



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

**AUTOMATICKÁ KALIBRÁCIA DOPRAVNEJ KAMERY
PRE MERANIE RÝCHLOSTI VOZIDIEL**

AUTOMATIC TRAFFIC CAMERA CALIBRATION FOR VEHICLE SPEED MEASUREMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARIÁN NOCIAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAKUB ŠPAÑHEL

BRNO 2019

Zadání bakalářské práce



22206

Student: **Nociar Marián**
Program: Informační technologie
Název: **Automatická kalibrace dopravní kamery pro měření rychlosti vozidel**
Automatic Traffic Camera Calibration for Vehicle Speed Measurement
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy zpracování obrazu. Zaměřte se zejména na problematiku analýzy dopravy ze stacionární kamery.
2. Dohleďte dostupné materiály na téma automatická kalibrace stacionární kamery a měření rychlosti vozidel.
3. Vyberte vhodnou metodu a navrhnete princip řešení problému automatické kalibrace kamery s využitím dopravních dat.
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

Táto bakalárska práca sa zaoberá návrhom a tvorbou systému pre automatickú kalibráciu dopravnej kamery za účelom merania rýchlosti vozidiel. Práca obsahuje teoretický úvod do problematiky kalibrácie kamery, popisuje princípy metód použitých v implementácii a zobrazuje navrhované riešenie, ktoré bolo implementované v jazyku Python3 s využitím knižníc OpenCV a NumPy. Dosiahnuté výsledky boli vyhodnotené nad datasetom Brno-CompSpeed, pričom priemerná chyba merania rýchlosti dosahuje hodnoty 6,71 km/h.

Abstract

This thesis deals with design and development of a system for Automatic Traffic Camera Calibration for Vehicle Speed Measurement. The thesis contains a theoretical introduction to the camera calibration, describes the principles of the methods used in the implementation and displays the proposed solution that was implemented in Python3 using OpenCV and NumPy libraries. The achieved results were evaluated over the BrnoCompSpeed dataset with the average speed measurement error reaching 6.71 km/h.

Kľúčové slová

Kalibrácia kamery, detekcia úbežníkov, meranie rýchlosti, automatická kalibrácia, doprava, Python3, OpenCV

Keywords

Camera calibration, vanishing points detectin, speed measurement, automatic calibration, traffic, Python3, OpenCV

Citácia

NOCIAR, Marián. *Automatická kalibrácia dopravnej kamery pre meranie rýchlosti vozidiel*. Brno, 2019. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jakub Špaňhel

Automatická kalibrácia dopravnej kamery pre meranie rýchlosti vozidiel

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jakuba Špaňhela. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Marián Nociar
15. mája 2019

Podakovanie

Touto cestou by som chcel poďakovať vedúcemu mojej bakalárskej práce Ing. Jakubovi Špaňhelovi, za jeho čas, cenné rady a informácie, a odbornú starostlivosť poskytnutú pri spracovávaní tejto bakalárskej práce.

Obsah

1	Úvod	3
2	Kalibrácia kamery	4
2.1	Existujúce systémy	4
2.2	Kalibrácia na princípe homografie	8
2.3	Kalibrácia pomocou úbežníkov	9
3	Použité metódy	11
3.1	Detekcia pohybujúcich sa objektov	11
3.2	Detekcia bodov vhodných k sledovaniu	13
3.3	Optický tok	16
4	Návrh systému	18
4.1	Kalibrácia	18
4.2	Detekcia a sledovanie vozidiel	25
4.3	Výpočet vzdialenosti	26
4.4	Výpočet mierky scény	26
4.5	Výpočet rýchlosti	27
5	Implementácia a testovanie	29
5.1	Dataset	30
5.2	Vyhodnotenie detekcie úbežníkov	30
5.3	Vyhodnotenie merania vzdialenosti	32
5.4	Vyhodnotenie merania rýchlosti	33
6	Záver	35
	Literatúra	36

Zoznam obrázkov

2.1	Projekcia vzdialenosti medzi bodmi (p_1, p_2) v obraze do skutočnej vzdialenosti d v rovine vozovky (Obr. prevzatý z [13])	5
2.2	Mapovanie obrazových súradníc (x, y) na (X, Y) pomocou homografie H	8
3.1	Typy oblastí: čierne okno obsahuje plochú oblasť, modré obsahuje hranu a červené roh	14
4.1	Schéma blokov so vzájomnou komunikáciou pre navrhovaný systém	19
4.2	Dve postupné Houghove transformácie bodov a priamok do paralelného súradného systému \mathcal{S} (Obr. prevzatý z [16])	20
4.3	Nájdenie a sledovanie vhodných bodov pre detekciu prvého úbežníka.	21
4.4	Vzisknuté diamond space priestory pre detekciu prvého úbežníka	22
4.5	Detekcia hrán na vozidlách a výpočet ich smeru v lokálnych maximách, pre určenie druhého úbežníka	23
4.6	Zobrazenie DS priestoru pre detekciu 2 úbežníka s aplikovaním masky	24
4.7	Proces vyhladenia masky popredia pomocou postupných morfológických operácií	25
4.8	Grafický výstup zobrazujúci ID a rýchlosť vozidiel.	28
5.1	Vyznačené a namerané vzdialenosti na rovine vozovky. Červené čiary smerujú k prvému úbežníku a zelené k druhému. (Obr. prevzatý z [13]).	30
5.2	Kumulatívny histogram absolútnych chýb pre implementovaný systém.	34

Kapitola 1

Úvod

V súčasnej dobe je čoraz viac kamier, ktoré monitorujú premávku na cestných komunikáciách. S rastúcim počtom vozidiel predstavujú efektívne riešenie pre analýzu dopravy. Len s pomocou záznamu je možné zistiť kopec informácií ako počet jazdných pruhov, rýchlosti prechádzajúcich vozidiel, ŠPZ a detegovať napríklad jazdu v protismere alebo nehodovú udalosť.

Hlavným cieľom tejto práce je implementovať systém, ktorý automaticky spracuje videá z dohľadovej kamery, za účelom získania rýchlostí prechádzajúcich vozidiel. K tomu je potrebné získať kalibračné parametre kamery. Tieto parametre sú identifikované plne automaticky, bez nutného vstupu od užívateľa pred spustením systému. Mnoho existujúcich riešení vyžaduje dodatočné informácie, ktoré sú získané priamo v teréne, čo spomaľuje nasadenie celého systému. Automatická kalibrácia je založená na detekcii vozidiel pomocou ktorých sa určia úbežníky. Systém preto nevyžaduje prítomnosť vodiacich čiar alebo veľký počet jazdných pruhov. Jediným hlavným predpokladom je, aby pozorovaný úsek cesty bol rovný. Pri kamerách, ktoré sledujú ostré zákruty alebo kruhový objazd, detekcia úbežníkov nie je možná.

Po získaní kalibračných údajov je možné merať vzdialenosť v rovine vozovky, čo predstavuje prvý krok k meraniu rýchlosti. Aby sme zistili prejdenú vzdialenosť vozidiel, najprv ich je nutné v scéne identifikovať a ďalej sledovať. Zostáva už len čas, za ktorý vozidlo prešlo danú vzdialenosť. Ten sa určí z rýchlosti snímania kamery. Výsledná rýchlosť je ale v relatívnych jednotkách, preto je ju nutné prepočítať pomocou získanej mierky scény na štandardné jednotky.

Táto práca sa najskôr zaoberá princípom kalibrácie dopravnej kamery a existujúcimi riešeniami, ktoré sú popísané v kapitole 2. V kapitole 3 sú obsiahnuté informácie o metódach a algoritmoch použitých pri implementácii. Samotný návrh systému je popísaný v kapitole 4 a výsledky testov implementovaného riešenia sú zhrnuté v kapitole 5.

Kapitola 2

Kalibrácia kamery

Jedna z najdôležitejších častí pre meranie rýchlosti vozidla je kalibrácia dopravnej kamery. Pre určenie rýchlosti musíme zistiť reálnu vzdialenosť, ktorú vozidlo prešlo v obraze za určitý čas. Tento rozmer v prejdenej trojrozmernej vzdialenosti je nutné extrahovať z rozmeru z dvojrozmerného snímku. Na obrázku 2.1 je znázornená projekcia obrazových bodov do roviny. Preto musí byť kalibrácia čo najpresnejšia, lebo presnosť kalibrácie kamery priamo ovplyvňuje aj výsledky vzdialenosti medzi ľubovoľnými bodmi v rovine pozorovaného obrazu.

Pre kalibrovanie kamery je zvyčajne nutné získať vnútorné a vonkajšie parametre kamery spoločne s mierkou scény alebo poznať vzdialenosť medzi rovinou vozovky a kamerou. Vo všeobecnom prípade to zahŕňa vysporiadanie sa s perspektívnym zobrazením, rôznym natočením kamery, neznámou vzdialenosťou kamery od roviny vozovky a možným radiálnym a tangenciálnym skreslením[13].

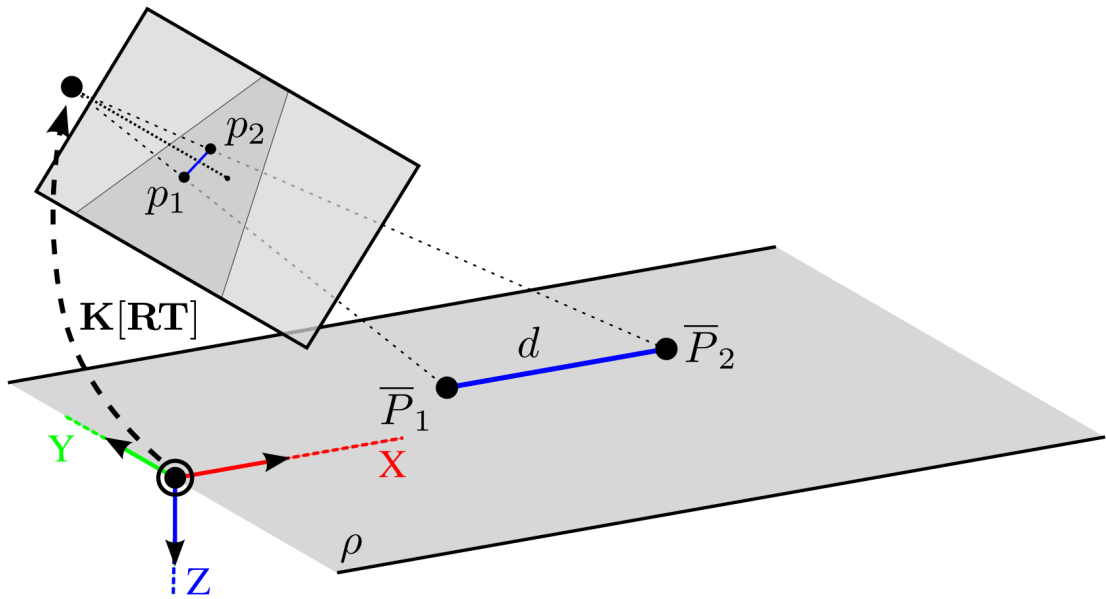
Pokiaľ tieto informácie o umiestnení kamery v priestore spolu s jej natočením a merítkom scény nie sú známe, je nutné tieto informácie získať z pozorovaného videa. V súčasnej dobe sa najčastejšie používajú dva druhy kalibračných informácií – matice KRT a úbežníky.

Je veľkou výhodou, pokiaľ sú tieto informácie získané plne automaticky. Väčšina systémov pracuje na poloautomatickom režime, kde niektoré informácie ako výška kamery nad vozovkou alebo reálna vzdialenosť dvoch bodov v obraze, sú zadané užívateľom pred spustením kalibračného procesu. Preto je dôležité získať tieto informácie čisto zo záznamu, aby sa urýchlil proces nasadenia systému pre ľubovoľnú dopravnú kameru.

2.1 Existujúce systémy

V tejto kapitole sú prezentované štyri rozdielne systémy pre analýzu dopravy s dôrazom na meranie rýchlosti vozidiel. Pre každý z týchto systémov platí, že musia získať informácie potrebné pre skalibrovanie kamery (vnútorné a vonkajšie parametre kamery). Od týchto parametrov závisí presnosť celého systému.

Práce používajú dva základné prístupy. Prvým je použitie skalibrovanej kamery. Pri skalibrovanej kamere je vopred známa presná poloha kamery, výška nad vozovkou, posun kamery od stredu vozovky, natočenie kamery vo všetkých troch osiach [3, 8]. Druhým prístupom je použitie nezkalibrovanej kamery, kde vyššie popísané parametre nie sú známe a je nutné ich získať priamo zo záznamu buď plne automaticky, alebo pomocou užívateľa [10, 5].



Obr. 2.1: Projekcia vzdialenosti medzi bodmi (p_1, p_2) v obraze do skutočnej vzdialenosti d v rovine vozovky (Obr. prevzatý z [13])

Ďalej sa systémy líšia v spôsobe detekovania a sledovania vozidiel. Populárnym prístupom je použitie detekcie popredia s Kalmanovým filtrom k sledovaniu vozidiel [10, 8]. Iné práce sú založené na sledovaní vhodných bodov a ich spájanie do skupín [3].

Detailnejší popis jednotlivých systémov je predstavený nasledovne.

D. Beymer et al., 1997

Systém [3] navrhnutý pánom Beymerom sleduje jednotlivé vozidlá a aktualizuje ich pozíciu v 3D svetových súradniciach, až kým vozidlá neopustia sledovaciu zónu. Z týchto pozícií následne počíta dopravné parametre zahrňujúce počty vozidiel v jazdných pruhoch, priemernú rýchlosť, zmeny v jazdných pruhoch a ďalšie. Tento systém je založený na detekcii významných bodov (corner features). Detekované body sa následne sledujú a zoskupujú do skupín, ktoré predstavujú individuálne vozidlá.

Pre definovanie roviny vozovky je nutné, aby užívateľ zadal štyri alebo viac bodov, zhodných medzi obrazovou rovinou a "svetovou" rovinou vozovky. Systém potom môže spočítať perspektívnu projekciu alebo homografiu medzi obrazovými súradnicami (x, y) a svetovými súradnicami (X, Y) . Homografia H je vyjadrená ako jednoduchá lineárna transformácia.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.1)$$

Pred spustením systému musí užívateľ okrem zhodných bodov homografie definovať aj oblasť záujmu pre vstup a výstup vozidiel v obraze.

Detekcia významných bodov je založená na gradiente pixelov ∇I . Spočíta sa matica $\nabla I \nabla I^T$. Body, ktoré budú nad preddefinovaným prahom sa vyberú s ich okolím 9×9 pre porovnanie v sledovacom module.

Sledovací modul využíva Kalmanov filter [18] pre sledovanie bodov. V každom snímku Kalmanov filter predpovedá, kde sa môže v ďalšom snímku vyskytovať sledovaný bod. Bod je úspešne sledovaný, ak pozícia bodu z ďalšieho snímku sa vzťahuje k odhadnutému okoliu. Spájanie sledovaných bodov do jedného objektu je založené na spoločnom pohybe týchto bodov. Spájací proces musí byť citlivý, aby odlíšil body patriace susednému vozidlu. Spojovací modul používa neorientovaný graf v čase. Vrcholy predstavujú sledované body, zatiaľ čo hrany určujú vzťahy medzi bodmi. V čase sa vypočítava relatívne posunutie bodov. Pokiaľ je posunutie väčšie ako prahová hodnota, bod je vylúčený zo zhluku.

T.N. Schoepflin et al., 2003

V práci [10] publikovanej Schoepflin and Dailey popisujú kalibráciu kamery pre meranie priemernej rýchlosti pomocou vytvorenia mapy aktivít pre cestu s viacerými pruhmi. Algoritmus pozostáva z troch základných častí: odhad polohy kamery, extrahovanie vlastností obrazu pre kalibráciu kamery a sledovanie vozidiel s následným meraním rýchlosti.

V prvej fáze odhadujú polohu kamery voči vozovke pomocou geometrickej techniky, ktorá je závislá na identifikácii rovnobežných a kolmých čiar v obraze. K tomu používajú mapu aktivít vytvorenú z dopravného toku s následnou Houghovu transformáciou na identifikáciu hraníc rovnobežnej vozovky, a hrany vozidiel pre vytvorenie čiar kolmých na hrany vozovky. Mapa aktivít sa mení v čase, keď vozidlá prechádzajú obrazom. Predpokladá sa, že jediný významný pohyb v obraze je spôsobený práve vozidlami. Mapa A je teda zostrojená z rozdielných intenzít medzi dvoma snímkami I_i a I_{i+1} .

$$A = \frac{1}{N} \sum_{i=1}^N |I_i - I_{i+1}| \quad (2.2)$$

Z tejto mapy aktivít môžu získať odhad úbežného bodu (vanishing point). Úbežníkové body spolu s hranicami jazdných pruhov umožňujú odhadnúť polohu kamery.

Druhá časť algoritmu odhaduje orientáciu a ohniskovú vzdialenosť kamery za predpokladu, že je známa poloha kamery vzhľadom k vozovke. Podobne ako pri výpočte polohy kamery, kalibráciu vykonávajú pomocou odhadnutého úbežného bodu, spolu so získanými hranicami jazdných pruhov.

Potom, ako bola kamera skalibrovaná, prechádzajú do tretej časti, kde merajú rýchlosť vozidiel. Po extrakcii masky popredia a jednotlivých dráh detekovaných v obraze získajú polohu vozidla, ktorá je daná súradnicami stredu jazdného pruhu a spodnou hranou vozidla.

Autori vykonali experimenty na ich vlastnom datasete, kde namerané výsledky porovnávali s dátami z indukčných senzorov uložených vo vozovke. Priemerná rýchlosť bola menšia alebo rovná v porovnaní s reálnou rýchlosťou, čo autori pripisujú nenulovému sklonu vozovky a malému rozlíšeniu videa potrebného k odhadu úbežníka a jazdných pruhov.

D. C. Luvizon et al., 2017

Ďalší predstavený systém [8] bol navrhnutý pánmi Luvizon, Nassu a Minetto. Počíta s mnohými predpokladmi, ako priamočiary pohyb vozidiel s konštantnou rýchlosťou, poznávacie značky umiestnené približne v rovnakej výške atď. Tieto predpoklady im umožňujú merať rýchlosť vozidla bez modelovania 3D priestoru a nevyžadujú presnú kalibráciu, alebo polohu kamery.

Prvý krok ich systému predstavuje detekciu pohybujúcich sa vozidiel, ktoré budú vymedzovať oblasť záujmu v ďalšom spracovaní. K tomu používajú segmentáciu popredia od

pozadia. Ideálne by každá oblasť mala obsahovať jedno vozidlo s jasne viditeľnou licenčnou značkou.

Tieto zvolené oblasti sa posielajú detektoru registračných značiek, ktorý detekuje obdĺžnik približne ohraničujúci tabuľku ŠPZ. Opierajú sa o poznatky Zheng et al. že oblasti okolo ŠPZ majú dlhé vodorovné hrany a malé množstvo náhodného šumu. Preto detekujú z obrazu vertikálny gradient G_x konvolúciou vstupnej oblasti s horizontálnym Sobelovým operátorom. Nájdené hrany následne filtrujú podľa maximálnej a minimálnej dĺžky v pixeloch, aby približne odpovedali rozmerom ŠPZ. Regióny, ktoré spĺňajú rozmery, sú nakoniec posielané klasifikátoru, ktorý odstráni všetky regióny, v ktorých sa nevyskytujú textové informácie.

Po detekovaní oblasti s ŠPZ, sa v tejto oblasti vyhľadajú vhodné body a tie sú sledované. Vyhľadanie bodov vhodných k sledovaniu sa vykoná iba raz pre každé vozidlo, hneď ako bola detekovaná ŠPZ. Cieľom je vytvoriť zoznam, ktorý obsahuje pozície sledovaných bodov. Kalibrácia je založená na definovaní štyroch známych bodov, čím sa získa homografná matica H . Z vytvoreného zoznamu pozícií sa vyjadří pohybový vektor \vec{d}_i , ktorý je pomocou matice H prevedený na vektor \vec{v}_i reprezentujúci vzdialenosť v metroch.

Autori vyhodnotili namerané výsledky nad vlastným datasetom obsahujúcim 20 videí. Z ich záverov vyplýva, že 96% meraní je v rozsahu akceptovateľného limitu $[-3\frac{km}{h}, +2\frac{km}{h}]$.

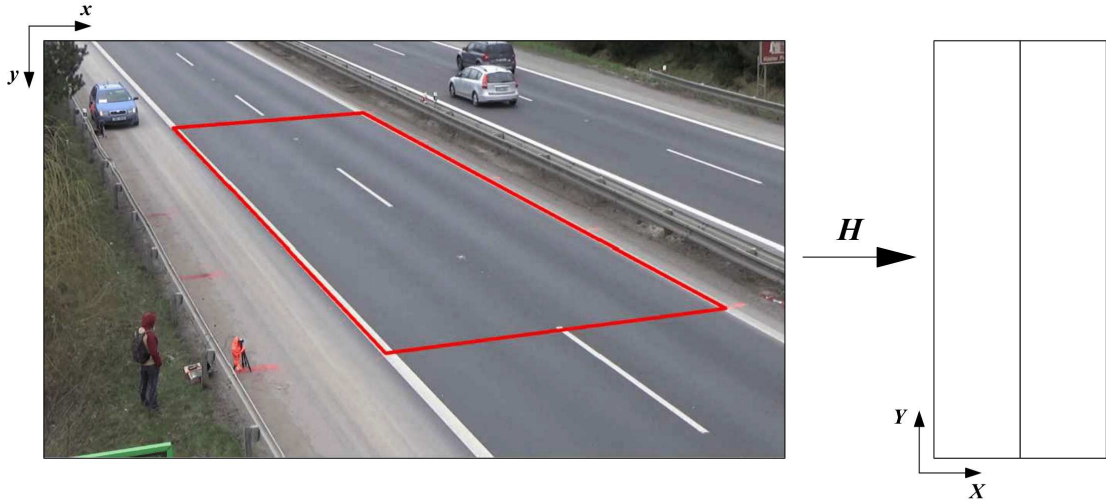
M. Dubska et al., 2015

Posledný systém, ktorý vybočuje od predchádzajúcich, je navrhnutý pani Dubskou a ďalšími [5]. Jedná sa o plne automatickú kalibráciu dopravnej kamery vrátane zistenia merítka scény, bez nutnosti zadávania akýchkoľvek informácií. Systém je rozdelený do dvoch základných častí.

Prvou časťou je detekovanie všetkých troch úbežníkov. Nástroj *Diamond Space* pre detekovanie úbežníkov používa Houghovu transformáciu založenú na paralelných koordinátoch. Prvý úbežník je v smere pohybujúcich sa vozidiel. Na vozidlách sa detekujú vhodné body, ktoré sa sledujú pomocou KLT sledovača. Počiatočná a koncová pozícia bodu určuje priamku, ktorá smeruje do 1. úbežníka. Všetky tieto priamky hlasujú v *DS*. Bod, ktorý v *DS* bol najviac hlasovaný predstavuje pozíciu prvého úbežníka. Druhý úbežník je paralelný k prvému. Opäť je použitý nástroj *DS* pre jeho detekciu. Mnoho hrán na vozidlách smeruje k tomuto úbežníku. Tieto hrany sa vyhľadajú v oblastiach popredia pomocou hranovej funkcie. Vyberú sa body s magnitudou väčšou ako zadaná prahová hodnota. Pre vyhovujúce body sa z gradientu vypočíta vektor, ktorý smeruje k druhému úbežníku. Po detekovaní prvých dvoch úbežníkov sa dopočíta ohnisková vzdialenosť a poloha tretieho úbežníka, čím je kalibrácia kamery ukončená.

Ďalším krokom je zostrojenie 3D boxu okolo pozorovaných vozidiel pomocou tangenciálnych čiar prechádzajúcimi bodmi miznutia. Vzhľadom k tomu, že strecha a boky vozidla sú zahnuté, zostrojený 3D box je menší ako v realite. Merítka scény je odhadnuté na základe známej strednej veľkosti vozidiel k zostrojenému 3D boxu.

Autori porovnali namerané vzdialenosti pomocou ich metódy s reálnymi vzdialenosťami a priemerná chyba bola 1.9%, a v najhoršom prípade 5.6%. Pri meraní rýchlosti odhadli rýchlosť aut idúcich priemernou rýchlosťou 75 km/h s presnosťou 98% čo predstavuje chybovú odchylku 1,5 km/h.



Obr. 2.2: Mapovanie obrazových súradníc (x, y) na (X, Y) pomocou homografie H .

2.2 Kalibrácia na princípe homografie

Tento prístup je založený na zadaní niekoľkých údajov. Môže sa jednať rovno o kalibračnú maticu KRT , alebo o skupinu bodov so známou polohou v rovine snímky, korešpondujúcou s príbuznou skupinou bodov so známou pozíciou v trojrozmernom priestore (viď. obrázok 2.2).

Vzťah medzi súradnicou bodu M v trojrozmernom priestore a jeho priemetom v rovine snímky m je $m = K[R|t]M$

$$K = \begin{bmatrix} f_x & s & j_0 \\ 0 & f_y & i_0 \\ 0 & 0 & 1 \end{bmatrix} [R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}. \quad (2.3)$$

Matica K obsahuje informácie o vnútorných parametroch kamery kde f_x a f_y sú ohniskové vzdialenosti, i_0 a j_0 predstavujú optický stred snímky a s skosenie obrazu, pričom obvykle $s = 0$. Matica $[R|t]$ nesie informácie o vonkajších parametroch kamery reprezentujúca rotáciu a posunutie kamerového systému. Závislosť medzi bodom v rovine snímky a bodom v trojrozmernom priestore je daná maticou homografie H . Pri predpoklade, že rovina vozovky je umiestnená v začiatku súradnicového systému trojrozmerného priestoru, teda súradnica $z = 0$. Matica H bude mať tvar:

$$H = \tau K \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}, \quad (2.4)$$

kde τ je ľubovoľný skalár.[7].

D. Beymer et al. [3] a D. C. Luvizon et al. [8] kalibrujú kameru pomocou štyroch známych bodov, čím získajú homografnú maticu H . Pomocou tejto matice transformujú získané súradnice do reálnych vzdialeností v metroch.

2.3 Kalibrácia pomocou úbežníkov

Pri dopravnej tematike sa javí tou správnou voľbou využitie úbežníkov. Týmto prístupom je možné získať dostatok informácií priamo z pozorovaného videa pre kalibráciu kamery. Úbežníky predstavujú vizuálny efekt, pri ktorom sa rovnobežné línie zbiehajú do jedného bodu zvanom vanishing point. Takýto úbežník v pozorovanej scéne nie je len jeden, ale rovno tri. Prvý úbežník je daný smerom pohybujúcich sa vozidiel. Druhý má kolmý smer k pohybujúcim sa vozidlám a tretí úbežník je kolmý na rovinu vozovky. Existujúce systémy pracujú s rôznym počtom týchto úbežníkov a ide ich rozdeliť na detekciu pomocou:

- **jedného úbežníka** – každý systém využíva detekciu v smere pohybujúcich sa vozidiel. Tieto systémy sa ale líšia v spôsobe ich získavania. Prvá skupina používa na jeho detekciu anotácie na vozovke, teda jazdné pruhy a vodiace čiary. Nevýhodou týchto systémov je ich nasadenie iba v prostredí, kde je pozorovaný väčší počet jazdných pruhov.

Druhá skupina používa na detekciu tohto úbežníka smer pohybujúcich sa vozidiel. Predpokladom pri tejto variante je, aby vozidlá mali paralelnú a vzájomne rovnobežnú trajektóriu v snímanom úseku. Výhoda tohoto prístupu je značná. Je ho možné nasadiť v prostredí, kde nie je viditeľné vodorovné značenie alebo v prostredí s menším počtom jazdných pruhov. Nevýhodou pri menšom počte jazdných pruhov je ale klesajúca presnosť detekcie príslušného úbežníka.

Tieto systémy sú navrhnuté tak aby nepotrebovali detekovať ďalšie úbežníky pre kalibráciu kamery. Namiesto toho pracujú s predpokladom, že kamera je nahnutá len v jednom smere a tým pádom sa druhý úbežník nachádza v nekonečne.

- **dvoch úbežníkov** – okrem úbežníka príslušného smeru pohybujúcich sa vozidiel sa detekuje aj ďalší najčastejšie kolmý na smer dopravy v rovine vozovky. Tento prístup na detekciu druhého úbežníka vyhľadáva hrany na vozidlách, ktoré sú kolme k smere ich pohybu a nesmerujú k prvému úbežníku.

Iným prístupom je detekcia úbežníka kolmého na rovinu vozovky. To sa dosiahne identifikáciou chodcov alebo stĺpov v sledovanom prostredí.

- **troch úbežníkov** – v tomto prípade sa na kalibráciu kamery využívajú všetky tri úbežníky. Tento prístup používa vo svojich prácach Sochor a kol.[12], Dubska a kol.[4], pričom tretí úbežník dopočítajú z pozícií predchádzajúcich úbežníkov za predpokladu, že sa hlavný bod nachádza v strede obrazu.

Systémy k získaniu kalibračných informácií prístupujú pomocou poloautomaticko a automatického režimu. Bližšie detaily k jednotlivým prístupom, ich výhody a nevýhody sú popísané v nasledujúcich podkapitolách.

2.3.1 Poloautomatická kalibrácia

Najčastejšie sa vyskytuje kalibrácia dopravnej kamery v poloautomatickej forme. Väčšina informácií sa extrahuje priamo zo záznamu, ale stále je nutné poznať pár údajov, ktoré musí zadať užívateľ pred spustením kalibračného procesu. V drvivej väčšine sa jedná o zistenie merítka scény pre prevod abstraktnej vzdialenosti v pixeloch, na odpovedajúcu reálnu vzdialenosť napríklad v metroch.

Poloautomatickú kalibráciu využíva Wang a spo. v práci [1]. Pred spustením systému požadujú vyznačenie dvojice paralelných čiar jazdného pruhu s ich známou šírkou. Okrem

týchto informácií je taktiež nutné vedieť vzdialenosť medzi kamerou a rovinou vozovky, prípadne dĺžku vyznačeného úseku rovnobežného s vozovkou.

Iný prístup zvolili Tuan Hue Thi a kol. kde v svojej práci [17] počítajú s informáciou o priemernej dĺžke a šírke vozidiel. Z obrazových dát zbierajú štatistické informácie o rozmeroch vozidiel, ich dĺžku a šírku pre určenie vertikálneho a horizontálneho merítka scény. Všetky dĺžky a šírky sú potom vložené do dvoch vektorov. Stredné hodnoty v týchto dvoch zoradených vektoroch sú potom použité na výpočet konečného merítka scény.

2.3.2 Automatická kalibrácia

Tieto systémy nevyžadujú žiadne dodatočné informácie od užívateľa, ktoré by napomáhali kalibračnému procesu. Pri väčšine poloautomatických riešení je nutné získať dodatočné informácie priamo v teréne, v ktorom sa vyznačia potrebné údaje a namerajú vzdialenosti, čo spôsobí dočasné obmedzenie premávky a nutnosť zaškolenej osoby, ktorá tieto merania vykoná. To omedzuje globálne nasadenia týchto systémov do reálneho sveta, keďže každá kamera musí byť samostatne kalibrovaná. Preto je dôležité získať kalibračné informácie čisto zo záznamu, čo urýchli proces nasadenia systému pre ľubovoľnú dopravnú kameru.

Systémy, ktoré pracujú na plne automatickom režime sú publikované v prácach Dubskej a kol. [4] a Sochora a kol. [12]. Vnútorne a vonkajšie parametre kamery sú získané pomocou detekcie všetkých troch úbežníkov. Líšia sa len v spôsobe určenia merítka scény. V prípade pani Dubskej sa zostrojuje 3D box okolo pozorovaných vozidiel a používajú sa ich stredné hodnoty k určenie merítka. Sochor naopak detekuje v obraze konkrétny typ modelu vozidla, s ktorým prirovnáva jeho odpovedajúci model.

Kapitola 3

Použité metódy

Táto kapitola predstavuje základné činnosti algoritmov, ktoré sú využité pri implementácii navrhovaného systému pre meranie rýchlosti. Pozostáva z troch častí zameraných na určitú oblasť problematiky. Jednou z hlavných úloh je vyhľadanie dynamických objektov v pozorovanej scéne. Rôzne prístupy a ich fungovanie je popísane v podkapitole 3.1. Aby bolo možné sledovať pohyb objektov, je nutné vyhľadať vhodné body, ktoré sú ľahko rozpoznateľné v obraze. Ich detekcii sa venuje podkapitola 3.2. V poslednej podkapitole 3.3 je popísaný princíp optického toku pre sledovanie týchto objektov.

3.1 Detekcia pohybujúcich sa objektov

Segmentácia popredia v pozorovanom obraze je jednou zo základných úloh, ktoré je nutné riešiť pri spracovaní obrazu. Jedná sa o separáciu objektov záujmu od pozadia v obraze. V tejto časti si popíšeme najčastejšie používané techniky pre detekciu pohybujúcich sa vozidiel.

3.1.1 Odčítanie pozadia

Odčítanie pozadia predstavuje metódu, kde sa vyhľadávajú pohybujúce sa objekty pomocou odlišného popredia snímku a pozadia, ktoré je v snímku nemenné. Týmto prístupom sa detekujú pohybujúce sa objekty pomocou rozdielu medzi aktuálnym snímkom a referenčným snímkom, často nazývaným aj modelom pozadia (angl. background model). Model pozadia môže byť buď statický alebo adaptívny. Statický model pozadia je veľmi náchylný na rušivé vplyvy a väčšinou sa používa vo vnútorných priestoroch, kde je zabezpečený konštatný jas prostredia a kde nedochádza k náhodným javom ako pohyb vetiev stromu. Takýto náhodný pohyb alebo rozdiel v jase prostredia by bol pri statickom pozadí vždy vyhodnotený ako pohybujúci sa objekt. Dynamický model nepracuje so stálym modelom pozadia, ale tento model priebežne aktualizuje, čím je zabezpečená reakcia na náhodné javy ako zmena jasu, vietor alebo pridanie statického objektu do pozadia scény.

3.1.2 Rozdiel dvoch snímok

Jedná sa o veľmi jednoduchú techniku, kde sa overuje rozdiel medzi dvomi po sebe idúcimi snímkami. K vytvoreniu pozadia B sa vždy použije predchádzajúci snímok, ktorý sa odčíta od aktuálneho snímku I :

$$|I(x, y, t) - B(x, y, t - 1)| > T, \quad (3.1)$$

kde $I(x, y, t)$ značí hodnotu intenzity pixelu na pozícií (x, y) v čase t od ktorej je odčítaná intenzita toho istého pixelu z predošlej snímky $t - 1$. Výsledkom tejto operácie je obraz zaznamenávajúci zmenu intenzít jasu. Hodnota prahu T predstavuje jediný faktor, ktorý môže ovplyvniť výsledok, a preto musí byť rozumne zvolená, aby sa ignorovali nepatrné zmeny v obraze, akou je napríklad zmena jasu. Preto za popredie sú považované len pixely, ktorých rozdielna intenzita prekračuje danú prahovú hodnotu. Reálne využitie tejto metódy sa neodporuča vzhľadom k jej citlivosti na šum a zmeny svetla. Ďalšou jej nevýhodou je, keď pohybujúci sa objekt má jednotnú farbu, lebo pri odčítaní rovnakej hodnoty je tento pixel pokladaný za pozadie, čo v dôsledku spôsobuje detekciu iba hrán pohybujúcich sa objektov.

3.1.3 Priemerovaný model pozadia (mean filter)

Táto metóda je založená na ukladaní niekoľkých posledných snímok, z ktorých sa spriemereje výsledný snímok predstavujúci pozadie obrazu. Výsledný model pozadia sa pre okamžitý čas vypočíta pomocou vzorca

$$B(x, y, t) = \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t - i), \quad (3.2)$$

kde N je počet predchádzajúcich snímok použitých k výpočtu pozadia. Po vypočítaní pozadia sa od neho odčíta aktuálny snímok. Popredie je teda definované ako

$$|I(x, y, t) - B(x, y, t)| > T, \quad (3.3)$$

kde T predstavuje prahovú hodnotu. Na podobnom princípe pracuje aj mediánový filter, ktorý vypočíta model pozadia z posledných N snímok pomocou mediánu. Výsledná detekcia je závislá od rýchlosti objektov a počtu snímok za sekundu. Nevýhodou týchto metód je pomerne veľká pamäťová náročnosť kvôli uchovávaní histórie snímok. Pokiaľ pohybujúce sa objekty budú mať podobnú farbu ako pozadie, tieto objekty splynú s pozadím. Ďalšou a závažnejšou nevýhodou je použitie globálneho prahu pre všetky pixely v obraze pričom prah nijak nereflektuje jasové zmeny v čase.

3.1.4 Zmes Gaussových kriviek

Stauffer a Grimson v [14] predstavili algoritmus pomocou ktorého sa určuje, či jednotlivé pixely v pozorovanom obraze patria k poprediu alebo k pozadiu pomocou zmesi K Gaussových kriviek. Každá z týchto K distribúcií je parametrizovaná strednou hodnotou μ , rozptylom σ a dodatočným parametrom váhy gaussiánu ω . Každý pixel je modelovaný samostatne pomocou tejto zmesi kriviek. V ktoromkoľvek čase sa pre konkrétny pixel $\{x_0, y_0\}$ uchováva jeho história

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}. \quad (3.4)$$

Tento model je nutné v čase aktualizovať, aby reagoval na vzniknuté zmeny v obraze, akými sú zmeny osvetlenia, pridanie statického objektu do scény, dažď, pohyb vetví, čo by viedlo k zvyšujúcej sa chybe pri detekcií popredia. Pravdepodobnosť, s ktorou pixel nadobúda aktuálnu hodnotu je daná vzťahom:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \quad (3.5)$$

kde K je počet distribúcií pričom sa najčastejšie používa 3 až 5, $\omega_{i,t}$ je odhadovaná váha i -tej Gaussovej zmesi v čase t , pričom súčet všetkých váh musí byť rovný jednej, $\mu_{i,t}$ je stredná hodnota i -tej Gaussovej zmesi v čase t , a kde η je Gaussova funkcia hustoty pravdepodobnosti

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)}. \quad (3.6)$$

Každá nová hodnota pixelu X_t je porovnaná s existujúcimi K Gaussianovými distribúcií, pokiaľ nie je nájdená zhoda. Zhoda je definovaná ako hodnota pixelu so strednou odchýlkou krivky menšou ako 2, 5. Pokiaľ žiadna z K distribúcií nezodpovedá aktuálnej hodnote pixelu, najmenej pravdepodobná distribúcia (Gausián s najmenšou váhou ω) je nahradená distribúciou s aktuálnou hodnotou. Po priradení je nutné prepočítať váhy ostatných distribúcií, do ktorých daný pixel nepatrí pomocou vzťahu

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}), \quad (3.7)$$

kde α je konštanta učenia. Tá definuje časovú konštantu, ktorá určuje rýchlosť akou sa menia parametre distribúcie.

Hodnoty parametrov μ a σ pre distribúciu do ktorej bol pixel priradený sa aktualizujú nasledovne

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (3.8)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t), \quad (3.9)$$

kde

$$\rho = \alpha \eta(X_t | \mu_k, \sigma_k). \quad (3.10)$$

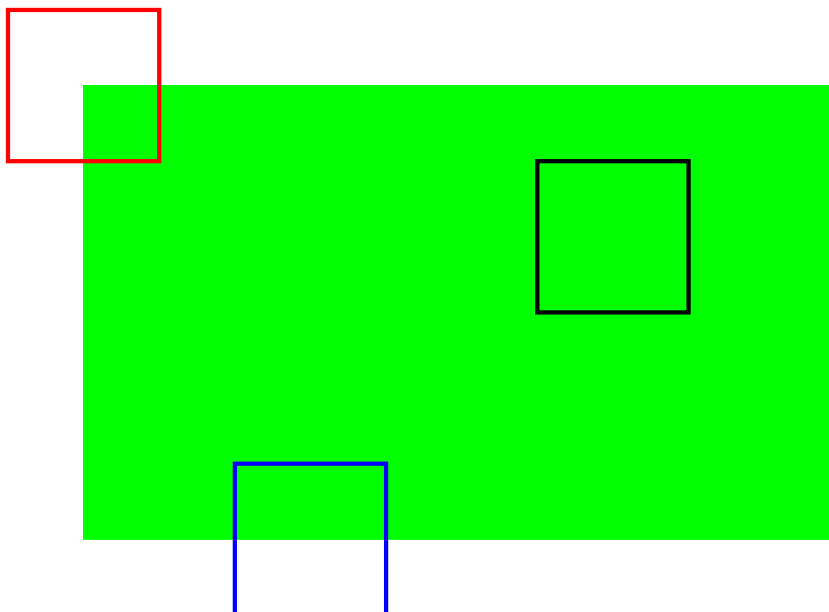
Po priradení pixelov určitej krivke ostáva určiť ktoré gausiány a tým padom aj pixely patria do popredia alebo pozadia. To je založené na predpoklade, že pixely v pozadí majú rovnaké alebo pomaly sa meniace parametre. Gaussové krivky sú zoradené podľa hodnoty ω/σ . Toto usporiadanie modelu predstavuje zoznam s otvoreným koncom, kde najpravdepodobnejšie distribúcie predstavujúce pozadie zostávajú na vrchole a menej pravdepodobné alebo nové distribúcie sú na spodku. Potom prvých B distribúcií je prehlásených za pozadie pomocou rovnice

$$B = \operatorname{argmin}_b \left(\sum_{k=1}^b \omega_k > T \right), \quad (3.11)$$

kde T je predom zvolený prah určujúci aká veľká časť dát je potrebná, aby sa mohlo jednat o pozadie.

3.2 Detekcia bodov vhodných k sledovaniu

Pri sledovaní pohybujúcich sa objektov je nutné vyhľadať vhodné body, ktoré sú ľahko detekovateľné v pozorovanej sekvencii videa. Akonáhle sú takéto body nájdené v aktuálnom snímku, je možné vyhľadať rovnaké body v ďalších snímkoch. Preto tieto body musia byť



Obr. 3.1: Typy oblastí: čierne okno obsahuje plochú oblasť, modré obsahuje hranu a červené roh

nemenné pre rotáciu, posunutie (transláciu), intenzitu a zmenu mierky. Pri pohľade na obrázok 3.1 vidíme tri vyznačené oblasti. Čierna oblasť predstavuje čistú plochu, ktorej presnú pozíciu je nemožné vyhľadať a sledovať v ďalšom snímku. Nech sa pohne kamkoľvek, vždy bude vyzerat rovnako. Modrá oblasť je umiestnená na hrane. Ak sa pohne vertikálne, jeho oblasť sa zmení, ale pri horizontálnom pohybe oblasť stále zostáva rovnaká. Tým sa omeďzuje prestor výskytu oblasti pri pohybe iba na horizontálnu os. Posledná červená oblasť je umiestnená v rohu. Kamkoľvek sa táto oblasť pohne, vždy bude iná, čo znamená že je jedinečná. Rohy sú teda považované za najvhodnejšie body v obraze, ktoré je možné použiť napríklad pri ich sledovaní. Hlavnou úlohou je teda vyhľadať také body, aby splňovali predchádzajúce požiadavky. V ďalších podkapitolách si detailnejšie popíšeme princíp Harrisovho detektoru [15] a Shi-Tomasiho detektoru [11].

3.2.1 Harrisov detektor

Jeden z prvých pokusov pre vyhľadanie rohov bol predstavený Chrisom Harrisom a Mikeom Stephensom v publikácii A combined Corner and Edge Detector[6], dnes nazývaný (Harris Corner Detector) Harrisov detektor. Hlavnou úlohou algoritmu je nájsť malé oblasti v obraze takzvané okná, ktoré spôsobia veľmi veľkú zmenu intenzity pri pohybe v X a Y smere.

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (3.12)$$

Stred okna je umiestnený na pozícií (x, y) . Intenzita pixelu na tomto mieste je $I(x, y)$. Ak sa toto okno mierne premiestni na novú pozíciu s posunom (u, v) , intenzita pixelu na novej pozícií bude $I(x + u, y + v)$. Potom $[I(x + u, y + v) - I(x, y)]$ vyjadruje rozdiel v intenzite pri posunutí okna. Funkcia okna $w(x, y)$ definuje váhy pixelov v požadovanom okne. $E(u, v)$ vyjadruje rozdiel medzi originálnym a posunutým oknom. Potrebujeme vyhľadať okná s veľkou hodnotou E , čo znamená nájdenie vysokých hodnôt v predchádzajúcom vzorci medzi

hranatými zátvorkami. Výpočet rovnice (3.12) by bol ale veľmi pomalý, preto sa použije Taylorov rozvoj pre úpravu vzťahu v hranatých zátvorkách

$$E(u, v) \approx \sum_{x,y} [I(x, y) + uI_x + vI_y - I(x, y)]^2. \quad (3.13)$$

Jedná sa o prepísanie pôvodného vzťahu použitím derivácie, kde I_x a I_y sú deriváty obrazu v X a Y smere. Ďalšou matematickou úpravou dostaneme výsledný vzorec:

$$E(u, v) \approx [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}, \quad (3.14)$$

kde

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}. \quad (3.15)$$

Po nájdení všetkých okien s veľkými zmenami, je nutné rozhodnúť, ktoré okná obsahujú rohy a ktoré nie. Preto je nutné vypočítať skóre pre každé okno:

$$R = \det(M) - k(\text{trace}(M))^2, \quad (3.16)$$

kde $\det(M) = \lambda_1 \lambda_2$, $\text{trace}(M) = \lambda_1 + \lambda_2$. λ_1 a λ_2 sú vlastné hodnoty M a k je konštanta s hodnotou medzi 0.04 – 0.06. Vlastné hodnoty M rozhodujú či oblasť obsahuje plochu, hranu alebo roh.

- Keď $\|R\|$ je malé, čo sa stane keď λ_1 a λ_2 je malá, oblasť obsahuje plochu
- Keď $R < 0$, čo sa stane keď $\lambda_1 \gg \lambda_2$ alebo naopak, oblasť obsahuje hranu
- Keď R je veľké, čo nastane keď λ_1 a λ_2 sú veľké približne rovnaké, oblasť obsahuje roh

3.2.2 Shi-Tomasiho Detektor

V predchádzajúcej podkapitole sme si popísali Harrisov detektor. J. Shi a C. Tomasi predstavili malú modifikáciu na predchádzajúci algoritmus v práci Good Geatures to Track [11] čím dosiahli lepšie výsledky v porovnaní s Harrisovým detektorom. Shi-Tomasiho detektor je podobný s predchádzajúcim až na časť, kde sa počíta skóre (R). Pozmenili jeho výpočet tak, aby pre určenie či pixel bol roh boli použité iba vlastné hodnoty. Namiesto predchádzajúceho vzťahu (3.16), Shi-Tomasi navrhli:

$$R = \min(\lambda_1, \lambda_2). \quad (3.17)$$

Pokiaľ výsledná hodnota R je väčšia ako prahová hodnota, potom okno obsahuje roh. λ_1 a λ_2 spolu s prahom vytvoria tri možné oblasti:

- Prvá oblasť je považovaná za roh, iba ak obidve vlastné hodnoty λ_1 a λ_2 sú nad minimálnou hodnotou λ_{min}
- Druhá oblasť je považovaná za hranu, pokiaľ jedna z vlastných hodnôt je menšia ako požadované minimum
- Posledná oblasť obsahuje plochu, ak obidve vlastné hodnoty sú pod požadovaným prahom

V knižnici OpenCV je táto metóda implementovaná vo funkcii `cv2.goodFeaturesToTrack()`. V tejto funkcii je možné definovať maximálny počet najlepších rohov, čo môže byť užitočné ak nechceme detekovať každý roh v obraze.

3.3 Optický tok

Optický tok popisuje zmeny v obraze spôsobené zdanlivým pohybom objektov v dvoch po sebe idúcich snímkoch v časovom intervale dt spôsobené pohybom objektu alebo kamery. To vykonáva na základe porovnávania zmien jasov po sebe idúcich obrazov v pozorovanej scéne. Optický tok teda predstavuje dvojrozmerné pole vektorov, kde každý vektor ukazuje posun pixelu v aktuálnom snímku na nasledujúci snímok. Jedná sa teda o dvojrozmerný vektor rýchlosti, pretože nesie informácie o smere a veľkosti rýchlosti pohybu jednotlivých pixelov naprieč obrazom. Použitie optického toku sa teda najčastejšie využíva v oblasti sledovania objektov.

Existujú dva hlavné prístupy pre popis optického toku v obraze. Prvým je optický tok, ktorý sa počíta pre každý pixel v obraze. Takýto optický tok nazývame hustým optickým tokom. Tento prístup nielenže je časovo náročný, ale je aj vo väčšine prípadov nemožný. Ako sme si popísali vyššie, obraz vieme rozdeliť na oblasti obsahujúce plochy, hrany a rohy. Preto pri pohybe objektu, ktorý má rovnaké jasové hodnoty, ako napríklad čistý list papiera, nevieme určiť pre každý bod tohto listu jeho odpovedajúci pohybový vektor. Optický tok vieme určiť len pre rohy alebo hrany, ktoré sú kolmé na pohyb objektu. Druhým prístupom je takzvaný riedky optický tok. Ten počíta optický tok len pre body ktoré obsahujú práve kolmé hrany alebo rohy. Tento prístup používa väčšina aplikácií. V reálnych podmienkach nie je nutné sledovať pohyb všetkých pixelov ale iba dôležitých objektov, čo samozrejme znižuje aj výpočetné nároky.

Výpočet optického toku je založený na dvoch predpokladoch:

1. Jasová hodnota pixelu pozorovaného objektu zostáva rovnaká vo všetkých po sebe idúcich snímkoch.
2. Okolné body v obraze sa pohybujú podobným smerom.

Predpokladajme, že máme pixel $I(x, y, t)$ v prvom snímku odkazujúci na bod v šedi na pozícií (x, y) a v čase t . Tento pixel sa posunie o vzdialenosť (dx, dy) v nasledujúcom snímku za čas dt . Na základe prvého predpokladu kde rovnaké pixely majú aj rovnakú intenzitu môžeme tento vzťah vyjadriť nasledovne:

$$I(x, y, t) = I(x + dx, y + dy, t + dt). \quad (3.18)$$

Pravú stranu tejto rovnice upravíme pomocou taylorovej rady a získame novú rovnicu

$$f_x u + f_y v + f_t = 0, \quad (3.19)$$

kde

$$f_x = \frac{\partial f}{\partial x}; \quad f_y = \frac{\partial f}{\partial y}; \quad u = \frac{dx}{dt}; \quad v = \frac{dy}{dt}. \quad (3.20)$$

Rovnica (3.19) je nazývaná rovnicou optického toku. V nej môžeme nájsť hodnoty f_x a f_y predstavujúce gradient obrazu. Podobne aj f_t je gradient pozdĺž času. V rovnici sú ale dve neznáme (u, v) . Nemôžeme vyriešiť dvojicu neznámich len s pomocou jednej rovnice preto existuje zopár metód ako ich vyriešiť. Jednou z nich je aj metóda Lucasa-Kanadeho.

3.3.1 Lucas-Kanade metóda

Metóda Lucas-Kanade [9] je široko používaná v diferenciálnej metóde pre odhad optického toku u počítačového videnia. Je založená na predpoklade, že všetky okolné pixely budú mať

podobný smer pohybu. Lucas-Kanade metóda používa okolie o veľkosti 3×3 okolo sledovaného bodu. Všetkých deväť bodov má teda podobný smer. Pri využití okolia uvažovaného pixelu získame 9 rovníc s dvoma neznámymi premennými, čo nám umožní zistiť hodnoty týchto premenných, ktoré predstavujú rýchlostný vektor stredového pixelu p . Táto metóda pre získanie lepšieho výsledku rieši rovnicu (3.19) optického toku pre všetky pixely v okolí, pomocou metódy najmenších štvorcov. Taktiež metóda predpokladá malý posun pixelu a jeho okolia medzi dvomi po sebe idúcimi snímkami

$$\begin{aligned} f_x(p_1)u + f_y(p_1)v &= -f_t(p_1) \\ f_x(p_2)u + f_y(p_2)v &= -f_t(p_2) \\ &\vdots \\ f_x(p_n)u + f_y(p_n)v &= -f_t(p_n), \end{aligned} \tag{3.21}$$

kde p_1, p_2, p_n sú pixely patriace okoliu sledovaného bodu, $f_x(p_n), f_y(p_n), f_t(p_n)$ parciálne derivácie obrazu I vzhľadom na polohu x, y a čas t , vyhotovené v bode p_n . Tieto rovnice môžu byť prepísané do maticovej formy $A\vartheta = b$, kde:

$$A = \begin{bmatrix} f_x(p_1) & f_y(p_1) \\ f_x(p_2) & f_y(p_2) \\ \vdots & \vdots \\ f_x(p_n) & f_y(p_n) \end{bmatrix}; \quad \vartheta = \begin{bmatrix} u \\ v \end{bmatrix}; \quad b = \begin{bmatrix} -f_t(p_1) \\ -f_t(p_2) \\ \vdots \\ -f_t(p_n) \end{bmatrix}. \tag{3.22}$$

Po úprave získame výsledný tvar rovnice pre výpočet rýchlostného vektoru.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_x(p_i)^2 & \sum_i f_x(p_i)f_y(p_i) \\ \sum_i f_y(p_i)f_x(p_i) & \sum_i f_y(p_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_x(p_i)f_t(p_i) \\ -\sum_i f_y(p_i)f_t(p_i) \end{bmatrix} \tag{3.23}$$

Táto metóda sa v praxi používa na sledovanie objektov. Vyhľadajú sa vhodné body v objekte a následne sa rozhodne na základe porovnania aktuálnej a predchádzajúcej snímky, či sa jedná o ten istý bod alebo nie. Výpočet je ale úspešný iba v prípade malého pohybu a zlyhá ak sa pozorovaný objekt posunie v nasledujúcom snímku o veľkú vzdialenosť. Tento problém rieši pyramídové vyjadrenie jednotlivých snímok. Keď ideme nahor v pyramíde čo reprezentuje zmenšenie obrazu, malé pohyby v pôvodnom snímku sa postupne zanedbávajú a z veľkých pohybov sa stávajú malé pohyby. Potom pri aplikácii Lucas-Kanadeho metódy na zmenšený obraz dostaneme optický tok spoločne s jeho odpovedajúcou mierkou.

Kapitola 4

Návrh systému

V tejto kapitole si popíšeme návrh celého systému a detailnejšie si rozoberieme činnosť jeho jednotlivých častí. Hlavnou úlohou tejto práce je získanie kalibračných informácií zo vstupného videa bez použitia dodatočných informácií zadaných užívateľom pred spustením aplikácie a následné meranie rýchlosti prechádzajúcich vozidiel.

Blokovú schému celého navrhovaného systému je znázornená na obrázku 4.1. Jediný vstup predstavuje video záznam z dohľadovej kamery. Toto video je načítavané pomocou funkcie `cv2.VideoCapture()` z knižnice OpenCV a následne spracovávané snímok po snímku. Zaujímavejší je proces získavania kalibračných informácií. Tie je nutné zo záznamu vyhľadať ako prvé, aby sme sa dostali k bloku meranie rýchlostí. Časť kalibračného bloku, konkrétne detekcia druhého úbežníka potrebuje na svoju správnu funkciu informácie o pozícií pohybujúcich sa vozidlách, preto na jej vstup sa posiela odpovedajúca maska popredia. Kalibračný blok sa spracúva len do momentu získania všetkých potrebných údajov. Potom sa stáva nadbytočným a zbytočne by spomaľoval činnosť ďalších blokov.

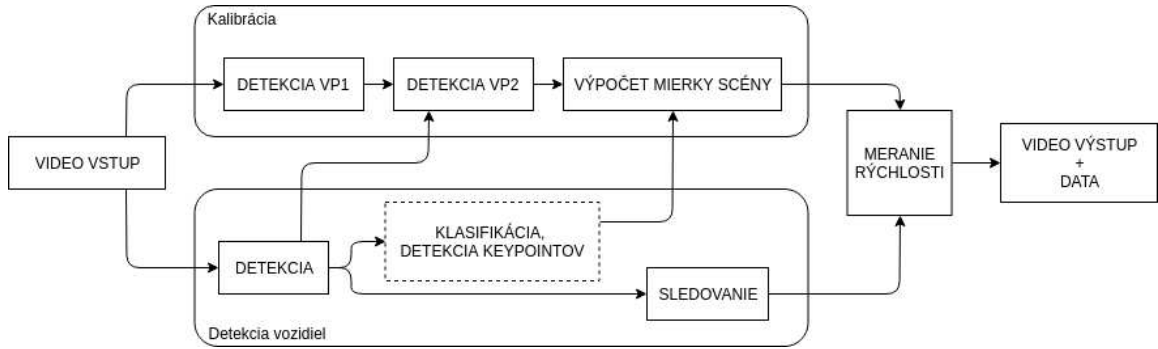
Druhým hlavným blokom je detekcia a sledovanie vozidiel. Tá obsahuje blok pre klasifikáciu vybraných typov vozidiel a detekciu kľúčových bodov, ktorých odpovedajúce pozície sa použijú na výpočet mierky scény. Tento blok je znázornený čiarkovane, čo znamená, že v práci sa aktívne nevykonáva. Používajú sa už nadetegované informácie z *BCS_keypoints_2D_3D* k videám z datasetu *BrnoCompSpeed* [13].

Poslednou ale najhlavnejšou komponentou systému je meranie rýchlosti, ktorá zo známych pozícií vozidiel spolu s kalibračnými informáciami dopočíta odpovedajúce rýchlosti. Tie spoločne s videom predstavujú výstup celého systému.

4.1 Kalibrácia

Hlavným cieľom je automaticky kalibrovať kameru bez akýchkoľvek manuálne zadaných informácií. Metóda, pomocou ktorej je kamera kalibrovaná je založená na publikácií [4]. Kalibrácia je založená na detekcii úbežníkov, ktoré sa určia na základe pohybu vozidiel. Úbežník ktorý je rovnobežný s pohybom vozidiel je označovaný ako \mathbf{u} . Druhý úbežník \mathbf{v} je orientovaný kolmo na smer idúcich vozidiel a tretí \mathbf{w} je kolmý na rovinu vozovky.

Pred začatím samotnej kalibrácie je nutné, aby záznam spĺňal niekoľko predpokladov. Základný predpoklad je aby snímaný úsek cesty bol (približne) rovný, a aby väčšina vozidiel sa pohybovala rovnobežne s vozovkou. Ďalšími predpokladmi sú: žiadne skreslenie kamery, štvorcové pixely a hlavný bod v strede obrazu. Tieto predpoklady väčšinou spĺňajú všetky dopravné kamery.



Obr. 4.1: Schéma blokov so vzájomnou komunikáciou pre navrhovaný systém

Cieľom kalibrácie je vypočítať ohniskovú vzdialenosť f zo získaných úbežníkov. Proces detekcie prvého a druhého úbežníka spolu s dopočítaním tretieho a optickej vzdialenosti je popísaný v ďalších kapitolách, ale najskôr sa zoznámime s nástrojom *Diamond Space*.

4.1.1 Diamond Space

Pre určenie prvých dvoch úbežníkov je nutné nájsť globálne maximum z veľkého množstva pretínajúcich sa čiar. Metóda *Diamond Space* (ďalej len *DS*) predstavená Dubskou a Heroutom [16] je použitá na vyhľadanie tohto priesečníka pomocou mapovania odpovedajúcich priamok, do konečného systému, v ktorom maximálna hodnota určuje výslednú pozíciu úbežníka. Túto metódu používajú aj práce pani Dubskej[4] alebo Sochora [12] pre získanie kalibračných informácií keďže nepotrebuje žiadne dodatočné informácie pre presnú kalibráciu kamery.

Diamond Space mapuje priamky z nekonečného súradnicového systému do kriviek v konečnom súradnicovom systéme. Pri tomto mapovaní sa používa paralelný súradný systém. V paralelnom súradnicovom systéme sa body zobrazujú ako priamky a priamky ako body. Táto transformácia sa v *DS* vykoná dvakrát. Pri transformácii sa mapuje bod na priamku a následne znova na bod alebo opačne priamka-bod-priamka. Princíp fungovania postupnej transformácie je zobrazený na obrázku 4.2.

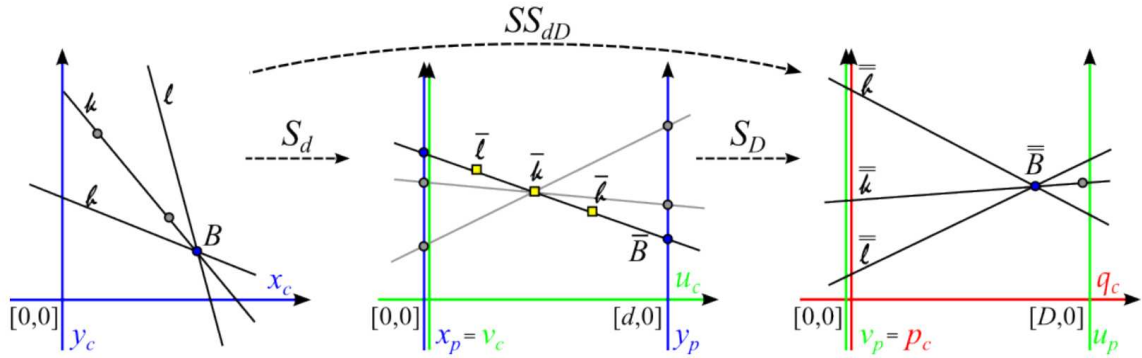
Tieto transformácie môžu byť vykonané dvomi spôsobmi. V paralelnom súradnom systéme môžu byť osi orientované rovnakým alebo rozdielnym smerom. Pri rovnakom smere označujeme priestor ako \mathcal{S} (straight), pri rozdielnom smere \mathcal{T} (twisted). Mapovanie do paralelného systému je možné vykonať ich kombináciou. Kombináciou rovnakého \mathcal{S} alebo rozdielneho \mathcal{T} systému získame štyri možné spôsoby transformácie: \mathcal{SS} , \mathcal{ST} , \mathcal{TS} a \mathcal{TT} . Tým vzniknú trojuholníkové podpriestory. Usporiadáním týchto trojuholníkových podpriestorov tak aby boli prepojené v závislosti na totožnosti príslušných osí získame výsledný priestor pripomínajúci diamant (od čoho je odvodený názov). Práve týmto prepojením všetkých štyroch transformácií je dosiahnutá konečnosť výsledného súradného systému.

Každá priamka (a, b, c) definovaná rovnicou $ax + by + c = 0$ je v transformovaná na zloženú priamku

$$\alpha = \text{sgn}(ab), \beta = \text{sgn}(bc), \gamma = \text{sgn}(ac) \quad (4.1)$$

$$(a, b, c) \rightarrow \left[\frac{\alpha a}{c+\gamma a}, \frac{-\alpha c}{c+\gamma a} \right], \left[\frac{b}{c+\beta b}, 0 \right], \left[0, \frac{b}{a+\alpha b} \right], \left[\frac{-\alpha a}{c+\gamma a}, \frac{\alpha c}{c+\gamma a} \right]. \quad (4.2)$$

Výsledkom sú štyri body, ktoré určujú polohu zloženej priamky v priestore *DS*. Táto krivka je vložená do priestoru pomocou inkrementovania pixelov čím hlasuje za polohu úbežníka.



Obr. 4.2: Dve postupné Houghove transformácie bodov a priamok do paralelného súradného systému \mathcal{S} (Obr. prevzatý z [16])

Bod za ktorý hlasovalo najviac kriviek sa vyhľadá pomocou globálneho maxima. Tento nájdený bod sa z paralelného súradnicového systému DS transformuje do homogénneho súradnicového systému:

$$[x, y, w]_d \rightarrow [Dy, \operatorname{sgn}(x)dx + \operatorname{sgn}(y)Dy - dDw, x]_o, \quad (4.3)$$

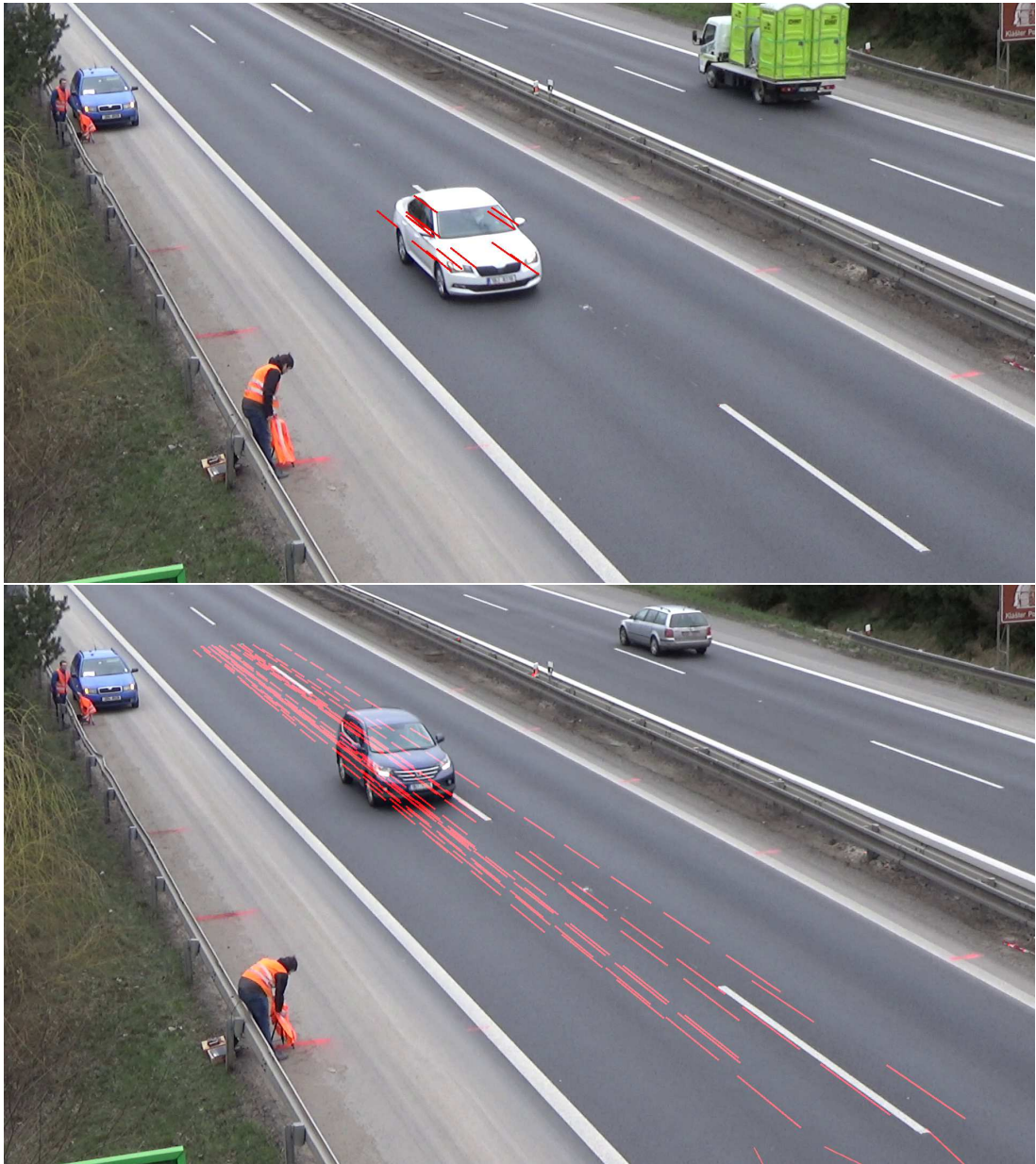
čím sa získa pozícia príslušného úbežníku.

4.1.2 Detekcia prvého úbežníka

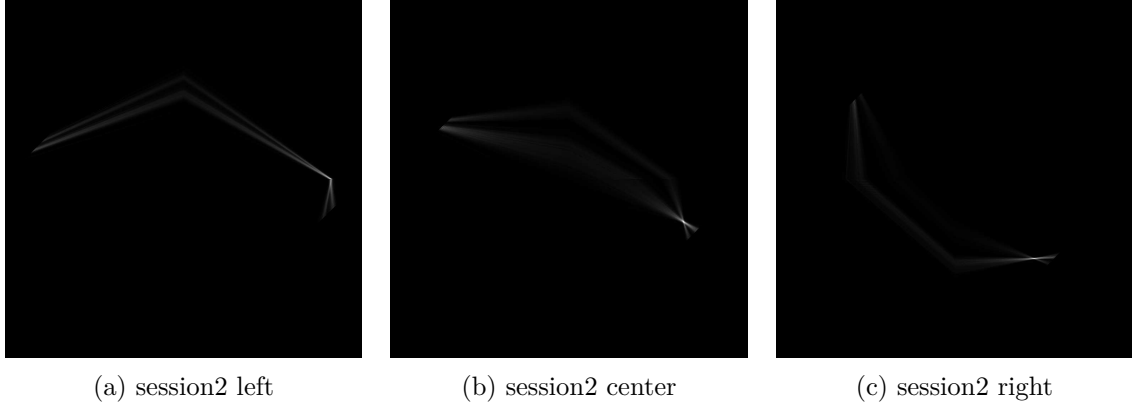
K získaniu prvého úbežníka \mathbf{u} , do ktorého smerujú deliace čiary na vozovke alebo pohybujúce sa vozidlá v priamom smere jazdy je možné využiť práve sledovanie kľúčových bodov na pohybujujúcich sa vozidlách. Prvým krokom je identifikovať vhodné body k sledovaniu. To sa deje pomocou funkcie `goodFeaturesToTrack()` implementovanej v knižnici OpenCV, ktorá tieto body vyhľadá vo vstupnom snímku. Naň môže byť aplikovaná maska, ktorá omezuje priestor vyhľadávania iba na pohybujúce sa objekty. Ďalším krokom je sledovanie zvolených bodov pomocou optického toku v obraze. K tomu je využitá metóda Lucas-Canade ktorá je taktiež implementovaná v knižnici OpenCV. Každý bod sa sleduje po určitú časovú dobu (počet snímkov n). Daný časový interval je rozumne zvolený aby sledovaný bod prešiel dostatočne veľkú vzdialenosť kvôli vyššej presnosti odpovedajúceho smerového vektora.

Pri sledovaní daného bodu sa v čase t_s uloží počiatočná pozícia daného bodu (x_{t_s}, y_{t_s}) a v čase t_e posledná pozícia sledovaného bodu (x_{t_e}, y_{t_e}) . Po uplynutí času t_e sa získané pozície uložia do zoznamu úsečiek. Na obrázku 4.3 je možné vidieť opovedajúce úsečky po sledovaní kľúčových bodov. Zo zoznamu sa odstránia všetky úsečky, ktorých vzdialenosť je menšia ako zvolená prahová hodnota. Tým sa odstránia body, ktoré boli chybné identifikované na statických objektoch ako napríklad anotácie na ceste.

Z dvojice počiatočných a koncových bodov úsečky sa vyjadry všeobecná rovnica priamky. Táto rovnica sa predá akumulátoru DS , ktorého činnosť je popísaná v predchádzajúcej podkapitole 4.1.1. Po skončení zhromažďovania priamok sa v priestore DS vyhľadá maximum, ktorého pozícia určuje polohu prvého úbežníka. Je dôležité určiť, kedy sa má ukončiť akumulovanie priamok aby bolo výsledné riešenie dostatočne presné. V tejto práci je použité omedzenie počtom priamok, ale sú možné aj iné alternatívy ako ukončenie vyhľadávania po určitom časovom intervale alebo po dosiahnutí globálneho maxima v DS priestore. Ukážky DS priestorov pre detekciu prvých úbežníkov sú zobrazené na obrázku 4.4.



Obr. 4.3: Nájdenie a sledovanie vhodných bodov pre detekciu prvého úbežníka.



Obr. 4.4: Vzisknuté diamond space priestory pre detekciu prvého úbežníka

4.1.3 Detekcia druhého úbežníka

Vyhľadanie druhého úbežníka \mathbf{v} musí začať až po úspešnom nájdení prvého úbežníka, lebo prvý omedzuje oblasť v priestore DS v ktorom sa hľadaný úbežník môže nachádzať. Tento úbežník je kolmý na smer pohybujúcich sa vozidiel a leží na rovine vozovky. Jeho detekcia je náročnejšia oproti prvému úbežníku, pri ktorom sa využíva smer jazdy pohybujúcich sa vozidiel.

Mnoho hrán na vozidlách smeruje práve k tomuto úbežníku. Preto je tieto hrany nutné vyhľadať len na pohybujúcich sa vozidlách. Pre rozlíšenie pohybujúcich sa vozidiel je použitý algoritmus zmesi Gaussových kriviek, ktorého činnosť je popísaná v podkapitole 3.1.4. Pomocou morfológických operácií sa získaná maska popredia vyčistí od šumu a vyplnenia sa diery vo vnútri objektu záujmu.

Po úspešnej detekcii popredia sa pomocou Sobelovho operátora vyhľadajú hrany obrazu v horizontálnom aj vertikálnom smere. Tým sa získajú gradienty a magnitudy. Magnitudy týchto modelov sa sčítajú. Body, ktoré obsahujú najväčšie hodnoty, sú súčasťou nejakej hrany v obraze. Tieto body je nutné vymaskovať s popredím, aby sa ďalej pracovalo len s hranami ktoré sú súčasťou pohybujúcich sa vozidiel. Detegované body sa ďalej filtrujú, aby sa využili len tie najsilnejšie body s magnitudou nad zvolenou prahovou hodnotou. Orientácia hrany by mohla byť spočítaná pomocou gradientov, ale kvôli malému okoliu by bol výsledný uhol nepresný. Preto sa využije okolie 9×9 okolo bodu $s = [x, y]$ a zostojí sa matica X .

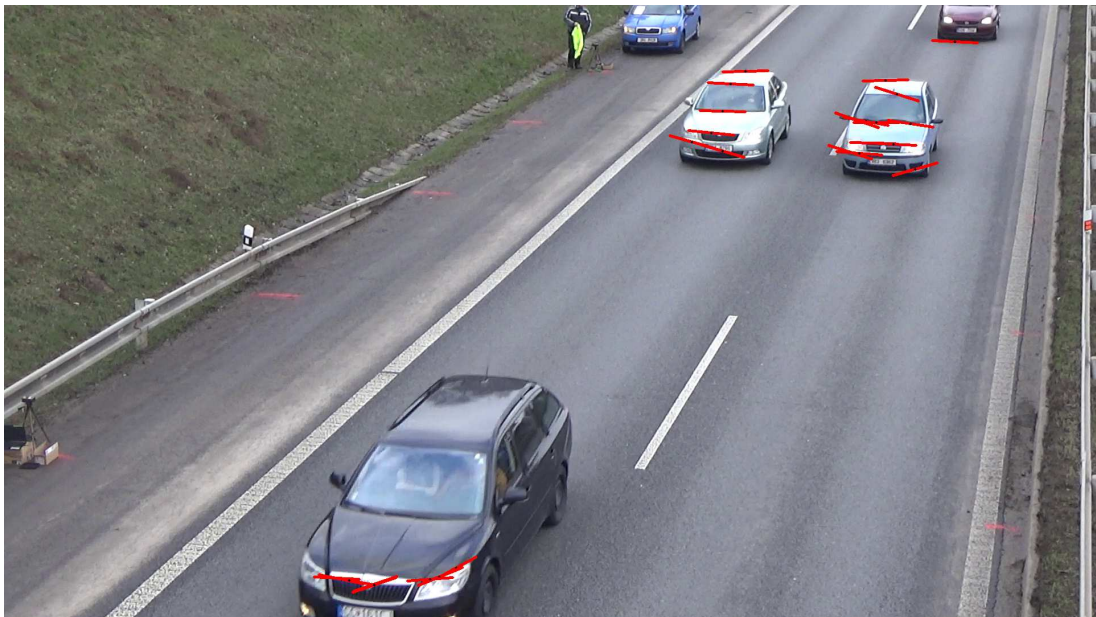
$$X = \begin{bmatrix} w_1(m_1 - x) & w_1(n_1 - y) \\ w_2(m_2 - x) & w_2(n_2 - y) \\ \vdots & \vdots \\ w_k(m_k - x) & w_k(n_k - y) \end{bmatrix}, \quad (4.4)$$

kde $[m_k, n_k]$ sú súradnice okolitých pixelov ($k = 1 \dots 81$) a w_k je ich odpovedajúca veľkosť magnitudy. Novovzniknutá matica X bude mať veľkosť 81×2 . Z nej sa spočíta singulárny rozklad matice:

$$W\Sigma^2W^T = SVD(X^T X), \quad (4.5)$$

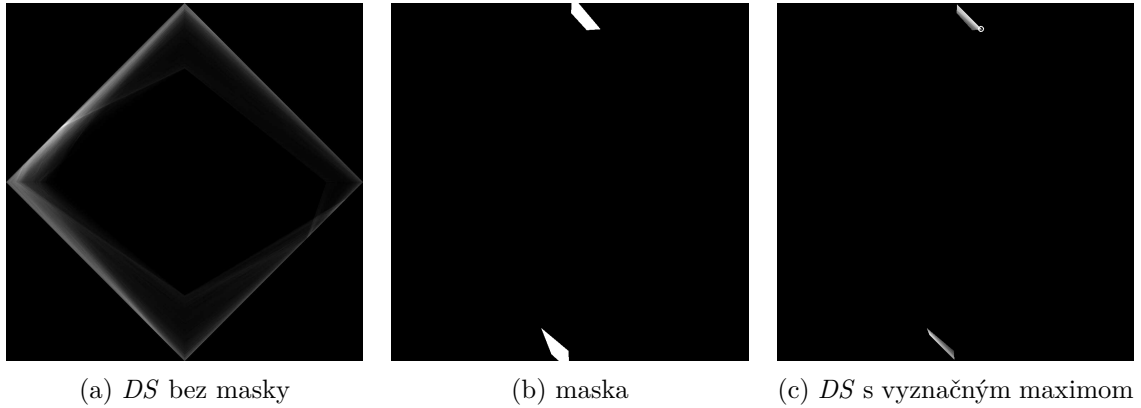


(a) Detegované hrany na vozidlách



(b) Orientácia hrán na vozidlách

Obr. 4.5: Detekcia hrán na vozidlách a výpočet ich smeru v lokálnych maximách, pre určenie druhého úbežníka



Obr. 4.6: Zobrazenie *DS* priestoru pre detekciu 2 úbežníka s aplikovaním masky

kde

$$W = [a_1, a_2] \quad (4.6)$$

$$\Sigma = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}. \quad (4.7)$$

Vektory a_1, a_2 predstavujú vlastné vektory a λ_1, λ_2 predstavujú vlastné hodnoty matice X . Smerový vektor hrany je daný vektorom a_1 a kvalitu tejto hrany je možné zistiť pomocou $\frac{\lambda_1}{\lambda_2}$. [12].

Akonáhle sú získané smerové vektory hrán s ich kvalitou, je nutné tieto hrany ďalej filtrovať. Hrany s kvalitou nižšou ako prah sú zahodené. Taktiež hrany, ktoré smerujú k prvému úbežníku alebo majú vertikálny smer, nie je nutné ďalej spracúvať. Preto sa použijú iba hrany s orientáciou v rozmedzí $< -40^\circ, +40^\circ >$. Príklad detekcie hrán so znázorneným smerom, ktoré hlasujú za druhý úbežník sú ukázané na obrázku 4.5. Opäť sa vyhovujúce hrany predávajú do *DS* priestoru.

Pred vyhľadáním maxima sa musí zostrojiť maska, ktorá omezuje priestor výskytu úbežníka. Tá je daná rozsahom ohniskovej vzdialenosti a uhlom horizontu. So známou pozíciou prvého úbežníka sa pre každú možnú pozíciu druhého úbežníka v *DS* priestore spočíta ohnisková vzdialenosť a uhol horizontu, a pokiaľ nepadne do zvoleného rozsahu, tak je táto pozícia vyradená z hľadania maxima. Príklad diamond space priestoru pred a po maskovaní je znázornený na obrázku 4.6. Po naakumulovaní dostatočného počtu priamok sa vyhledá maximum ktoré opovedá druhému úbežníku.

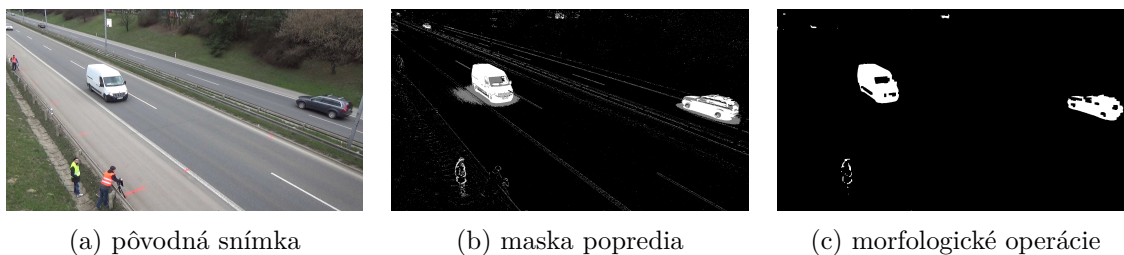
4.1.4 Výpočet tretieho úbežníka a ohniskovej vzdialenosti

Po nájdení prvých dvoch úbežníkov (\mathbf{u} a \mathbf{v}) vieme dopočítať ohniskovú vzdialenosť a pozíciu tretieho úbežníka \mathbf{w} , ktorého smer je kolmý na rovinu vozovky. Tento výpočet je možný len za predpokladu, ak pozícia hlavného bodu je v strede obrazu.

Ohnisková vzdialenosť sa vypočíta pomocou nasledujúceho vzorca:

$$f = \sqrt{-(U - P) \cdot (V - P)}, \quad (4.8)$$

kde $U = (u_x, u_y)$ je pozícia prvého úbežníku, $V = (v_x, v_y)$ predstavuje pozíciu druhého úbežníku a $P = (p_x, p_y)$ je pozícia hlavného bodu obrazu.



Obr. 4.7: Proces vyhladenia masky popredia pomocou postupných morfológických operácií

So známou ohniskovou vzdialenosťou je možné vypočítať svetové pozície úbežníkov U' , V' , W' a obrazovú pozíciu tretieho úbežníka W .

$$P' = (p_x, p_y, 0) \quad (4.9)$$

$$U' = (u_x, u_y, f) \quad (4.10)$$

$$V' = (v_x, v_y, f) \quad (4.11)$$

$$W' = (U' - P') \times (V' - P') \quad (4.12)$$

$$W = \left(\frac{w'_x}{w'_z} f + p_x, \frac{w'_y}{w'_z} f + p_y \right) \quad (4.13)$$

Po vypočítaní ohniskovej vzdialenosti a tretieho úbežníka je kamera plne kalibrovaná.

4.2 Detekcia a sledovanie vozidiel

Detekcia vozidiel je založená na rozpoznaní pohybujúcich sa objektov v obraze pomocou metódy zmesy Gaussových kriviek [14]. Tá rozdelí oblasť pozorovanej scény na pozadie obsahujúce statické objekty v obraze a popredie obsahujúce dynamické zmeny v obraze oproti predchádzajúcim snímkom. Tým získame masku popredia, ktorú je nutné ďalej spracovať. Masku totiž obsahuje aj nežiadúci šum spôsobený vonkajšími podmienkami (zmena jasu, pohyb listov). Pomocou morfológických operácií otvárania, uzatvárania a dilatácie sa z tejto masky odstáni náhodný šum a z časti sa uzavrujú diery vo vnútri objektu. Postupný proces vyhladenia masky popredia je zobrazený na obrázku 4.7.

Po získaní upravenej masky pre každý snímok je nutné v ňom detekované objekty sledovať. To sa deje pomocou algoritmu založenom na sledovaní ťažiska. V maske sa vyhľadajú obrysy pre všetký vozidlá a vypočítajú sa ich ťažiská (centroids). Sledovanie ťažiska je založené na euklidovskej vzdialenosti medzi existujúcim ťažiskom vozidla detekovanom v predchádzajúcej snímke a novým ťažiskom vozidla v aktuálnej snímke.

Pri prvom výskyte ťažiska sa vytvorí nový objekt predstavujúci nové vozidlo, ktoré nesie informácie o polohe ťažiska, čísla vozidla (ID) a časovač, po ktorého uplynutí sa vozidlo pokladá za stratené a teda mimo obraz záznamu. Pri ďalšom výskyte v nasledujúcom snímku pred vytvorením nového vozidla musíme rozhodnúť, či sa tento nový objekt nedá priradiť k niektorému už existujúcemu vozidlu. Predpokladom pri tejto metóde je, aby vzdialenosť medzi ťažiskom objektu v snímku F_{t+1} a predchádzajúcim ťažiskom v snímku F_t bola menšia ako veľkosť boxu ohraničujúca toto vozidlo. Nová poloha ťažiska sa teda porovná so všetkými registrovanými vozidlami v obraze, a pokiaľ je táto podmienka splnená pre niektoré vozidlo, tak sa jedná o totožný objekt pre ktorý sa aktualizuje jeho nová poloha. V opačnom prípade sa jedná o scela nové vozidlo, ktoré sa pridá do zoznamu sledovaných objektov.

Z tohto zoznamu je nutné vozidlá ktoré sa stratia alebo sú mimo oblasti záujmu odstrániť. Každý objekt má počítadlo, ktorého hodnota sa inkrementuje s každým novým snímkom. Pokiaľ sa existujúcemu vozidlu aktualizuje jeho poloha, počítadlo sa vynuluje. Po dosiahnutí prahovej hodnoty počítadla sa objekt vymaže zo zoznamu.

4.3 Výpočet vzdialenosti

Základnou úlohou aby sme mohli merať rýchlosť vozidiel je vedieť vypočítať vzdialenosť ktorú prešli. K tomu potrebujeme premietnuť pozíciu ľubovoľného bodu $X = [x, y, 1]$ v obraze na svetovú pozíciu $X' = [x', y', z']$ v rovine vozovky.

Ako prvé sa určí rovina φ ktorá je rovnobežná s rovinou vozovky. Po kalibrovaní kamery vieme pozíciu tretieho úbežníka, ktorého smer je kolmý na túto rovinu. Normalizovaný vektor roviny n_φ vypočítame pomocou nasledujúce vzťahu:

$$n_\varphi = \frac{W'}{\|W'\|}. \quad (4.14)$$

Nepoznáme len vzdialenosť d medzi rovinou a kamerou, preto sa táto hodnota nastaví ľubovoľne (najčastejšie $d = 1$), čo má za následok, že vzdialenosť medzi bodmi nebude odpovedať reálnym jednotkám. Pre meranie vzdialenosti v reálnych jednotkách je nutné získať mierku scény popísané v nasledujúcej kapitole. Akonáhle poznáme parametre roviny φ , môžeme bod premietnuť do roviny vozovky:

$$\rho(X) = \varphi \cap \overleftarrow{OX'}, \quad (4.15)$$

kde $O = [p_x, p_y, 0]$ predstavuje polohu kamery pričom $[p_x, p_y]$ sú pozície hlavného bodu a $X' = [x, y, f]$ je pozícia bodu v obraze spolu s ohniskovou vzdialenosťou. Vzdialenosť medzi dvoma bodmi X a Y je:

$$d_r(X, Y) = |\rho(X)\rho(Y)|. \quad (4.16)$$

4.4 Výpočet mierky scény

Aby bolo možné merať vzdialenosť v metroch, je nutné nájsť hodnotu λ vyjadrujúcu mierku scény. K tomu potrebujeme vedieť reálnu vzdialenosť v rovine vozovky. Na určenie mierky používam dataset *BCS_keypoints_2D_3D* obsahujúci kľúčové body na vozidlách ktoré boli nadetegované pre dataset *BrnoCompSpeed*. Tento dataset obsahuje detekované 2D kľúčové body v obraze zhodné s rovnakými 3D bodmi na CAD modely vozidla.

Kľúčové body sa vyhľadávajú len na malej skupine vozidiel. Tieto vozidlá sú detekované pomocou neurónovej siete, ktorá vyhľadáva vozidlá konkrétnej značky, modelu, roku výroby. Jedná sa o deväť typov vozidiel značky Škoda, ktoré majú v datase *BrnoCompSpeed* [13] početné zastúpenie. Pre tieto najčastejšie typy vozidiel sa lokalizuje poloha 20 kľúčových bodov pomocou ďalšej neurónovej siete. Niektoré z týchto bodov majú nejednoznačnú pozíciu v 3D modely (napríklad predné svetlá), preto sa využíva iba 12 kľúčových bodov (4 kolesá, 2 ŠPZ, 2 značka, 4 rohy na streche vozidla). Body ktoré nie sú na popredí modelu sa ďalej nepoužívajú. Pre všetky typy vozidiel je zostrojený ich odpovedajúci 3D model z ktorého je možné vypočítať reálnu vzdialenosť medzi ľubovoľnými dvoma kľúčovými bodmi. [2].

Z datasetu *BCS_keypoints_2D_3D* sa určí merítka scény ako podiel medzi reálnou vzdialenosťou a vzdialenosťou v obraze medzi dvoma kľúčovými bodmi priradenými ku kolesám vozidla. Tieto body sú najbližšie k rovine vozovky a pri umiestnení kamery vo výške nad rovinou vozovky predstavujú najmenšie skreslenie na plochu v obraze. Po vypočítaní mierky pre všetky dvojice v datasete sa výsledná mierka určí ako aritmetický priemer týchto hodnôt. Tým získame všetky informácie potrebné k tomu aby sme mohli merať rýchlosti vozidiel v absolútnych jednotkách.

4.5 Výpočet rýchlosti

Po získaní kalibračných informácií sa sledujú vozidlá pomocou metódy popísanej v podkapitole 4.2. Pre každé nové vozidlo sa vytvorí objekt, ktorý obsahuje navyše informácie o čísle snímku n v ktorom sa vyskytlo v scéne a informácie o ohraničujúcom boxe vozidla. Bod v strede spodnej hrany boxu vozidla je práve použitý ako referenčný bod p pre sledovanie prejdetej vzdialenosti. Pomocou rovnice

$$v = \frac{\lambda \|P_e - P_s\|}{t_e - t_s}. \quad (4.17)$$

vieme vypočítať prejetú vzdialenosť bodu p premietnutú do roviny vozovky medzi počiatočným n_s a koncovým n_e snímkom. Táto vzdialenosť je ale v abstraktných jednotkách, preto ju musíme vynásobiť získanou mierkou scény λ .

Pre vypočítanie rýchlosti musíme zistiť čas za ktorý vozidlo vzdialenosť prešlo. Poznáme rozpetie výskytu $n_e - n_s$ v scéne. Aby sme ho prepočítali na čas potrebujeme vedieť počet snímok za sekundu. Tento údaj sa dá získať priamo z knižnice OpenCV pomocou funkcie `.get(cv2.cv.CV_CAP_PROP_FPS)`, ale výsledok nie je vždy spoľahlivý, preto je ho lepšie nastaviť manuálne. Tým poznáme všetky hodnoty pre jej výpočet. Samotné meranie je založené na detekcii prvého a posledného výskytu sledovaného bodu p . Ukážka grafického výstupu programu je na obrázku 4.8.



Obr. 4.8: Grafický výstup zobrazujúci ID a rýchlosť vozidiel.

Kapitola 5

Implementácia a testovanie

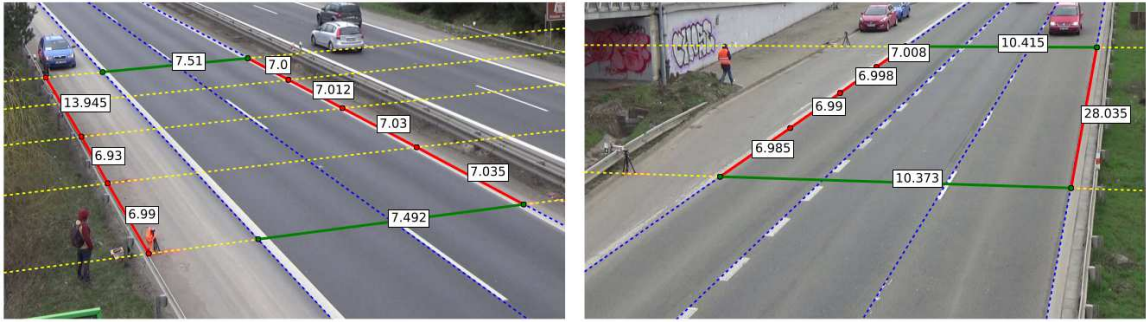
Celý systém je implementovaná v jazyku Python3, kvôli dobrej podpore funkcií a knižníc. Jedná sa o interpretovaný jazky takže nie je problém s prenosom a spúšťaním kódu na rôznych platformách. Program beží nad virtuálnym prostredím, ktoré oddeľuje prostredie pre Python, v ktorom sú nainštalované potrebné knižnice. Okrem štandardných knižníc sú použité opencource knižnice OpenCV a NumPy, ktoré poskytujú základné funkcie pre spracovanie videa popísané v kapitole 3.

Program pozostáva zo štyroch súborov: `info.py`, `calibration.py`, `vehicles.py` a `speed.py`. V súbore `info.py` sú informácie ktoré musí užívateľ definovať pred spustením ako napríklad cesta k videu alebo FPS videa. Potom stačí spustiť kód `speed.py`, ktorý najskôr získa kalibračné informácie kamery pomocou funkcií implementovaných v súbore `calibration.py` a následne sa vykonáva detekcia a meranie rýchlosti prechádzajúcich vozidiel, kde pre každé vozidlo sa vytvorí nový objekt podľa triedy definovanej v súbore `vehicles.py`.

Výstupom systému sú obrazové data, ktoré sú doplnené informáciami o číse vozidla a jeho aktuálnej rýchlosti. Okrem toho sa generuje JSON súbor s nasledujúcou štruktúrou:

```
{
  "camera_calibration":
    {
      "vp1": [x,y],
      "vp2": [x,y],
      "pp": [x,y],
      "scale": lambda
    },
  "cars":
    [{
      "id": number,
      "frames": [numbers],
      "posX": [numbers],
      "posY": [numbers]
    }]
},
```

ktorý obsahuje kalibračné informácie (prvý a druhý úbežník, hlavný bod a mierka scény) a informácie o polohe vozidla v danom čase. Štruktúra výstupného súboru je zhodná s da-



Obr. 5.1: Vyznačené a namerané vzdialenosti na rovne vozovky. Červené čiary smerujú k prvému úbežníku a zelené k druhému. (Obr. prevzatý z [13]).

tasetom *BrnoCompSpeed* [13] kvôli vyhodnoteniu získaných výsledkov. Výsledky testov sú prezentované v nasledujúcich podkapitolách.

5.1 Dataset

Implementácia systému bola vyvíjaná a otestovaná na základe dat obsiahnutých v datase *BrnoCompSpeed* [13]. Tento dataset zhromažďuje videá zo šiestich rôznych lokalít. Každá lokalita je snímaná z troch rôznych pozícií umiestnenia kamery, ktorá zachytáva pohľad z lavej, prostrednej a pravej strany. To dáva dokopy 18 videí, pričom dĺžka každého videa je približne jedna hodina. Videá sú natočené vo full-HD rozlíšení so snímkovou frekvenciou 50 fps na kamerách Panasonic HC-X920, Panasonic HDC-SD90, Sony Handycam HDR-PJ410.

V týchto videách sa vyskytlo dokopy 20865 vozidiel a okrem videozáznamov sú v datase presné informácie o rýchlostiach prechádzajúcich vozidiel. K získaniu týchto referenčných rýchlostí boli použité optické brány s technológiou LIDAR, ktoré boli umiestnené pri ceste kolmo na smer vozovky so vzájomnou vzdialenosťou približne 28 metrov od seba. Tie merali časový interval vstupu a výstupu automobilu medzi prvým a druhým laserovým lúčom. Aby bolo meranie presné, ich časy boli synchronizované pomocou GPS.

V pozorovaných úsekoch sú na rovne vozovky pomocné značky, medzi ktorými sú namerané vzdialenosti ako je možné vydiť na obrázku 5.1. To napomáha overeniu výsledkov z automatickej kalibrácie, prípadne určenia merítka scény vzhľadom k známej vzdialenosti. Navyše dataset obsahuje výsledné informácie niektorých systémov, ktoré boli pustené nad týmito videami a skript, ktorý porovná presnosť systému vzhľadom k reálnym dátam, čo zjednodušuje overenie kvality systému.

5.2 Vyhodnotenie detekcie úbežníkov

Najväčší vplyv pre správne fungovanie systému majú vonkajšie parametre kamery a to hlavne pozície úbežníkov. Od nich závisí hodnota ohniskovej vzdialenosti, ktorá sa využíva pri prepočte polohy bodu v obraze na svetovú polohu v rovne vozovky. Správne odhadnutie úbežníkov má teda významný vplyv na presnosť celého systému.

Vyhodnotenie presnosti úbežníkov zabezpečuje funkcia `evalPureCalibration()` ktorá je súčasťou testovacieho skriptu. Tá pracuje so súradnicami bodov v obraze pre ktoré sú známe odpovedajúce svetové pozície. Pre tie isté obrazové body sa na základe kalibračných informácií kamery spočítajú polohy bodov v rovne vozovky. K vyhodnoteniu sa teda pou-

žíva pomer medzi skutočnou polohou a premietnutou polohou bodu. Pri tomto teste nie je braná do úvahy mierka scény. Výsledky reprezentujúce odchylky detekovaných úbežníkov je možné vidieť v tabuľke 5.1.

Video	absolútna chyba			relatívna chyba [%]		
	priemer	median	95 percentil	priemer	median	95 percentil
s1 left	0.31	0.36	0.55	9.66	10.34	15.51
s1 center	0.02	0.01	0.04	0.62	0.60	1.07
s1 right	0.03	0.01	0.06	1.06	1.20	1.69
s2 left	0.05	0.04	0.10	4.43	4.01	8.65
s2 center	0.17	0.03	0.81	12.63	3.98	26.23
s2 right	0.09	0.02	0.44	6.44	2.35	13.49
s3 left	0.02	0.01	0.06	2.50	1.17	6.26
s3 center	0.55	0.08	1.58	58.56	8.03	170.90
s3 right	0.03	0.02	0.06	2.88	2.48	6.61
s4 left	0.08	0.04	0.20	7.24	2.92	17.19
s4 center	0.16	0.05	0.41	14.43	3.50	35.48
s4 right	0.03	0.02	0.05	2.21	1.79	5.69
s5 left	0.16	0.02	0.71	10.45	5.42	23.26
s5 center	0.22	0.03	1.03	13.08	3.05	31.03
s5 right	0.24	0.07	1.07	14.51	8.26	34.62
s6 left	0.13	0.03	0.56	10.57	3.25	23.92
s6 center	0.20	0.09	0.84	16.82	10.07	35.32
s6 right	0.05	0.03	0.22	4.31	3.56	9.01
Celkom	0.15	0.03	0.84	12.71	3.51	34.63

Tabuľka 5.1: Absolútne a relatívne chyby odchýlok medzi reálnymi a projektovanými bodmi z obrazu. Chybná projekcia bodov je spôsobená zlým odhadom polohy úbežníkov

Z výsledkov vyplýva, že najhoršie dopadla detekcia úbežníkov pre centrické umiestnenie kamery nad vozovkou. Pozorovaný úsek pre stredovú kameru je najkratší, čím dochádza k malej zmene natočenia hrany. Pri porovnaní získaných hodnôt úbežníkov s hodnotami v datasete je zrejmé, že najväčší problém predstavuje práve detekcia druhého úbežníka. Jeho poloha sa najviac vzdialuje od odhadnutej polohy iným systémom, čo ovplyvňuje hodnotu ohniskovej vzdialenosti a negatívne pôsobí na ďalšie etapy systému. To je spôsobené pravdepodobne tým, že do *DiamondSpace* priestoru sú akumulované aj hrany ktoré nesmerujú len k druhému úbežníku. Tieto hrany nevieme účinne filtrovať kvôli meniacej sa polohy kamery.

K zlepšeniu výsledku by pomohlo zmenšiť rozsah uhlu horizontu alebo presnejší odhad rozsahu ohniskovej vzdialenosti. Aby bol ale systém transparentný, tieto hodnoty sa nedajú nastaviť optimálne pre globálne nasadenie. K zvýšeniu presnosti by pomohla aj lepšia funkcia pre hľadanie maxima v *DS* priestore. Aktuálna vracia pozíciu prvého výskytu najväčšej hodnoty. S použitím optimalizačnej metódy, ktorá by brala do úvahy okolie maxima by mohlo dôjsť k zlepšeniu výsledku.

5.3 Vyhodnotenie merania vzdialenosti

Ďalším hodnotením presnosti systému je meranie vzdialenosti, kde sa využíva získané merítko scény pre prevod do štandardnej vzdialenosti. Vyhodnotenie pozostáva z dvoch častí. V prvom prípade funkcia `evalCamCalibWithScale()` vyhodnocuje vzdialenosť bodov smerujúcich len k prvému úbežníku a v druhom prípade vyhodnocuje vzdialenosť všetkých bodov. Podobne ako pri vyhodnotení presnosti úbežníkov aj tu sa využívajú body v obraze ku ktorým sú známe svetové súradnice. Pre obrazové body sa vypočíta ich projekcia do roviny s použitím kalibračných údajov. Následne sa porovná vzdialenosť dvoch bodov medzi svetovými súradnicami so vzdialenosťou ich premietnutých pozícií v rovine vynásobené mierkou scény. Najdôležitejšie sú výsledky pre meranie vzdialenosti práve v smere k prvému úbežníku, keďže týmto smerom sa pohybujú automobily. Výsledky presnosti v meraní vzdialenosti sú uvedené v tabuľke 5.2.

Video	všetko		smer k 1VP	
	priemer [m]	median [m]	priemer [m]	median [m]
s1 left	1.01	1.09	0.78	0.78
s1 center	1.68	1.64	2.64	2.64
s1 right	0.18	0.12	0.31	0.31
s2 center	3.54	2.75	4.34	2.76
s2 right	0.56	0.33	0.48	0.26
s3 left	0.12	0.05	0.04	0.03
s3 center	4.07	3.62	3.56	3.57
s3 right	1.33	1.36	1.37	1.38
s4 left	0.39	0.13	0.12	0.09
s4 center	1.76	1.62	1.94	1.66
s4 right	0.12	0.07	0.07	0.06
s5 left	0.95	0.78	0.98	0.67
s5 center	1.69	1.59	1.45	0.81
s5 right	0.82	0.54	0.45	0.35
s6 left	0.59	0.31	0.19	0.17
s6 center	1.91	1.76	1.49	1.08
s6 right	0.59	0.30	0.70	0.37
Celkom	1.23	0.74	1.17	0.61

Tabuľka 5.2: Priemerné a mediánové odchyľky v meraní všetkých vzdialenosti na rovine vozovky a v meraní vzdialeností smerujúcich iba k prvému úbežníku.

Z výsledkov je jasné, že systém nedosahuje najpresnejšie meranie vzdialenosti čo sa premietne aj do merania rýchlosti. Odchyľka merania vzdialenosti je priamoúmerná s chybou určenia úbežníkov a merítka scény. Merítko sa počítalo zo vzdialenosti bodov nadetkových na stred kolies, ktoré neležia na rovine vozovky, čo vnáša ďalšie nepresnosti do celého systému. Najhoršie dopadli merania opäť pre kamery umiestnené nad stredom vozovky. Priemerné meranie vzdialenosti v smere k prvému úbežníku vykazuje lepšie hodnoty ako merania vo všetkých smeroch, čo potvrdzuje práve nepresnosť detekcie druhého úbežníka.

5.4 Vyhodnotenie merania rýchlosti

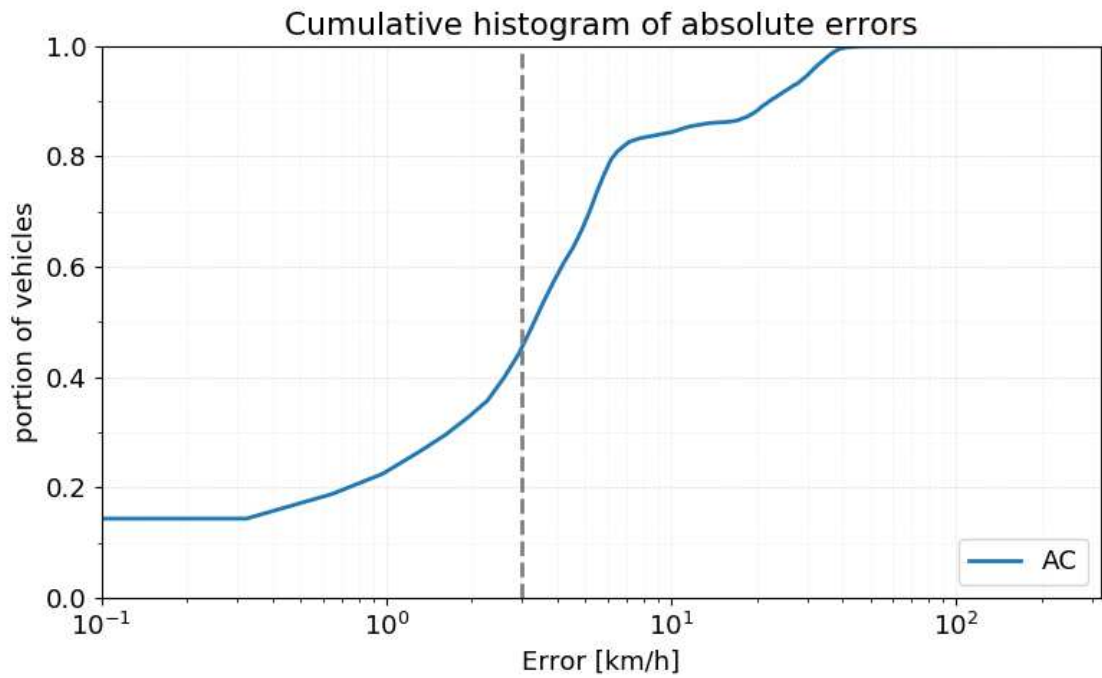
Posledný test predstavuje samotné meranie rýchlosti. To je závislé okrem kalibračných informácií aj na presnosti detekcie vozidiel. Pre každé vozidlo sú uložené informácie o snímkoch, v ktorých sa vyskytlo spolu s jeho odpovedajúcou polohou v obraze. Tieto informácie reprezentujú vzdialenosť, ktorú vozidlo prešlo za určitý čas, z čoho sa vypočíta rýchlosť. Táto rýchlosť je porovnaná s reálnou rýchlosťou vozidiel, ktorá bola nameraná pomocou optických brán priamo v teréne.

Práve detekcia a sledovanie vozidiel v tomto systéme nevykazuje najlepšie výsledky. Najväčší problém predstavuje práve detekcia vozidiel, ktorá je závislá na maske popredia. V maske sa vyhľadávajú kontúry, ktoré sú prehlásené za vozidlá. Problém nastáva ak sa vozidlá prekrývajú. Vtedy to systém vyhodnotí ako jedno vozidlo. Opačný problém je ak na jednom vozidle je detekovaných viacej kontúr, čím sa pre jeden skutočný objekt vytvorí viacero záznamov. Problém robí aj malý pohyb kamery spôsobený otrasmi alebo príliš rýchle jasové zmeny, ktoré do masky zavádzajú ďalší šum. Tieto nedostatky sa v malej miere dajú filtrovať na základe veľkosti kontúr, avšak všetky problémy to nevyrieši.

Video	absolútna chyba [km/h]			relatívna chyba [%]		
	priemer	median	95 percentil	priemer	median	95 percentil
s1 left	2.40	1.70	4.52	3.00	2.26	4.63
s1 center	7.05	6.25	8.85	9.19	8.28	10.16
s1 right	0.70	0.34	1.59	0.91	0.47	2.05
s2 left	3.48	3.35	5.02	4.05	3.98	5.66
s2 center	32.87	32.57	39.69	38.46	38.52	38.95
s2 right	3.19	3.11	4.05	3.69	3.65	4.23
s3 left	0.39	0.32	1.00	0.65	0.56	1.53
s3 center	30.24	29.94	38.37	50.75	50.69	51.75
s3 right	11.06	10.89	13.99	18.47	18.47	19.51
s4 left	0.84	0.59	2.07	0.81	0.65	1.82
s4 center	21.70	21.55	26.77	22.70	22.98	23.67
s4 right	0.67	0.41	1.61	0.68	0.42	1.48
s5 left	5.32	5.35	6.77	6.91	7.35	8.30
s5 center	5.76	5.76	7.33	7.47	7.59	8.85
s5 right	1.69	0.95	5.10	2.17	1.29	6.50
s6 left	1.90	1.90	2.94	2.65	4.26	15.65
s6 center	5.37	4.04	12.50	7.68	5.58	18.11
s6 right	3.94	3.91	5.25	5.55	5.41	7.48
Celkom	6.71	3.66	30.75	8.28	4.55	38.47

Tabuľka 5.3: Absolútne a relatívne chyby odchýlok medzi odhadnutou rýchlosťou vozidiel a ich skutočnou rýchlosťou.

Kvôli tomu som pri vyhodnotení rýchlosti použil informácie o polohe vozidiel, ktoré sú dostupné vo výsledkoch datasetu. Tabuľka 5.3 teda obsahuje výsledky merania rýchlosti z informácií o kalibračných parametroch kamery získanými od navrhovaného systému, a informácií o vozidlách zverejnených v datasete. Dáta opäť ukazujú najhoršie presnosti merania pre kamery umiestnené nad stredom vozovky, kde došlo k najhoršiemu odhadu po-



Obr. 5.2: Kumulatívny histogram absolútnych chýb pre implementovaný systém.

lohy druhého úbežníka. Kumulatívny histogram absolútnych chýb je na obrázku 5.2. Z neho vyplýva, že rýchlosť pre zhruba 42% vozidiel je odhadnutá s chybou menšou ako $3 \frac{km}{h}$.

Kapitola 6

Záver

Bakalárska práca sa zaoberá získaním kalibračných údajov zo záznamu z dohľadovej kamery pre následné určenie rýchlosti prechádzajúcich vozidiel. Pred návrhom systému boli preštudované existujúce systémy, ktoré riešia podobnú problematiku spolu s algoritmi, ktoré sú využité pri implementácii systému.

Hlavným cieľom bolo získanie kalibračných údajov plne automaticky zo záznamu videa, bez použitia vstupu od užívateľa pred spustením systému. To je docielené vďaka detekcie úbežníkov v pozorovanej scéne, z ktorých sú dopočítané zvyšné kalibračné parametre.

Celý systém je implementovaný v jazyku Python s využitím knižníc OpenCV a NumPy pre spracovanie videa. Systém implementuje tri kľúčové časti pre kalibráciu kamery, detekciu vozidiel a meranie rýchlosti vozidiel. Pre meranie rýchlosti v známych jednotkách (km/h) je použitý dataset *BCS_keypoints_2D_3D*, ktorý obsahuje nadetegované kľúčové body na vybraných typoch vozidiel, kú ktorým je známa ich odpovedajúca 3D pozícia. S pomocou týchto informácií sa dopočíta merítka scény.

Testovanie úspešnosti implementovaného systému bolo realizované na datasete *Brno-CompSpeed*. Testy okrem merania rýchlosti boli zamerané aj na presnosť odhadu úbežníkov a meranie vzdialenosti medzi známou svetovou polohou bodov v scéne a polohou projekovaných bodov do roviny vozovky. Meranie vzdialeností medzi známymi bodmi v obraze vykazuje priemernú chybu 1.23m, ktorú väčšinou spôsobuje chybné určenie polohy druhého úbežníka. To sa prejavuje aj pri samotnom meraní rýchlosti. Priemerná absolútna chyba rýchlosti vozidiel dosahuje hodnoty 6.71km/h. Rýchlosť s presnosťou do 3km/h bola dosiahnutá približne na 40% meraných vozidlách. Systém teda nedosahuje uspokojivé výsledky, aby nahradil kontrolu rýchlosti radarovými systémami, ale výsledky sú dostačujúce napríklad pre zber štatistických dát.

V budúcnosti by bolo vhodné zvýšiť presnosť detekcie hlavne druhého úbežníka, ktorého poloha bola určená s najmenšou presnosťou vzhľadom k jeho reálnej polohe. Problém predstavuje aj samotná detekcia vozidiel, ktorá je založená na maske popredia. V situáciách, kde sa vozidlá prekrývajú sú tieto vozidlá identifikované ako jeden objekt. Pri ďalšom vývoji by bolo vhodné na ich detekciu použiť lepšiu metódu alebo neurónovú sieť. Najdôležitejším by ale bola implementácia vlastného podsystému pre získanie merítka scény. Pri spustení systému nad iným záznamom ako z datasetu prebieha meranie len v relatívnych jednotkách, ktoré neodpovedajú známym jednotkám.

Literatúra

- [1] and; and: Research on Lane-Marking Line Based Camera Calibration. In *2007 IEEE International Conference on Vehicular Electronics and Safety*, Dec 2007, s. 1–6, doi:10.1109/ICVES.2007.4456361.
- [2] Bartl, V.; Špaňhel, J.; Dobeš, P.; aj.: Automatic Camera Calibration by Landmarks on Rigid Objects. *Computer Vision and Image Understanding*, ISSN 1077-3142.
- [3] Beymer, D.; McLauchlan, P.; Coifman, B.; aj.: A real-time computer vision system for measuring traffic parameters. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1997, ISSN 1063-6919, s. 495–501, doi:10.1109/CVPR.1997.609371.
- [4] Dubska, M.; Herout, A.; Juranek, R.; aj.: Fully Automatic Roadside Camera Calibration for Traffic Surveillance. *Intelligent Transportation Systems, IEEE Transactions on*, ročník 16, č. 3, June 2015: s. 1162–1171, ISSN 1524-9050, doi:10.1109/TITS.2014.2352854.
- [5] Dubska, M.; Herout, A.; Sochor, J.: Automatic Camera Calibration for Traffic Understanding. In *Proceedings of the British Machine Vision Conference*, BMVA Press, 2014, doi:http://dx.doi.org/10.5244/C.28.42.
- [6] Harris, C.; Stephens, M.: A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 1988, s. 147–151.
- [7] Kováč, O.: Teoretická analýza geometrickej kalibrácie kamery. 2014-07-30.
URL <http://www.posterus.sk/?p=17190>
- [8] Luvizon, D. C.; Nassu, B. T.; Minetto, R.: A Video-Based System for Vehicle Speed Measurement in Urban Roadways. *IEEE Transactions on Intelligent Transportation Systems*, ročník 18, č. 6, June 2017: s. 1393–1404, ISSN 1524-9050, doi:10.1109/TITS.2016.2606369.
- [9] Patel, D.; Upadhyay, S.: Optical Flow Measurement using Lucas Kanade Method. *International Journal of Computer Applications*, ročník 61, 01 2013: s. 6–10, doi:10.5120/9962-4611.
- [10] Schoepflin, T. N.; Dailey, D. J.: Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *IEEE Transactions on Intelligent Transportation Systems*, ročník 4, č. 2, June 2003: s. 90–98, ISSN 1524-9050, doi:10.1109/TITS.2003.821213.

- [11] Shi, J.; Tomasi, C.: Good features to track. In *IEEE CVPR*, 1994, ISBN 0-8186-5825-8, ISSN 1063-6919, s. 593–600, doi:10.1109/CVPR.1994.323794. URL <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=323794>
- [12] Sochor, J.; Juranek, R.; Herout, A.: Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*, ročník 161, 2017: s. 87 – 98, ISSN 1077-3142, doi:<http://dx.doi.org/10.1016/j.cviu.2017.05.015>. URL <http://www.sciencedirect.com/science/article/pii/S1077314217301108>
- [13] Sochor, J.; Juránek, R.; Špaňhel, J.; aj.: Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement. *IEEE Transactions on Intelligent Transportation Systems*, 2018: s. 1–11, ISSN 1524-9050, doi:10.1109/TITS.2018.2825609.
- [14] Stauffer, C.; Grimson, W. E. L.: Adaptive Background Mixture Models for Real-Time Tracking. In *CVPR*, IEEE Computer Society, 1999, ISBN 0-7695-0149-4, s. 2246–2252. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr1999.html#StaufferG99>
- [15] opencv dev team: Harris Corner Detection. 2014. URL https://docs.opencv.org/3.0-alpha/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html
- [16] Markéta Dubská (Brno University of Technology), A. H. B. U. o. T.: Real Projective Plane Mapping for Detection of Orthogonal Vanishing Points. In *Proceedings of the British Machine Vision Conference*, BMVA Press, 2013.
- [17] Thi, T. H.; Lu, S.; Zhang, J.: Self-Calibration of Traffic Surveillance Camera using Motion Tracking. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, Oct 2008, ISSN 2153-0009, s. 304–309, doi:10.1109/ITSC.2008.4732673.
- [18] Welch, G.; Bishop, G.: An Introduction to the Kalman Filter. 1995.