícím snímku. Pro získávání dat o pozici vozidla využívá detekovaných vozidel a analýzou optického toku pomocí pomocné třídy `OpticalFlow`.

Calibrator Jedná se o část, která je aktivní pouze v módu kalibrace. Cílem této části je nalezení dominantních úběžníků v obraze pro kalibraci kamery. Postup řešení odpovídá navrženému postupu 4.3. Následně je provedena detekce jízdních pruhů též na základě získaných trajektorií vozidel a aproximace místa pro zastavení dle návrhu 4.4.

TrafficLightObserver Na základě detekovaných statických pozice semaforů ve scéně provádí detekce aktuálního stavu semaforu analyzováním barevného spektra jednotlivých výsečí obrazu dle metody navržené v 4.1. Pro určení stavu se využívá pouze semafor s nejvyšším počtem pixelů odpovídající zvoleným barvám. Změna stavu je zaznamenána pouze v případě, kdy je nový stav detekován na více po sobě jdoucích snímcích. Tím dochází k eliminování chybných detekcí vzniklých na základě změny osvětlení, nebo nevhodnou frekvencí snímání.

Observer Za využití znalosti pozic a trajektorií vozidel a aktuálního stavu semaforu určuje, zda-li došlo k jízdě na červenou. V rámci módu kalibrace akumuluje pozice vozidel v jízdních pružích pro detekci místa pro zastavení.

⁵https://github.com/tensorflow/models/tree/master/research/object_detection

⁶COCO - Common Objects in Context viz <http://cocodataset.org>

VideoPlayer a ViolationWriter Jedná se o třídy starající se o vizualizaci výsledků. **VideoPlayer** též umožňuje interakci s uživatelem s možností předčasného ukončení provádění detekce na záznamu. **ViolationWriter** zapisuje získané statistiky detekovaných průjezdů. Též provádí zápis krátkých video sekvencí zachycujících daný přečin společně s anotací pozic zkoumaného vozidla na jednotlivých snímcích.

5.3 Úspěšnost použitých metod

Experimenty byly prováděny na operačním systému *macOS* (2.7 GHz Intel Core i5, 8 GB operační paměti) bez využití grafické karty pro akceleraci výpočtu. Na zmíněném zařízení docházelo ke kompletní inicializaci scény (nalezení úběžníků, jízdních pruhů a aproximace místa pro zastavení) po ~3 minutách, avšak tato doba vysoce závisí na počtu průjezdů vozidel jelikož implementovaný postup je na počtu vozidel při kalibraci naprosto závislý.

V rámci této sekce představím vyhodnocení úspěšnosti výstupů dílčích částí programu. Jedná se o schopnost:

- lokalizovat pozici semaforu ve scéně,
- určit aktuální stav semaforu,
- úspěšného zachycení průjezdu vozidel,
- detekce jízd na červenou.

V rámci experimentů byly využity záznamy s dostatečným počtem průjezdu automobilů pro automatickou detekci úběžníků a jízdních pruhů. Navržený program je ovšem též využitelný i v případech, kdy je využito ruční kalibrace. například při nevhodně zvoleném úhlu kamery nebo v oblastech kde je vozkova ve snímané scéně členitá – což znemožňuje její automatickou kalibraci.

Experimentování lze rozdělit na dvě logické části. V první části bylo zkoumány jednotlivé průjezdy vozidel, zda-li jsou zaznamenány ve správném čase. V druhé fázi byla předmětem zkoumání samotná schopnost detekování průjezdu vozidel na červenou – jelikož takových přestupků, nebylo na zkoumaných záznamech vysoké množství byla zde monitorována i jízda na červenou. Výstup programu však tyto dva činy odděluje a tak možné analyzovat průjezdy na červenou odděleně od průjezdů na oranžovou.

Lokalizace pozice semaforu ve scéně

V rámci zkoumaných 19ti záznamů byl v 16ti případech semafor úspěšně lokalizován. Což tvoří úspěšnost lokalizace přesahující 80%, 2 ze 3 záznamů na nichž nebyl semafor lokalizován byly pořízeny v noci. Využitý model neuronové sítě poskytuje dostatečnou přesnost detekce semaforu na záběrech pořízených během dne. Navržené metody jsou též odolné vůči chybné detekci v případě, že se ve scéně nachází větší počet využitelných semaforů. V takovém případě se při určení stavu využívá semafor s nejvyšším počtem pixelů příslušící zvoleným barvám. Využití všech detekovaných semaforů pro eliminaci chybné detekce lze vidět na obrázku 5.2.

K chybnému určení pozice dochází zejména v případech kdy je záznam pořízen za špatných světelných podmínek, kdy je zachycené světlo na snímku rozptýleno i mimo samotný semafor. Dochází tím k narušení charakteristických poznávacích tvarových i barevných rysů, což znesnadňuje jeho detekci. Problém je nejvíce patrný na záznamech pořízených v noci,



Obrázek 5.2: Eliminace chybně detekovaného semaforu využitím ostatních detekcí.

kdy je semafor jedním z dominantních zdrojů světla v pozorované scéně. Řešením by bylo vytvořit testovací data obsahující též záběry a anotace semaforů v nočních a horších světelných podmínkách.

Určení aktuálního stavu semaforu

Vyhodnocení schopnosti správně určit aktuální stav semaforu byla prováděna na záznamech, kde došlo k úspěšné detekci alespoň jedné pozice. Přesnost určení jeho stavu je určena poměrem snímků, na kterých je stav detekovaný programem odlišný od anotovaného stavu, vůči celkovému počtu snímků v záznamu. Naměřené hodnoty jsou vždy zatíženy chybou, která vzniká při ruční anotaci změn stavů. Vyhodnocení úspěšnosti je viditelné v tabulce 5.1. Na některých záznamech, docházelo k jevu, kdy nebyl rozpoznán přechodu ze stavu „Stůj!“ na stav „Pozor!“, během kterého svítí jak červené, tak oranžové světlo. Důvodem bylo příliš specifické osvětlení pozorované scény, na samotnou detekci jízdy na červenou tento jev však vliv nemá, jelikož průjezd zastavovací čarou v obou případech je klasifikován jako jízda na červenou.

	počet snímků	chybné detekce	úspěšnost
ideální podmínky	43262	1335	97%
pouze část semaforu	17090	948	94%

Tabulka 5.1: Úspěšnost určení aktuálního stavu semaforu. Zaokrouhлено na celá procenta

Dle tabulky provedených experimentů lze usoudit, že po provedení úspěšné lokalizace semaforu ve scéně je při záběrech pořizovaných během dne navržená metoda pro určení aktuálního stavu dostatečně přesná. Úspěšnost určení aktuálního stavu vždy překračují hranici 90% a to včetně záznamu, kde je semafor viditelný pouze z části a je třeba tak určovat aktuální stav pouze ze znalosti stavů dvou jeho světél.

Úspěšnost detekce průjezdu vozidel

V rámci experimentování bylo nejdříve provedeno ohodnocení schopnosti detekovat průjezdy jednotlivých vozidel viz tabulka 5.2. Průjezd vozidla je detekován pouze v místě aproximované zastavovací čáry. Jízdní pruhy, které na úrovni zastavovací čáry zasahují i mimo zkoumaný obraz nejsou oblastí zkoumaní. Číslo snímku každého detekovaného průjezdu je porovnáno s ručně anotovaným průjezdem vozidel. Je stanoven práh počtu snímků, který určuje toleranci vzájemného přiřazení dvojice. V opačném případě je určeno, zda-li se jedná o nedetekovaný průjezd vozidla, nebo o chybně detekovaný průjezd.

V rámci experimentování se vyskytli též záznamy, kde selhalo určení prvního a druhého úběžníku ve scéně, čímž bylo znemožněno provedení detekce průjezdů vozidel. Jednalo se o případy, kdy byla před křižovatkou zakřivena. Takové záznamy nejsou součástí vyhodnocení viditelného v tabulce 5.2.

		počet vozidel	detekovaná vozidla	chybně detekovaná	nedetekovaná vozidla	podíl chybných detekcí	úspěšnost detekce
automatická kalibrace	ideálně umístěná kamera	1772	1286	5	486	0.0%	72.4%
	vzdálená kamera	109	88	10	31	11.4%	73.9%
ruční kalibrace	ideálně umístěná kamera	1772	1367	5	415	0.0%	76.9%
	vzdálená kamera	109	88	8	29	9.0%	75.2%

Tabulka 5.2: Úspěšnost detekce průjezdu vozidel. Procenta zaokrouhlena na jedno desetinné místo.

V rámci provedených experimentů byla přesnost detekce průjezdu automobilu vždy vyšší než 70%. Problém nezaznamenání průjezdu vozidla tkví zejména v neschopnosti jeho detekce na snímku použitým modelem neuronové sítě. Případy, kdy je vozidlo špatně sledováno a překročí tak zastavovací linii v chybný čas jsou výjimkou. Častěji dochází k detekci chybných průjezdů na záznamu pořízeném pod nižším úhlem k vozovce ze vzdálenější kamery. Dochází zde k překryvu jednotlivých vozidel u zastavovací čáry a při příjezdu následujícího vozidla může být analýzou optického toku chybně detekován pohyb vozidla předešlého. Řešení obou zmíněných problémů je možné zvýšením úspěšnosti detekce použité neuronové sítě, díky níž by bylo možné klást jednotlivým detekcím při měřeních vyšší váhu, než-li pohybu získaného zkoumáním optického toku.

Kvalita detekce jízdy na červenou

Následně bylo provedeno vyhodnocení schopnosti detekovat jízdu na červenou viz tabulka 5.3. Na některých záznamech však takových prohřešků nebylo dostatečné množství pro proká-

zání funkčnosti použitých metod, proto byla v rámci experimentování považována za přestupek rovněž jízda na oranžovou. Hodnocení úspěšnosti detekce přestupků je určeno čistě kvantitativním porovnáním. V rámci experimentování totiž bylo zjištěno, že podíl chybně anotovaných/detekovaných průjezdů je výrazně vyšší než u pouhé detekce průjezdů (většina chybně detekovaných průjezdů jsou zároveň průjezdy na oranžovou nebo červenou). Jedná se zejména o případy kdy automobil překročí čáru v nízké rychlosti a jeho pohyb je tedy zaznamenán na větším rozsahu snímků a ruční anotace je v takovém případě také nepřesná. Tato chyba je přítomna též při hodnocení úspěšnosti detekce samotných průjezdů vozidel, avšak jedná se o minoritu detekovaných průjezdů, tudíž je tam tato chyba zanedbatelná. Pro získání počtu chybně detekovaných průjezdů byla analyzována výstupní videa zachycující jednotlivé přestupky za využití skriptu pro jejich přehrávání `violation_player.py` který je také součástí paměťového média přiloženého k této práci.

		počet přestupků	detekované přestupky	chybně detekované	podíl chybných detekcí	úspěšnost detekce
automatická kalibrace	vysoký provoz	117	68	3	4.4%	56.7%
	střední provoz	21	19	0	0.0%	90.5%
ruční kalibrace	vysoký provoz	117	74	4	5.4%	61.2%
	střední provoz	21	20	0	0.0%	95.2%

Tabulka 5.3: Úspěšnost detekce přestupků. V rámci experimentování je za přestupek považována jak jízda na červenou tak i jízda na oranžovou. Každý řádek představuje data získaná z jednoho zkoumaného záznamu. Procenta zaokrouhlena na jedno desetinné místo.

Úspěšnost detekce přestupků je úzce spjata se schopností detekovat průjezdy jednotlivých vozidel skrze zastavovací linii. V rámci předešlé části zabývající se úspěšností detekce průjezdu vozidel byl popsán problém týkající se vozidel dojíždějících do stojící kolony automobilů, kdy dochází k přiřazení pohybu již stojícímu vozidlu. Tento problém lze mimo již navržené řešení též eliminovat vhodnějším umístěním kamery, tak aby snímala vozovku pod vyšším úhlem a docházelo tak k menšímu překryvu vozidel na snímcích.

K chybné detekci dochází také v případě, kdy vozidlo protíná zastavovací linii v čase blízkém změny stavu semaforu. Důvodem je fakt, že zastavovací linie je aproximována na základě chování vozidel a nekoresponduje tedy přesně s reálnou zastavovací linií. Problém by bylo možné řešit akumulací vyššího počtu vozidel při metodě použité pro její odhadnutí místa pro zastavení, došlo by tak ale též k výraznému zvýšení počtu snímků potřebných pro jeho odhadnutí.

5.4 Možnost dalšího vývoje

V předešlé části byla v rámci rozboru provedených experimentů již prezentována možná zlepšení navrženého systému pro získání vyšší přesnosti.

S využitím navrženého programu, který zapisuje detekované jízdy na červenou ve formě krátkých video sekvencí společně s anotacemi pozic vozidel na daných snímcích, by bylo možné podstoupit detekované průjezdy dalšímu zpracování za účelem potrestání pachatele – například detekcí poznávací značky vozidla. Dále by bylo možné, při využití přesnějšího popisu objektů vozidel ve scéně, zakomponovat do výpočtu měření rychlosti projíždějících vozidel.

V rámci navrženého programu je umožněna detekce jízdnic pruhů na jejímž základě by bylo možné extrahovat statistická data průjezdů vozidel pro jednotlivé pruhy. Na základě tříd detekovaných vozidel by bylo možné určit, zda-li se jedná o osobní vozidla nebo jiné dopravní prostředky, což jsou informace, které lze využít v oblasti dopravního inženýrství.

Kapitola 6

Závěr

V rámci mé práce jsem se zabýval postupem automatické detekce jízdy na červenou ze záznamu pořízeném stacionární kamerou. Čtenáři byly nejprve představeny aktuální přístupy řešených podproblémů, které bylo třeba během návrhu řešit. Následně jsem navrhl konkrétní postup pro detekci jízdy na červenou s automatickou kalibrací kamery za využití dříve představených metod. Na základě provedených experimentů byly poté některé z navržených metod modifikovány pro získání přesnějších výsledků.

Vytvořený program implementuje navržené postupy a umožňuje uživateli jak automatickou kalibraci scény, tak ruční kalibraci semaforů a jízdních koridorů společně s místem pro zastavení. Program automaticky lokalizovat pozici semaforu ve scéně a určit jeho aktuální stav, čehož je bez problémů schopen na snímcích pořízených během dne, kde dosahuje úspěšnosti určení aktuálního stavu přesahující 90%. Nižší úspěšnost detekce semaforu na snímcích pořízených v noci dává prostor dalšímu vývoji v této oblasti. Program dále provádí detekci a sledování automobilů v obraze na jejichž základě dochází ke kalibraci pozorované scény a aproximaci jízdních pruhů společně s místem pro zastavení což umožňuje rozpoznat jízdu na červenou pozorovaných vozidel.

Výstupem programu jsou krátké záznamy detekovaného přestupku pro každé vozidlo, které přestupek spáchalo společně s anotací vozidla na každém pořízeném snímku. Ty by mohly být dále využity pro další zpracování.

Na základě dosažených výsledků při provedených experimentech se hlavním problémem jeví využitý model neuronové sítě pro detekci vozidel, jelikož na mnoha snímcích není schopen jednotlivá vozidla rozpoznat. Implementované postupy jsou schopny vliv této nedokonalosti snížit za využití optického toku pro měření změny polohy vozidla ve scéně. Kvalitněji natrénovaný model by výrazně snížil počet nedetekovaných vozidel, čímž by bylo docíleno vyšší úspěšnosti zachycení jednotlivých průjezdů. Program přesto dosahuje na základě experimentů více než 90% úspěšnosti detekce jízd na červenou při sledování středního provozu. Podíl chybně detekovaných průjezdů vozidel skrze zastavovací linii se pohybuje pod hranicí 6%.

Literatura

- [1] Albawi, S.; Abed Mohammed, T.; Alzawi, S.: Understanding of a Convolutional Neural Network. 2017, doi:10.1109/ICEngTechnol.2017.8308186, [Online; citováno 12.05.2019].
URL https://www.researchgate.net/publication/319253577_Understanding_of_a_Convolutional_Neural_Network
- [2] Bradski, D. G. R.; Kaehler, A.: *Learning OpenCV, 1st Edition*. O'Reilly Media, Inc., první vydání, 2008, ISBN 978-0-596-51613-0.
- [3] Canny, J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník PAMI-8, č. 6, 1986: s. 679–698, ISSN 0162-8828, doi:10.1109/TPAMI.1986.4767851.
- [4] Coughlan, J. M.; Yuille, A. L.: Manhattan World: compass direction from a single image by Bayesian inference. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, ročník 2, 1999, s. 941–947, doi:10.1109/ICCV.1999.790349.
- [5] Coughlan, J. M.; Yuille, A. L.: The Manhattan World Assumption: Regularities in Scene Statistics which Enable Bayesian Inference. In *Advances in Neural Information Processing Systems 13*, editace T. K. Leen; T. G. Dietterich; V. Tresp, MIT Press, 2001, s. 845–851.
URL <http://papers.nips.cc/paper/1804-the-manhattan-world-assumption-regularities-in-scene-statistics-which-enable-bayesian-inference.pdf>
- [6] D. Lucas, B.; Kanade, T.: IJCAI'81 Proceedings of the 7th international joint conference on Artificial intelligence. 04 1981, s. 674–679.
- [7] Dubska, M.; Herout, A.; Havel, J.: PClines — Line detection using parallel coordinates. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011: s. 1489 – 1494, doi:10.1109/CVPR.2011.5995501.
- [8] Dubska, M.; Sochor, J.; Herout, A.: Automatic Camera Calibration for Traffic Understanding. *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 2014, doi:10.5244/C.28.42.
- [9] Dubska, M.; Herout, A.: Real Projective Plane Mapping for Detection of Orthogonal Vanishing Points. In *Proceedings of BMVC 2013*, The British Machine Vision Association and Society for Pattern Recognition, 2013, ISBN 1-901725-49-9, s. 1–10, doi:10.5244/C.27.90.
URL http://www.fit.vutbr.cz/research/view_pub.php.cs?id=10400

- [10] Dubská, M.; Herout, A.; Juránek, R.; aj.: Fully Automatic Roadside Camera Calibration for Traffic Surveillance. *IEEE Transactions on Intelligent Transportation Systems*, ročník 16, č. 3, 2015: s. 1162–1171, ISSN 1524-9050, doi:10.1109/TITS.2014.2352854.
- [11] Duda, R. O.; Hart, P. E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, ročník 15, č. 1, 1972: s. 11–15, ISSN 0001-0782, doi:10.1145/361237.361242.
URL <http://doi.acm.org/10.1145/361237.361242>
- [12] Duda, R. O.; Hart, P. E.: *Pattern Classification and scene analysis*. John Wiley & Sons, Inc., 1973.
- [13] Fischler, M. A.; Bolles, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, ročník 24, č. 6, 1981: s. 381–395.
- [14] Godbehere, A. B.; Matsukawa, A.; Goldberg, K.: Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *2012 American Control Conference (ACC)*, 2012, ISSN 2378-5861, s. 4305–4312, doi:10.1109/ACC.2012.6315174.
- [15] Hsu, S.; Huang, C.; Chuang, C.: Vehicle detection using simplified fast R-CNN. In *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018, s. 1–3, doi:10.1109/IWAIT.2018.8369767.
- [16] Huang, J.; Rathod, V.; Sun, C.; aj.: Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, ISSN 1063-6919, s. 3296–3297, doi:10.1109/CVPR.2017.351.
- [17] Julliard, T.; Nozick, V.; Talbot, H.: Image Noise and Digital Image Forensics. 2016, ISBN 978-3-319-31959-9, s. 3–17, doi:10.1007/978-3-319-31960-5_1.
- [18] Kaewtrakulpong, P.; Bowden, R.: An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. *Proceedings of 2nd European Workshop on Advanced Video-Based Surveillance Systems; September 4, 2001; London, U.K.*, 2002, doi:10.1007/978-1-4615-0913-4_11.
- [19] Kalman, R.: A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of basic Engineering*, ročník 82, 1960: s. 35–45, doi:10.1115/1.3662552.
- [20] Kiryati, N.; Eldar, Y.; Bruckstein, A.: A probabilistic Hough transform. *Pattern Recognition*, ročník 24, č. 4, 1991: s. 303–316, ISSN 0031-3203, doi:https://doi.org/10.1016/0031-3203(91)90073-E.
URL <http://www.sciencedirect.com/science/article/pii/003132039190073E>
- [21] Kiryati, N.; Eldar, Y.; Bruckstein, A. M.: A probabilistic hough transform. *Pattern Recognition*, ročník 24, č. 4, 1991: s. 303–316, ISSN 0031-3203.
- [22] Laganière, R.: *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing Ltd., 2011, ISBN 978-1-849513-24-1.

- [23] Lamberský, V.: *Vývoj algoritmů pro odhad stavu experimentálního vozidla*. Diplomová práce, Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav mechaniky těles, mechatroniky a biomechaniky, 2010.
- [24] Lezama, J.; Randall, G.; Grompone von Gioi, R.: Vanishing Point Detection in Urban Scenes Using Point Alignments. *Image Processing On Line*, ročník 7, 2017: s. 131–164, doi:10.5201/ipol.2017.148.
- [25] Liu, W.; Anguelov, D.; Erhan, D.; aj.: SSD: Single Shot MultiBox Detector. 2016, ISBN 978-3-319-46447-3, s. 21–37, doi:10.1007/978-3-319-46448-0_2.
- [26] Ministerstvo dopravy: Statistiky přestupků a trestných činů za rok 2018. 2019, [Online; navštíveno 10.4.2019].
URL [https://www.mdcz.cz/Statistiky/Silnicni-doprava/Statistiky-k-bodovemu-hodnoceni/Statistiky-prestupku-a-trestnych-cinu/Statistiky-prestupku-a-trestnych-cinu-za-rok-2-\(3\)](https://www.mdcz.cz/Statistiky/Silnicni-doprava/Statistiky-k-bodovemu-hodnoceni/Statistiky-prestupku-a-trestnych-cinu/Statistiky-prestupku-a-trestnych-cinu-za-rok-2-(3))
- [27] O’Shea, K.; Nash, R.: An Introduction to Convolutional Neural Networks. *ArXiv e-prints*, 2015, [Online; citováno 13.05.2019].
URL https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks
- [28] Ren, S.; He, K.; Girshick, R.; aj.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 39, 2015, doi:10.1109/TPAMI.2016.2577031.
- [29] Retting, R. A.; Ferguson, S. A.; Hakkert, A. S.: Effects of Red Light Cameras on Violations and Crashes: A Review of the International Literature. *Traffic Injury Prevention*, ročník 4, č. 1, 2003: s. 17–23, doi:10.1080/15389580309858.
URL <https://doi.org/10.1080/15389580309858>
- [30] Saini, S.; Nikhil, S.; Konda, K. R.; aj.: An efficient vision-based traffic light detection and state recognition for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, s. 606–611, doi:10.1109/IVS.2017.7995785.
- [31] SharkD: The HSV color model mapped to a cylinder. Licencováno pod CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0/>,.
URL https://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder.png
- [32] SharkD: RGB-Cube The RGB color model mapped to a cube. Licencováno pod CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0/>,.
URL https://commons.wikimedia.org/wiki/File:RGB_Cube_Show_lowgamma_cutout_a.png
- [33] Shi, Z.; Zou, Z.; Zhang, C.: Real-Time Traffic Light Detection With Adaptive Background Suppression Filter. *IEEE Transactions on Intelligent Transportation Systems*, ročník 17, č. 3, 2016: s. 690–700, ISSN 1524-9050, doi:10.1109/TITS.2015.2481459.
- [34] Smith, A. R.: Color Gamut Transform Pairs. *SIGGRAPH Comput. Graph.*, ročník 12, č. 3, 1978: s. 12–19, ISSN 0097-8930, doi:10.1145/965139.807361.
URL <http://doi.acm.org/10.1145/965139.807361>

- [35] Sochor, J.: *Traffic Analysis from Video*. Diplomová práce, Brno University of Technology, Faculty of Information Technology, 2014.
- [36] Thacker, N.; Lacey, A.: *Tutorial: The Likelihood Interpretation of the Kalman Filter*. Tina Memo, ročník 1996-002, [Online; citováno 13.05.2019].
URL <http://www.tina-vision.net/docs/memos/1996-002.pdf>
- [37] Tsai, F.; Chang, H.: Detection of vanishing points using Hough transform for Single View 3D Reconstruction. 10 2013, s. 1182–1189.
- [38] Varghese, A.; Sreelekha, G.: Background subtraction for vehicle detection. In *2015 Global Conference on Communication Technologies (GCCT)*, 2015, s. 380–382, doi:10.1109/GCCT.2015.7342688.
- [39] W. Kuhn, H.: The Hungarian Method for The Assignment Problem. *Naval Res. Logist. Quart.*, ročník 2, 1955: s. 83–97, doi:10.1002/nav.3800020109.
- [40] Xiang, X.; Zhai, M.; Lv, N.; aj.: Vehicle Counting Based on Vehicle Detection and Tracking from Aerial Videos. *Sensors*, ročník 18, 2018, doi:10.3390/s18082560.
- [41] Young, I.: Image analysis and mathematical morphology, by J. Serra. *Cytometry*, ročník 4, 1983: s. 184–185, doi:10.1002/cyto.990040213.

Příloha A

Obsah DVD

- Program pro detekci jízdy na červenou
- Program pro zpětné přehrávání detekovaných přestupků
- Zdrojové texty bakalářské práce
- PDF soubor obsahující elektronickou kopii této práce
- Popis instalace a základního ovládání přiložených programů
- Dokumentace programu pro detekci jízdy na červenou
- Ukázka videí využitých při experimentování
- Ukázka videí zachycujících detekované přestupky
- Programy umožňující ruční anotaci a vyhodnocení přesnosti detekce
- Ilustrační video
- Ilustrační plakát