



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

**ZAŘÍZENÍ VAROVNÉHO SYSTÉMU PRO UDRŽENÍ
VOZIDLA V JÍZDNÍM PRUHU**

WARNING SYSTEM TO KEEP THE VEHICLE IN THE LANE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Vítězslav Fendrich

VEDOUCÍ PRÁCE

SUPERVISOR

doc. RNDr. Jitka Poměnková, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Bc. Vítězslav Fendrich

ID: 164201

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Zařízení varovného systému pro udržení vozidla v jízdním pruhu

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami a algoritmy pro detekci jízdních pruhů na komunikaci z obrazového signálu a proveďte rešerši možných přístupů. Navrhněte a implementujte vhodný algoritmus pro detekci vybočení z jízdního pruhu. Navrhněte a sestavte vhodný hardware, umístitelný na čelní sklo vozidla, na kterém bude možné algoritmus pro detekci vybočení z jízdního pruhu implementovat. Navrhněte rozsah, vlastnosti a parametry databáze pro nastavení resp. trénování algoritmu a následné testování.

Implementujte algoritmus na navržený hardware, celé zařízení oživte a zprovozněte. Proveďte testování zařízení na pořízených videosekvencích, ve vozidle. Výsledky statisticky zhodnoťte a vyvoďte závěry a doporučení pro používání realizovaného zařízení pro detekci a varování vybočení z jízdního pruhu.

DOPORUČENÁ LITERATURA:

[1] THEODORIDIS, S., KOUTROUMBAS, K. Pattern Recognition, Fourth Edition, Academic Press, 2009, 961 s., ISBN: 9781597492720.

[2] WESTIN, L. K. Receiver operating characteristic (ROC) analysis. Evaluating discriminance effects among decision support systems. 28 pages. [online] Cited 2014-02-11. Available at: <http://nutkin.cs.umu.se/research/reports/2001/018/part1.pdf>

Termín zadání: 4.2.2019

Termín odevzdání: 16.5.2019

Vedoucí práce: doc. RNDr. Jitka Poměnková, Ph.D.

Konzultant: Ing. Tobiáš Malach

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Tato práce se zabývá návrhem zařízení, které detekuje vybočení vozidla z jízdního pruhu, a to prostřednictvím obrazového signálu kamery. Zařízení je navrženo tak, aby bylo umístitelné na čelní sklo vozidla. Úvodní část práce je věnována zmapování způsobů detekce jízdních pruhů z obrazového signálu. Následně je proveden výběr vhodného zařízení, konkrétně nejnovějšího modelu Raspberry Pi včetně kamerového modulu. Pro zařízení byla navržena a následně prostřednictvím technologie 3D tisku vytvořena vhodná krabička. V další části práce je vybrán, navržen a implementován vhodný algoritmus a dále zvolen rozsah a parametry testovací databáze. Závěrečná část práce obsahuje vyhodnocení úspěšnosti detekce, prostřednictvím pořízené testovací databáze.

Klíčová slova

Cannyho detektor, detekce jízdních pruhů, Houghova transformace, LDWS, Raspberry Pi

Abstract

This thesis addresses designing a device that detects lane departure of a vehicle via a video feed from a camera module. This device is intended to be attached onto the windshield of the vehicle. The initial part of the thesis will cover the current methods of lane departure detection through a video feed. In the following part the selection of suitable hardware, specifically the latest model of a Raspberry Pi, has been made. Afterwards a suitable container for the aforementioned hardware has been designed and created using a 3D printer. Subsequently an appropriate LDWS algorithm is chosen and designed. In the next part, the range and parameters of a testing database through which the proper functionality of the device will be tested on are chosen. The final part of the thesis contains evaluation of the success rate of detection via the acquired database.

Keywords

Canny edge detector, lane detection, Hough transform, LDWS, Raspberry Pi

Bibliografická citace:

FENDRICH, Vítězslav. *Zařízení varovného systému pro udržení vozidla v jízdním pruhu*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Jitka Poměnková.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma *Zařízení varovného systému pro udržení vozidla v jízdním pruhu*, jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **16. května 2019**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Tobiášovi Malachovi a také doc. RNDr. Jitce Poměnkové, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **16. května 2019**

.....
podpis autora

Obsah

1.	Úvod.....	6
2.	Detekce jízdních pruhů	7
2.1	Inverzní perspektivní mapování (IPM)	7
2.2	Detekce hran.....	8
2.2.1	Filtrace obrazu	9
2.2.2	Hranové detektory.....	12
2.2.3	Houghova transformace	15
2.2.4	Rozdělení obrazu do oblastí zájmu	16
2.2.5	Detekce vybočení vozidla	16
2.3	Zpracování snímané scény před vozidlem	18
2.3.1	Vodorovné dopravní značení na pozemních komunikacích	18
2.3.2	Aplikace vybraných algoritmů na obrazový signál	22
3.	HARDWARE.....	24
3.1	Raspberry Pi	24
3.1.1	GPIO	25
3.2	Kamera	27
3.3	Napájení	28
3.4	Akustická signalizace	28
3.5	Návrh boxu pro zařízení.....	29
4.	SOFTWARE	31
4.1	Operační systém Raspberry Pi	31
4.2	Vzdálený přístup	32
4.3	Kamera Raspberry Pi	33
4.4	Knihovna OpenCV	34
4.5	Vývojové prostředí.....	36
4.6	Test knihovny OpenCV	36
4.7	Program Recorder.py	37
5.	LDWS algoritmus	39
5.1	Popis fungování algoritmu	39
6.	Vyhodnocení spolehlivosti algoritmu	49
6.1	Testovací databáze	49

6.2	Vyhodnocení spolehlivosti algoritmu	50
6.3	Výpočet intervalu spolehlivosti.....	52
6.4	Doporučení pro použití zařízení.....	53
7.	Závěr	54

Seznam symbolů a zkratek

Zkratky:

CSI	...	Camera Serial Interface
CMOS	...	Complementary Metal-Oxide-Semiconductor
DSI	...	Display Serial Interface
FPS	...	Frames per second, počet snímků za sekundu
GB	...	Gigabyte
GPIO	...	General Purpose Input/Output
IoT	...	Internet of Things, internet věcí
IP	...	Internet Protocol
IPM	...	Inverzní perspektivní mapování
LDWS	...	Lane departure warning system
LoG	...	Laplacian of Gaussian
MB	...	Megabyte
ROI	...	Region of Interest, oblast zájmu
SPI	...	Serial Peripheral Interface
STP	...	Shielded Twisted Pair
UART	...	Universal Asynchronous Receiver and Transmitter
USB	...	Universal Serial Bus
UTP	...	Unshielded Twisted Pair

Seznam obrázků

Obrázek 2-1 Perspektiva IPM (převzato z[2])	7
Obrázek 2-2 Model IPM zobrazení (převzato z[3])	8
Obrázek 2-3 Náhled scény před vozidlem	18
Obrázek 2-4 Příklad značení na vozovce o šířce menší než 6 m (na základě [16])..	19
Obrázek 2-5 Příklad značení na vozovce o šířce 6 až 7 m (na základě [16])	19
Obrázek 2-6 Příklad značení na vozovce širší než 7 m (na základě [16])	20
Obrázek 2-7 Příklad značení směrově rozdělené pozemní komunikace (na základě [16])	21
Obrázek 2-8 Zpracování obrazu vybranými algoritmy	22
Obrázek 3-1 Raspberry Pi 3 Model B+	25
Obrázek 3-2 Kamerový modul v2 pro Raspberry Pi	27
Obrázek 3-3 3D model krabičky pro zařízení	30
Obrázek 4-1 Konfigurační menu Raspberry Pi 3	32
Obrázek 4-2 Uživatelské rozhraní programu Recorder.py	37
Obrázek 5-1 Ukázka ROI dělící čáry	41
Obrázek 5-2 Ukázka ROI vodící čáry	41
Obrázek 5-3 Binarizovaný obraz pro detekci hran	42
Obrázek 5-4 Detekované hrany z binarizovaného snímku	43
Obrázek 5-5 Detekované hrany z obrazu ve stupních šedi	43
Obrázek 5-6 Vykreslení detekovaných čar do původního snímku	44
Obrázek 5-7 Princip detekce vybočení vozidla	45
Obrázek 5-8 Detekce vozidla v pruhu	46
Obrázek 5-9 Ukázka detekce vybočení vlevo	46
Obrázek 5-10 Ukázka detekce vybočení vpravo	47
Obrázek 5-11 Blokový diagram navrženého algoritmu	48
Obrázek 6-1 Ukázka chybné detekce šrafování	51
Obrázek 6-2 Ukázka chybné detekce připojovacího pruhu	52
Obrázek 6-3 Ukázka falešně pozitivní detekce vybočení v zatáčce	52

Seznam tabulek

Tabulka 3-1 Parametry Raspberry Pi 3 B+	25
Tabulka 3-2 Popis funkce jednotlivých pinů GPIO sběrnice.....	26
Tabulka 3-3 Parametry snímače Sony IMX219PQ	27
Tabulka 5-1 Soupis funkcí knihovny <i>Processing</i>	40
Tabulka 6-1 Návrh parametrů testovací databáze.....	50
Tabulka 6-2 Úspěšnost detekce pro jednotlivé třídy silnice	50
Tabulka 6-3 Vypočtené intervaly spolehlivosti	53

1. ÚVOD

Pokročilé asistenční systémy neustále monitorující situaci okolo vozidla již v dnešní době nejsou výsadou pouze prémiových vozidel. Nicméně vzhledem k tomu, že se stále jedná o novou a neustále se zdokonalující technologii, je zpravidla vyhrazena pouze nejnovějším modelům jednotlivých automobilek. Dle statistických údajů registru vozidel České republiky je však aktuální průměrné stáří vozidel rovno 15,4 roku. Jedná se o poměrně vysoké číslo, ze kterého je zřejmé, že moderními asistenčními systémy, které výrobci masivněji nasazují do vozidel až v posledních třech letech, disponuje jen zlomek vozidel pohybujících se po českých silnicích. Z hloubkové analýzy dopravních nehod v Jihomoravském kraji vyplývá, že téměř 50 % dopravních nehod je způsobeno nepozorností řidiče. Na dálnicích a rychlostních silnicích bývají dopravní nehody často závažné díky vysoké rychlosti pohybujících se vozidel.

Cílem této práce je návrh zařízení, které bude možné dodatečně umístit do vozidla na čelní sklo a které bude hlídat, zda se vozidlo nachází v jízdním pruhu. V případě, že z něj vozidlo začne vybočovat, například vlivem nepozornosti řidiče, bude zařízení akusticky varovat řidiče před hrozícím nebezpečím. Vzhledem k neustále narůstajícímu výpočetnímu výkonu mikropočítačů je možné zvolit robustní algoritmus, který bude poskytovat stabilní výsledky detekce jízdních pruhů. Mikropočítače se v současné době těší stále vyšší popularitě, mimo jiné díky stoupající oblibě IoT aplikací. Díky tomu bude takové zařízení také cenově dostupné.

První část práce bude věnována zmapování a následnému teoretickému rozboru aktuálně používaných technik detekce jízdního pruhu z obrazového signálu kamery. Následně bude vytvořen vhodný algoritmus, který bude dostatečně spolehlivý a také vhodný z hlediska rychlosti zpracování získaného obrazu s ohledem na dostupný výpočetní výkon zařízení. Další část bude věnována výběru vhodného zařízení pro danou aplikaci, které bude vyhovovat z hlediska výpočetního výkonu, rozměrů a možnosti připojení obrazového snímače. Celé zařízení musí být koncipováno tak, aby tvořilo jeden celek, který bude možné snadno umístit na čelní sklo vozidla, aby nerušilo výhled a neodporovalo tak platné legislativě. V experimentální části práce bude navržené zařízení sestaveno a oživeno spolu s navrženým algoritmem, jehož funkcionality bude otestována prostřednictvím databáze testovacích videosekvencí. Tato databáze bude navržena a pořízena tak, aby postihla spektrum různých kvalitativních parametrů. V závěru práce bude provedeno vyhodnocení výsledků testování zařízení.

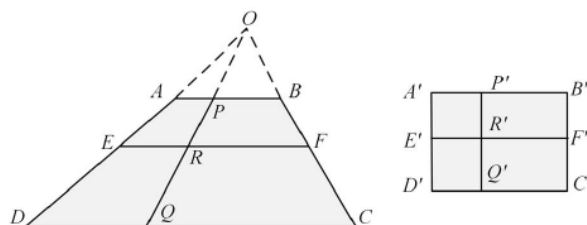
2. DETEKCE JÍZDNÍCH PRUHŮ

Systémy, jež jsou schopny detekovat přítomnost jízdních pruhů na vozovce a současně řidiče varovat při jejich neúmyslném překročení, označujeme souhrnně jako Lane Departure Warning System (dále jen LDWS). Jedná se o asistenční systém, který prostřednictvím obrazového signálu kamery, umístěvané nejčastěji do středu horní části čelního skla, hlídá, zda nedochází k neúmyslnému vybočení vozidla z jízdního pruhu, nejčastěji vlivem nepozornosti řidiče. Při detekci takového vybočení pak systém řidiče varuje vizuálně, případně akusticky. Ze způsobu detekce jízdních pruhů je zřejmé, že systém LDWS ke své správné funkci potřebuje souvislé vodorovné značení na vozovce a také odpovídající vizuální podmínky – řádně očištěné čelní sklo a osvětlenou vozovku před vozidlem v případě snížené viditelnosti. Přesto je nutné počítat s omezenou funkčností systému LDWS při velmi špatných klimatických podmínkách, které viditelnost negativně ovlivňují – zejména mlha, sněžení a hustý déšť.

V současnosti existuje řada různých způsobů detekce jízdních pruhů z obrazového signálu kamery. Některé metody fungují na základě detekce hran a přímek z obrazového signálu, jiné prostřednictvím transformace obrazu. Existují také přístupy založené na barevné a světelné analýze obrazu. Následující část této práce tedy bude věnována popisu nejčastěji používaných metod.

2.1 Inverzní perspektivní mapování (IPM)

Jednou z metod využívajících transformace obrazového signálu je inverzní perspektivní mapování (dále jen IPM), které řeší problém perspektivního zobrazení snímané scény – tedy, že vzdálenější objekty vnímáme jako menší ve srovnání s objekty blízkými, což názorně ilustruje Obrázek 2-1. V tomto případě je snímanou scénou vozovka s jízdními pruhy, které se s přibývajícím vzdáleností zužují do jednoho bodu – tzv. úběžníku (označen O). Podstatou metody IPM je převod ze středového do ortogonálního promítání, což má za následek stejnou vzdálenost všech bodů v obraze od kamery. Zjednodušeně by se dalo říci, že perspektivní zobrazení je převedeno na zobrazení z ptačí perspektivy [1].



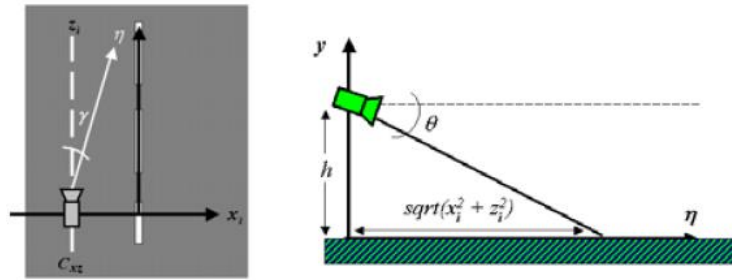
Obrázek 2-1 Perspektiva IPM (převzato z[2])

Matematicky by se dalo IPM popsat jako projekce z trojrozměrného Euklidovského prostoru $\mathbf{W} = \{(x, y, z)\} \in \mathbf{E}^3$ do dvojrozměrného planárního zobrazení $\mathbf{I} = \{(u, v)\} \in \mathbf{E}^2$, čímž dosáhneme požadovaného odstranění perspektivního zobrazení. Výpočet IPM modelu je pak tvořen pomocí dvou rovnic (2.1 a 2.2), kde Y (2.1) označuje úhel svíraný optickou osou kamery vůči ose y . Ξ pak označuje úhel svíraný optickou osou kamery a osou z . α představuje zorné pole kamery a n, m označuje rozlišení obrazu [3].

$$u(x, 0, z) = \frac{\gamma(x, 0, z) - (Y - \alpha)}{\frac{2\alpha}{n - 1}} \quad (2.1)$$

$$v(x, 0, z) = \frac{\theta(x, 0, z) - (\Xi - \alpha)}{\frac{2\alpha}{m - 1}} \quad (2.2)$$

Kde $\gamma = \tan^{-1}\left(\frac{z}{x}\right)$ a $\theta = \tan^{-1}\left(\frac{h}{\sqrt{x^2 + z^2}}\right)$



Obrázek 2-2 Model IPM zobrazení (převzato z[3])

Výhodami tohoto přístupu jsou především přesnost a robustnost algoritmu. Nevýhodou je pak vyšší výpočetní náročnost. Důležité je u IPM také vhodné umístění kamery do pomyslného středu obrazu a vhodná volba úhlu, kterou svírá optická osa kamery vůči vozovce. Pokud by byl úhel příliš úzký, byla by analyzována příliš malá plocha před vozidlem, a obdobně při příliš širokém úhlu by systém neposkytoval relevantní údaje, což by vedlo především k vysokému nárůstu chybných detekcí vybočení z jízdního pruhu [2].

2.2 Detekce hran

Jinou možností detekce jízdních pruhů na vozovce je použití algoritmu na detekci hran a následné použití Houghovy transformace pro detekci přímek. Při využití tohoto přístupu je nutné provést nejprve filtraci vstupního obrazu a následně aplikovat vybraný algoritmus detekující hrany v obrazovém signálu.

Poté je využita Houghova transformace, která aproximuje přímky podle hranových bodů. Z důvodu úspory výpočetního výkonu je velmi vhodné provést rozdělení vstupního obrazu do dvou, případně tří „Oblastí zájmu“ (Regions of Interest, dále jen ROI). V každé oblasti zvlášť je poté detekován jeden jízdní pruh (levý, středový a pravý), a to pomocí výše uvedené Houghovy transformace [4].

2.2.1 Filtrace obrazu

Vodorovné dopravní značení bývá zpravidla bílé na černé (případně šedé) vozovce a pro lidské oko není problém přesně stanovit jeho hranice. Pro počítačový algoritmus však rozpoznání tak triviální není, a z toho důvodu jsou používány různé filtry obrazu za účelem zvýšení pravděpodobnosti úspěšné detekce požadovaných oblastí na vozovce. Mezi nejčastější obrazové úpravy patří převod barevného obrazu na stupňů šedi, filtrování šumu, binarizace, zostření obrazu, případně jeho jasová a barevná korekce [5].

Převod obrazu do stupňů šedi

Jednou z prvních a také nejčastěji prováděných obrazových úprav je zpravidla převod barevného obrazu do stupňů šedi. Důvodem tohoto převodu je především úspora výpočetního výkonu, neboť místo tří barevných kanálů je zpracováván pouze jeden. Tento postup je aplikován u algoritmů, při nichž barva obrazu není stěžejní pro další zpracování, v takovém případě jde o přípravu obrazu pro hranový detektor. Při převodu obrazu do stupňů šedi je na každý pixel obrazu aplikován vzorec (2.3), kde proměnné R , G a B označují hodnoty jednotlivých barevných kanálů obrazu (červená, zelená a modrá) [15].

$$Y = 0,299R + 0,587G + 0,114B \quad (2.3)$$

Vyhlazení obrazu

Vyhlazení obrazu slouží k odstranění šumu vzniklého při jeho digitalizaci. Přítomnost šumu by mohla negativně ovlivňovat algoritmus detekující hrany v obraze. Odstranění šumu je zpravidla aplikováno na snímek, který je již převedený do stupňů šedi. Uvnitř zpracovávaného snímku I je nejprve pro každý jednotlivý pixel vypočtena hodnota lokálního průměru μ dle vzorce (2.4) [15]. Proměnné M a N označují sousední pixely, jejichž rozdíl reprezentuje proměnná Δ .

$$\mu = \frac{1}{MN} \sum_{n_1, n_2 \in \Delta} I(n_1, n_2) \quad (2.4)$$

Následně je pro každý pixel vypočtena odchylka dle vzorce (2.5):

$$\sigma^2 = \frac{1}{MN} \sum_{n_1, n_2 \in \Delta} I(n_1, n_2) - \mu^2 \quad (2.5)$$

Výsledná hodnota pixelu ve filtrovaném obraze je vypočtena dle (2.6), kde hodnota v^2 označuje průměrnou hodnotu všech lokálních odchylek. Tím je eliminována nevýhoda vyhlazení obrazu prostým průměrováním všech pixelů – při velkém rozdílu sousedních pixelů (např. na hraně) dochází k ovlivnění ostroty hrany, což může negativně ovlivnit další zpracování obrazu [15].

$$J(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} [I(n_1, n_2) - \mu] \quad (2.6)$$

Jinou možností vyhlazení obrazu je použití tzv. Gaussova filtru, což je speciální operátor využíváný pro potlačení tzv. Gaussova šumu. V principu je Gaussův filtr podobný prostému průměrování, avšak je použito jiné jádro, které reprezentuje model Gaussova rozložení. Rovnice (2.7) popisuje algoritmus výpočtu Gaussova filtru pro dvourozměrný obraz, kde proměnné x, y označují souřadnice a σ je směrodatná odchylka [14].

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.7)$$

V praxi se využívá implementace algoritmu Gaussova filtru prostřednictvím konvoluční masky, jejíž jádro tvoří matice s hodnotami ovlivňovanými pouze směrodatnou odchylkou σ , která určuje poloměr rozostření obrazu v pixelech. Příkladem konvoluční masky s jádrem o rozměru 3x3 je vzorec (2.8) [14].

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.8)$$

Binarizace obrazu

Binarizace umožňuje segmentaci obrazu na jednotlivé objekty a pozadí, v tomto případě se jedná o oddělení jízdních pruhů od pozadí – vozovky. Segmentace je pak docílena prostřednictvím převodu obrazu do černobílého formátu, zpravidla pomocí tzv. prahování. Segmentace prahováním se dosáhne tím, že cílovým objektům je v obraze přiřazena hodnota 1 a pozadí hodnota 0, výstupem je pak binární obraz. Takto je popisováno tzv. prahování prosté.

Dále existuje např. prahování adaptivní, částečné či s více prahy, nicméně pro algoritmus detekce jízdních pruhů zpravidla dostačuje použití prahování prostého [15]. Základním problémem je vhodné určení hodnoty prahu, tedy meze, jež určuje, jaká hodnota jasu bude přiřazena cílovému objektu a jaká pozadí. Hodnota prahu může být definována pevně, což je ale pro dynamicky se střídající scénu před vozidlem naprosto nevhodné. Je tedy nutné stanovit hodnotu prahu z analýzy histogramu obrazu. Histogramem rozumíme vektor absolutních četností výskytu jednotlivých barev ve snímku. Histogram poskytuje základní informaci o úrovni jasu v analyzovaném snímku [16]. Jednou z metod využívajících vektorovou analýzu histogramu je iterativní metoda nalezení prahu. V této metodě je během každé iterace i upřesňován aktuální práh T^i (2.9), a to až do okamžiku bodu stability, tzv. idempotence (kdy se hodnota prahu po dvou po sobě jdoucích průchodech nezmění). Tato metoda je založena na myšlence, že aktuální práh T^i od sebe dělí právě dvě množiny pixelů – pixely pozadí a pixely objektů. V každé iteraci je také stanovena průměrná hodnota $E^i(bg)$ (2.10) a z těchto dvou hodnot je stanoven práh další iterace T^{i+1} jako aritmetický průměr. Pro první iteraci lze uvažovat práh $T^0 = 1$, tj. pozadí je tvořeno pixely s jasovou hodnotou 0, ostatní pixely se považují za objekty [9].

$$T^{i+1} = \frac{E^i(bg) + E^i(fg)}{2} \quad (2.9)$$

$$E^i(bg) = \frac{h_c^w(T^i)}{h_c(T^i)} \quad (2.10)$$

$$E^i(fg) = h_c^w(255) - \frac{h_c^w(T^i)}{h_c(255) - h_c(T^i)} \quad (2.11)$$

Další možností stanovení prahu z analýzy histogramu je použití tzv. Otsuovy metody nalezení prahu [7]. Tento algoritmus předpokládá existenci dvou tříd pixelů v obraze (pozadí a objekt) a z toho vyplývající bimodální histogram takového obrazu. Na základě tohoto předpokladu algoritmus vypočítává optimální práh oddělující tyto třídy tak, aby odchylky uvnitř každé z nich byly minimální. Děje se tak prostřednictvím váhovaných sum odchylek jednotlivých tříd (2.12). Váhy pravděpodobnosti ω_1 a ω_2 dvou tříd jsou oddělené prahem t a proměnné σ_0^2 , σ_1^2 označují odchylky jednotlivých tříd [7].

$$\sigma_\omega^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (2.12)$$

Váhy pravděpodobnosti ω_1 a ω_2 jsou vypočteny z histogramu p :

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \quad (2.13)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \quad (2.14)$$

Minimalizace odchylek uvnitř jednotlivých tříd je docíleno maximalizováním odchylek mezi oběma třídami(2.15):

$$\begin{aligned} \sigma_b^2(t) &= \sigma^2 - \sigma_\omega^2(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \\ &= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \end{aligned} \quad (2.15)$$

Přičemž průměrné hodnoty jednotlivých tříd $\mu_{0,1T}(t)$ jsou získány ze vztahů:

$$\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_0(t)} \quad (2.16)$$

$$\mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)} \quad (2.17)$$

$$\mu_T = \sum_{i=0}^{L-1} ip(i) \quad (2.18)$$

Princip fungování algoritmu:

1. Výpočet histogramu a pravděpodobnosti pro každou složku intenzity
2. Nastavení počátečních hodnot $\omega_i(0)$ a $\mu_i(0)$
3. Pro všechny prahové hodnoty $t = 1 \dots$ maximální intenzita
 - a. Aktualizovat hodnoty ω_1 a μ_i
 - b. Spočítat hodnotu $\sigma_b^2(t)$
4. Požadovaný práh odpovídá maximální hodnotě $\sigma_b^2(t)$

2.2.2 Hranové detektory

Klíčovým úkolem pro úspěšnou detekci jízdních pruhů v obraze je detekce hran. Hrana v obraze je chápána jako ostrý přechod, kde dochází ke změně hodnoty jasu, případně barvy, a to na úrovni jednotlivých pixelů. Pro účely této práce je požadována detekce hran, které tvoří přechod mezi povrchem vozovky a vodorovným dopravním značením.

Existuje mnoho algoritmů detekujících hrany v obraze, přičemž základní metodou těchto algoritmů je výpočet pomocí diferenciálního operátoru. Tento způsob je aktuálně nejčastěji používaným řešením problematiky tzv. počítačového vidění. Mezi nepoužívanější metody se řadí Robertsův, Sobelův, Prewittové, Kirschův, Gauss-Laplaceův a Cannyho operátor [9].

Sobelův operátor

Sobelův operátor provádí dvourozměrné měření prostorového gradientu, čímž zvýrazňuje oblasti s vysokou prostorovou frekvencí korespondující s výskytem hran. Typicky je používán k nalezení přibližné absolutní hodnoty gradientu v každém pixelu vstupního černobílého obrazu. Výpočet je prováděn prostřednictvím dvou konvolučních matic G_x a G_y o rozměrech 3×3 , které jsou vzájemně otočeny o 90° [8].

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.19 \text{ a, b})$$

Tyto matice jsou navrženy tak, aby v maximální možné míře odpovídaly hranám ve vertikálním (G_x) a horizontálním (G_y) směru. Matice mohou být použity společně či každá zvlášť pro výpočet absolutní hodnoty gradientu v požadovaném směru. Rozsah absolutní hodnoty gradientu je dán vztahem (2.20):

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.20)$$

Pro urychlení výpočtu se v praxi výpočet aproximuje vztahem (2.21):

$$G = |G_x| + |G_y| \quad (2.21)$$

Výhodou užití Sobelova operátoru je zjištění úhlu orientace hrany (resp. jeho relativní polohy vůči pixelové mřížce obrazu), který je dán vztahem (2.22):

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.22)$$

Prewittové operátor

Obdobně jako u Sobelova operátoru, také Prewittové operátor používá k výpočtu dvě matice (G_x a G_y) o rozměru 3×3 , avšak s jinými hodnotami (2.23 a, b) uvnitř matic. Obdobně tyto matice lze použít dle požadovaného směru jak zvlášť, tak dohromady. Proměnná A pak označuje zdrojový obraz.

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \cdot A \quad Gy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} \cdot A \quad (2.23 \text{ a, b})$$

Výpočet hodnoty gradientu je prováděn dle vztahu (2.20), úhel gradientu je pak možné vypočítat ze vztahu (2.24):

$$\Theta = \arctan 2 \cdot (Gy, Gx) \quad (2.24)$$

Výhodou tohoto operátoru je jednodušší implementace ve srovnání se Sobelovým operátorem, avšak je nutné počítat s vyšší citlivostí na přítomnost šumu v obraze. Algoritmus je zpravidla aplikován na černobílé obrazy [9].

Laplaceův operátor

Laplaceův operátor (zkráceně Laplaceián) je řazen mezi algoritmy pracující s druhou derivací prostorového gradientu. Laplaceián má stejné vlastnosti v obou směrech, díky čemuž je invariantní vůči rotaci. Na rozdíl od dvou výše uvedených operátorů tak jádro Laplaceiánu tvoří pouze jedna matice (2.25), prostřednictvím které jsou detekovány hrany jak ve vertikálním, tak v horizontálním směru. Existuje také varianta určená pro detekci diagonálních hran (2.26). Nevýhodou Laplaceiánu je velmi vysoká citlivost na přítomnost šumu v obraze z důvodu výpočtu prostřednictvím druhé derivace.

$$G = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (2.25)$$

$$G = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (2.26)$$

Výsledným gradientem mohou být kladné i záporné hodnoty intenzity jasu, v praxi se proto počítá s absolutní hodnotou gradientu. Jelikož má Laplaceián obecně pozitivní a negativní odezvu na každou hranu, využití absolutní hodnoty tuto odezvu zvýrazní. V případě potřeby vyšší přesnosti detekce je možné využít rozšířené jádro, které tvoří matice o velikosti 5x5, kde prostřední prvek má hodnotu 24 a všechny ostatní -1.

Z důvodu velmi vysoké citlivosti Laplaceiánu na přítomnost šumu v obraze se v praxi používá kombinace Laplaceiánu s vhodným filtrem, který odstraní šum a současně zachová hrany v obraze. Nejčastěji je pro tento účel využíván Gaussův filtr (2.7), kombinace těchto algoritmů se pak označuje jako Laplacian of Gaussian (LoG). Tento algoritmus využívá faktu, že první i druhá derivace a následně konvoluce jsou lineárními operacemi, a z toho důvodu je možné zaměnit postup jejich výpočtu. Nejprve je tedy provedeno filtrování obrazu pomocí Gaussova filtru a následně proveden výpočet Laplaceiánu. Algoritmus LoG je vypočten dle rovnice (2.27), přičemž σ označuje Gaussovo rozložení odchylky [10].

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.27)$$

Kde proměnné x, y označují souřadnice dvourozměrného skalárního pole.

Cannyho detektor

Cannyho hranový detektor je v současné době považován za optimální algoritmus pro úspěšnou detekci hran v obraze. Mezi jeho vlastnosti patří nízká chybovost, přesná lokalizace polohy detekované hrany v obraze a také jednoznačná odezva tak, aby nedocházelo ke dvojité odezvě na jednu hranu. Oproti výše uvedeným jednoduchým operátorům je Cannyho detektor komplexním algoritmem, jehož funkce se skládá z více kroků. Mezi ty základní patří [11]:

1. Odstranění šumu z obrazu – na vstupní obraz je aplikován zpravidla Gaussův filtr pro vyhlazení obrazu a odstranění šumu.
2. Určení gradientu – je vypočtena aproximace první derivace prostřednictvím jednoduchého operátoru detekujícího hrany v obraze, nejčastěji se jedná o Sobelův operátor.
3. Nalezení lokálních maxim – v obraze s detekovanými hranami jsou vyhledána lokální maxima těchto hran a následně jsou odebrány body, jejichž gradient této hodnoty nedosahuje. Cílem tohoto kroku je ztenčení detekovaných hran z důvodu jednoznačnosti detekce. Tento proces je označován jako „non-maximal suppression“.
4. Prahování s hystezí – posledním krokem detekce hran je eliminace nevýznamných hran prostřednictvím prahování s hystezí tvořené dvěma hodnotami $T1$ a $T2$, přičemž platí $T1 > T2$. Pokud je hodnota jasu zpracovávaného pixelu vyšší než $T1$, je změněna na maximální hodnotu (zpravidla 255), naopak je-li nižší než práh $T2$, je označena za nevýznamnou a změněna na 0. Tímto způsobem vznikne binarizovaný obraz. V praxi je hodnota $T2$ určena jako polovina $T1$. Použití hystereze zajišťuje to, aby detekované hrany byly celistvé a nebyly fragmentovány.

2.2.3 Houghova transformace

Klasická Houghova transformace se používá pro detekci přímek v obraze, z toho důvodu je hojně využívána při řešení problematiky tzv. počítačového vidění. Jde také o často využívaný algoritmus detekce jízdních pruhů na vozovce z obrazového signálu, který byl zpracován vhodným hranovým detektorem a je tedy binarizovaný.

Transformace je založena na skutečnosti, že každou přímkou v rovině (x, y) lze jednoznačně popsat dvojicí parametrů (ρ, θ) , které určují vzdálenost od středu souřadnicového systému (ρ) a úhlu od osy x (θ) . Přímkou lze v uvedeném případě popsat rovnicí (2.28) [12]:

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta \quad (2.28)$$

Pro účely detekce jízdních pruhů z obrazu reprezentují proměnné (x, y) souřadnice konkrétního pixelu v obraze, θ označuje úhel přímky od osy x v rozsahu od 0° do 90° a ρ vyjadřuje vzdálenost přímky v pixelech od počátku soustavy. Aplikování Houghovy transformace na soustavu bodů tvořících hranu vede ke vzniku dvourozměrné funkce $A(\rho, \theta)$, která reprezentuje počet hranových bodů vyhovujících rovnici (2.28). V praxi je využíván postup, kdy je při každém výpočtu rovnice (2.28) pro dané θ hodnota zaznamenána ve dvourozměrném poli Houghova prostoru, označovaného jako akumulátor (akumuluje evidenci o výskytech). Lokální maxima funkce A pak mohou být využita pro detekci přímek procházejících hranovými body (pixely) [13].

2.2.4 Rozdělení obrazu do oblastí zájmu

Z důvodu úspory výpočetního výkonu je žádoucí, aby byla scéna rozdělena do dvou, případně tří separátních oblastí, tzv. oblastí zájmu (ROI). Z hlediska pořadí jednotlivých kroků je vhodné provést rozdělení binarizovaného obrazu, poté pro každou oblast odděleně aplikovat vybraný hranový detektor a následně prostřednictvím Houghovy transformace detekovat levý a pravý jízdní pruh. Pro následný algoritmus detekce vybočení z jízdního pruhu je nutné stanovit počátek rozdělení do jednotlivých oblastí. Označíme-li počátek C , nalézající se ve středu osy x , pak x_{max} označuje šířku ROI (2.29) [4]:

$$C = \frac{x_{max}}{2} \quad (2.29)$$

2.2.5 Detekce vybočení vozidla

Po úspěšné detekci jízdních pruhů na vozovce je dalším krokem použití algoritmu detekujícího opuštění daného pruhu a následné vizuální či akustické varování řidiče před nebezpečím. Jednou z možností je analýza vybočení vozidla na základě parametrů Euklidovské vzdálenosti. Uvažovány jsou celkem čtyři stavy – levé vybočení, pravé vybočení, nebezpečí a bezpečný stav. Pokud se vozidlo nachází přibližně uprostřed jízdního pruhu, systém je ve stavu „bezpečí“.

V případě, že vozidlo vlivem nepozornosti řidiče začne vybočovat do některé ze stran, přepne se systém do stavu „levé či pravé vybočení“ a řidič obdrží varování o riziku. Při velmi rychlém vybočení vozidla systém ve stavu „nebezpečí“ okamžitě varuje řidiče před nebezpečnou situací [4].

Algoritmus detekce vybočení je definován prostřednictvím čtyř parametrů – E_L , E_R , E_{LR} a ξ , které jsou vypočteny na základě Euklidovské vzdálenosti. Parametr E_L definuje Euklidovskou vzdálenost mezi počátkem C a středem značení levého jízdního pruhu (mpl):

$$E_L = \sqrt{(C_x - mpl_x)^2 + (C_y - mpl_y)^2} \quad (2.30)$$

Pro výpočet parametru E_R – Euklidovské vzdálenosti mezi počátkem (C) a středovým bodem značení pravého jízdního pruhu se postupuje analogicky:

$$E_R = \sqrt{(C_x - mpr_x)^2 + (C_y - mpr_y)^2} \quad (2.31)$$

Parametr E_{LR} označuje Euklidovskou vzdálenost mezi středovými body levého a pravého jízdního pruhu:

$$E_{LR} = \sqrt{(mpl_x - mpr_x)^2 + (mpl_y - mpr_y)^2} \quad (2.32)$$

V případě vybočení vozidla do jedné ze stran stoupá vzdálenost mezi středy levého a pravého pruhu. Tento parametr hraje významnou roli při rozhodování stavů „nebezpečí“ a „bezpečí“. Dále je definována proměnná ξ , jež označuje poměr vybočení. Jde o hodnotu poměru Euklidovské vzdálenosti mezi počátkem a středem levého jízdního pruhu k Euklidovské vzdálenosti počátku vůči středu pravého pruhu (2.33):

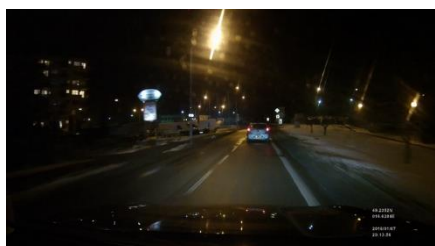
$$\xi = \frac{E_L}{E_R} \quad (2.33)$$

Je-li hodnota poměru vybočení rovna 1, je systém ve stavu „nebezpečí“, toto však platí pouze za předpokladu, že je hodnota E_{LR} nižší, než stanovený práh.

V opačném případě systém přejde do stavu „bezpečí“. Za situace, že jedoucí vozidlo začne pozvolna vybočovat směrem vlevo, dochází k poklesu hodnoty poměru vybočení ξ , současně klesá vzdálenost E_L , zatímco E_R stoupá. Analogická situace nastává při vybočení vozidla vpravo. Dle je ještě nutné pro správnou funkci algoritmu experimentálně stanovit prahové hodnoty, při kterých systém přechází z jednoho stavu do druhého [4].

2.3 Zpracování snímané scény před vozidlem

Pro scénu snímanou před vozidlem je charakteristická vysoká dynamičnost a prakticky neustále se měnící světelné podmínky. Ty závisí zejména na ročním období, denní době, ale také na aktuálním počasí. Charakter osvětlení scény je dále ovlivňován také umělými zdroji světla, zejména pouličními lampami a světlomety vozidla. Z toho důvodu může docházet jak k přesvětlení, tak k nedostatečnému osvětlení snímané scény, a to zejména za jízdy ve tmě či za nepříznivých meteorologických podmínek (mlha, sněžení či hustý déšť).



Obrázek 2-3 Náhled scény před vozidlem

2.3.1 Vodorovné dopravní značení na pozemních komunikacích

Pro zajištění správné funkce aplikovaného detektoru je nutné prostudovat způsob značení jízdních pruhů na pozemních komunikacích v našich podmínkách tak, jak je definuje platná legislativa. Tyto údaje jsou uvedeny v tzv. technických podmínkách, jejichž autorem je Ministerstvo dopravy České republiky, odbor pozemních komunikací.

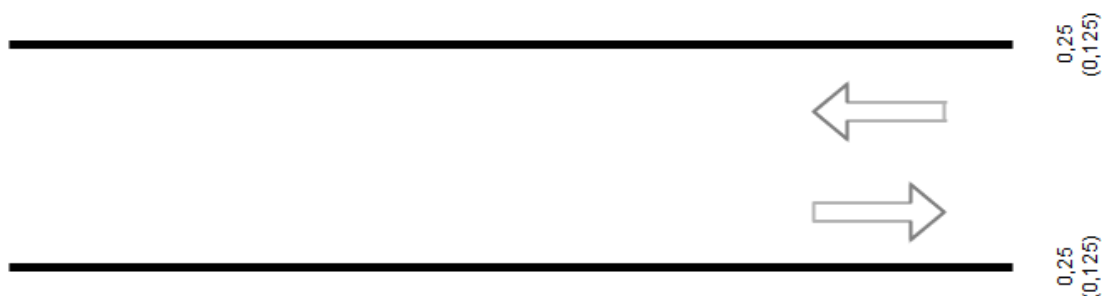
Vodorovnému značení je věnován dokument TP 133 – Zásady pro vodorovné dopravní značení na pozemních komunikacích [16], v němž je uveden přesný popis jednotlivých používaných značek včetně jejich rozměrů.

Na pozemních komunikacích rozlišujeme následující základní vodorovné značky:

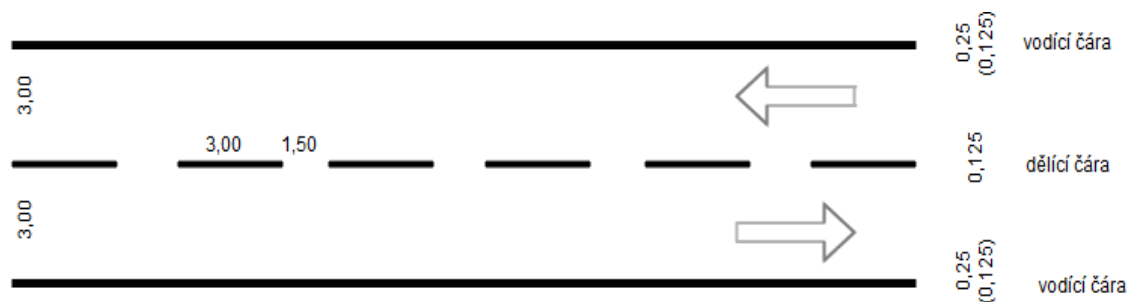
- Podélná čára souvislá – v šířce 0,125 m slouží k oddělení jízdních pruhů, v šířce 0,25 m pak slouží k oddělení odbočovacího nebo připojovacího pruhu od průběžného jízdního pruhu a k oddělení vyhrazeného jízdního pruhu

- Dvojitá podélná čára souvislá – v šířce 0,125 m s mezerou shodné šířky se používá ke zvýrazněnému oddělení protisměrných jízdních pruhů.
- Podélná čára přerušovaná – v šířce 0,125 m slouží k oddělení jízdních pruhů, vedení jízdních pruhů v prostoru křižovatky, oddělení jízdního pruhu od tramvajového tělesa v úrovni vozovky, oddělení jízdního pruhu pro pomalá vozidla; a dále v šířce 0,25 m slouží k oddělení odbočovacího nebo připojovacího pruhu od průběžného jízdního pruhu, vyznačení okraje jízdního pásu ve směru hlavní pozemní komunikace a také k oddělení vyhrazeného jízdního pruhu.
- Podélná čára souvislá doplněná čarou přerušovanou – v šířce 0,125 m s mezerou shodné šířky se užívá k oddělení jízdních pruhů a oddělení jízdního pruhu pro pomalá vozidla.
- Vodící čára – v šířce 0,25 m slouží k vyznačení okraje vozovky na směrově rozdělené pozemní komunikaci; v šířce 0,125 m potom vyznačuje okraj vozovky směrově nerozdělené pozemní komunikace při šířce zpevněné krajnice $< 0,5$ m.

Dle uvedeného dokumentu se jízdní pruhy vyznačují a oddělují dělicími čarami o šířce čáry 0,125 m v případě, že je šířka vozovky mezi vodíci čarami větší než 6 m. V opačném případě se jízdní pruhy nevyznačují, viz Obrázek 2-4. Minimální šířka pro vyznačení jednoho jízdního pruhu je stanovena na 2,75 m. Příklad vodorovného značení na vozovce, jejíž šířka je větší než 6 m a současně menší než 7 m ilustruje Obrázek 2-5.



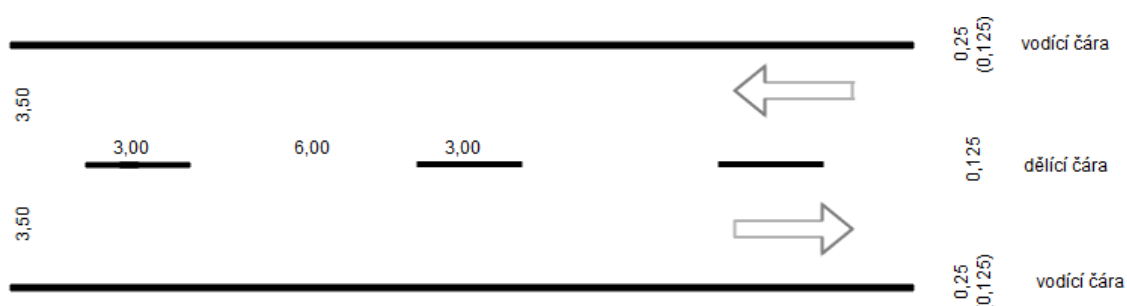
Obrázek 2-4 Příklad značení na vozovce o šířce menší než 6 m (na základě [16])



Obrázek 2-5 Příklad značení na vozovce o šířce 6 až 7 m (na základě [16])

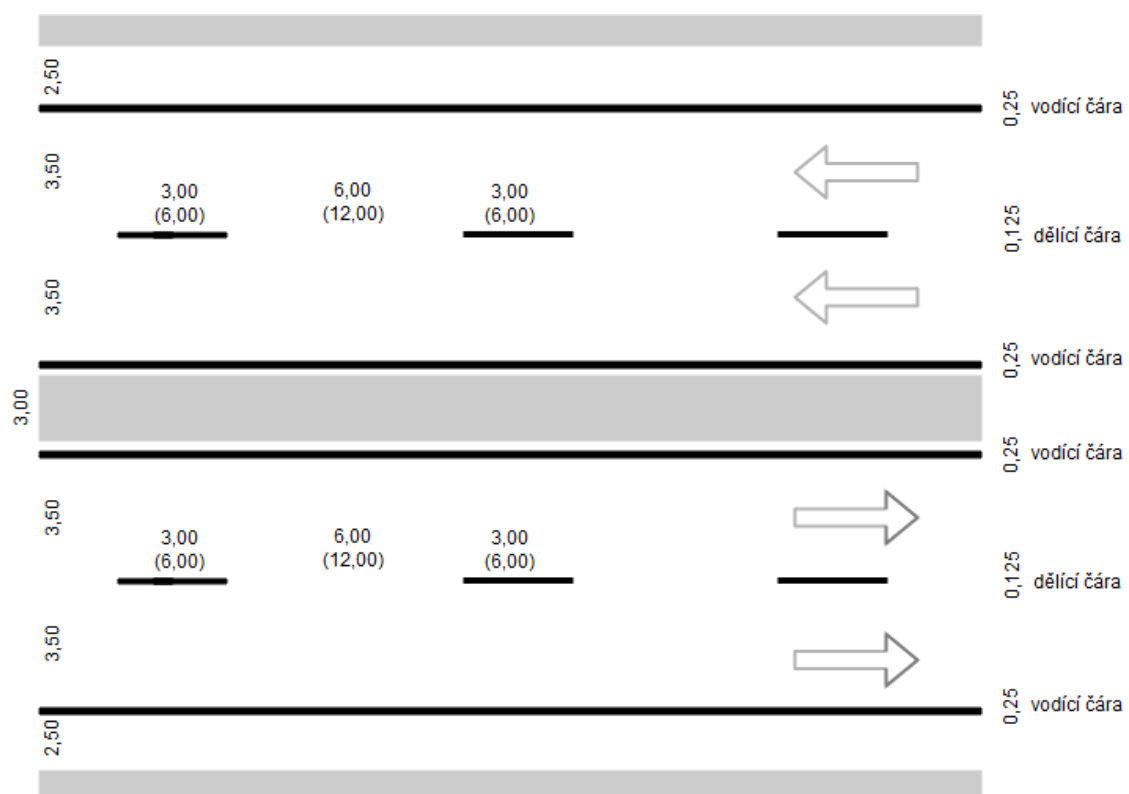
Na obousměrné pozemní komunikaci se šířkou vozovky 7 m a více je vodorovné značení provedeno dle skladebných prvků šířkového uspořádání příslušné kategorie pozemní komunikace dané ČSN 73 6101, případně ČSN 73 6110. Příklad takového uspořádání vodorovného značení ilustruje Obrázek 2-6. V praxi se nejčastěji jedná o silnice I. a II. třídy. Rozdělení silnic do tří tříd je definováno v zákoně č. 13/1997 Sb., *Zákon o pozemních komunikacích*, kde §5 definuje třídy silnic dle svého určení a dopravního významu takto:

- Silnice I. třídy – určena zejména pro dálkovou a mezinárodní dopravu
- Silnice II. třídy – určena pro dopravu mezi okresy
- Silnice III. třídy – určena k vzájemnému spojení obcí nebo jejich napojení na ostatní pozemní komunikace



Obrázek 2-6 Příklad značení na vozovce širší než 7 m (na základě [16])

Šířkové uspořádání směřově rozdělené pozemní komunikace, jehož příklad ilustruje Obrázek 2-7, je typické pro dálnice a silnice pro motorová vozidla.



Obrázek 2-7 Příklad značení směrově rozdělené pozemní komunikace (na základě [16])

2.3.2 Aplikace vybraných algoritmů na obrazový signál

Pro účel výběru vhodného algoritmu detekujícího jízdní pruhy bylo provedeno experimentální ověření funkce jednotlivých výše popsaných algoritmů. Za tímto účelem bylo pořízeno několik krátkých testovacích sekvencí autokamerou Truecam A7S, která je využívána jako černá skříňka pro záznam scény před vozidlem. Náhled jednotlivých testovacích sekvencí je uveden na obrázku 2-3. Tyto sekvence byly záměrně pořízeny za různých světelných podmínek – za denního světla, za šera a za tmy s umělým osvětlením. Jednotlivé algoritmy byly testovány pomocí výpočetního softwaru MATLAB, ve kterém je většina požadovaných algoritmů implementována.

Testovací snímky byly nejprve z barevného formátu převedeny do stupňů šedi, byl aplikován Gaussův filtr pro vyhlazení obrazu a odstranění šumu a dále byly pro srovnání použity jednotlivé detektory hran (Sobelův, Prewittové a Cannyho detektor). Následně bylo také experimentálně testováno vykreslení přímek na základě detekovaných jízdních pruhů, a to prostřednictvím Houghovy transformace.



Původní snímek (a)



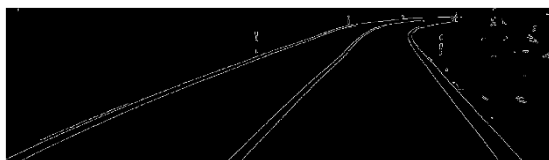
Převod do stupňů šedi a aplikace Gaussova filtru (b)



Sobelův operátor (c)



Prewittové operátor (d)



Cannyho detektor (e)



Houghova transformace na základě Cannyho detektoru (f)

Obrázek 2-8 Zpracování obrazu vybranými algoritmy

Experimentálně získané výsledky se shodují s teoretickými předpoklady. Nejlepších výsledků z detektorů hran dosahuje poměrně stabilně Cannyho detektor, který při vhodné volbě obou prahových hodnot vykresluje souvisle detekované okraje jízdních pruhů na vozovce. U Sobelova a Prewittové operátoru se hůře hledá optimální prahová hodnota, kdy jsou jasně patrné detekované jízdní pruhy a současně není obraz zatížen hranami detekovanými z okolní scenerie. Z obrázků (c) a (d) (Obrázek 2-8) je patrné, že i přes poměrně vysokou míru nežádoucí detekce hran okolní scenerie, nejsou detekované linie jízdních pruhů souvislé. Z toho důvodu byl pro vykreslení přímek na základě detekovaných linií zvolen Cannyho hranový detektor. Na obrázku (f) (Obrázek 2-8) jsou pomocí Houghovy transformace vykresleny přímky spojující body, které tvoří lokální maxima uvedené funkce.

3. HARDWARE

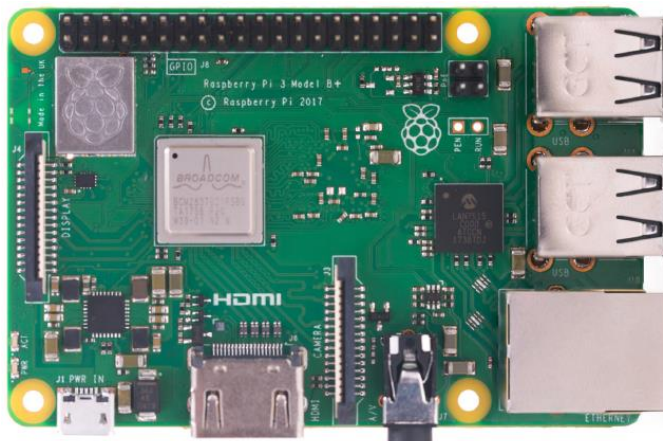
Dle účelu použití zařízení bylo stanoveno několik základních požadavků, kterým by měl navržený hardware vyhovovat. Požadavky jsou následující: kompaktní velikost zařízení, vysoký výpočetní výkon, možnost připojení kompatibilního kamerového modulu, v neposlední řadě také přijatelná cena zařízení a jeho dostupnost. Z výčtu je patrné, že některé požadavky jsou téměř protichůdné, je tedy nutné najít vhodný kompromis. Vzhledem k tomu, že bude zařízení umístěno na čelním skle vozidla, je výsledná velikost celého zařízení faktorem stěžejním. Dle platné legislativy, konkrétně dle § 34 odst. 3 vyhlášky MDS č. 341/2002 Sb., „...*nesmí být v zorném poli řidiče umístěny žádné předměty, které by omezovaly výhled řidiče všemi směry*...“. V praxi je obvykle tolerováno umístění takového zařízení za vnitřní zpětné zrcátko. Toto umístění vyhovuje i z hlediska detekce vybočení vozidla, pro kterou je vhodné umístění snímací kamery přibližně ve středu horizontální roviny čelního skla vozidla. Dalším stěžejním faktorem je výpočetní výkon zařízení. Vzhledem k tomu, že při zpracování obrazu je nutné analyzovat velký počet obrazových bodů pořizovaných kamerou v reálném čase, je nasnadě, že zařízení musí disponovat vysokým výpočetním výkonem, ideálně procesorem architektury ARM s více jádry.

3.1 Raspberry Pi

Raspberry Pi (dále jen RPi) je malý jednodeskový počítač o velikosti platební karty (85x56x17 mm). Platforma RPi byla vyvinuta v roce 2012 britskou nadací Raspberry Pi Foundation, jejímž cílem byla podpora výuky informatiky na školách. Vzhledem ke své všestrannosti a cenové dostupnosti se tato brzy rozšířila i mezi širokou veřejnost a v současnosti se jedná o jedno z nejpobulárnějších zařízení v daném segmentu. Aktuálně je na trhu k dispozici několik různých verzí tohoto zařízení – od nejjednoduššího Raspberry Pi Zero až po nejnovější model Raspberry Pi 3 Model B+.

Na základě výše specifikovaných požadavků bylo vybráno zařízení Raspberry Pi 3 Model B+ (Obrázek 3-1), které je v současnosti nejnovějším a nejvýkonnějším modelem Raspberry Pi. Je osazen 64bitovým čtyřjádrovým procesorem architektury ARM o taktu 1,4 GHz a disponuje operační pamětí o velikosti 1 GB. Podrobná specifikace zařízení je uvedena v tabulce č. 1. Z hlediska konektivity je zařízení vybaveno čtyřmi porty standardu USB 2.0, dále HDMI portem pro přenos obrazového (a zvukového) signálu a také kompozitním audio a video portem formátu jack 3,5 mm. Síťovou konektivitu zabezpečuje ethernet port (RJ-45) a dále WLAN standardu IEEE 802.11ac podporující kmitočet 2,4 a 5 GHz. Podporována je také bezdrátová komunikace Bluetooth standardu 4.2. Pro připojení kompatibilní kamery je určen specializovaný CSI camera port a obdobně pro připojení LCD panelu je určen DSI (Display Serial Interface) port.

Vzhledem k tomu, že RPi nedisponuje vlastním úložištěm ve formě pevného disku či paměti typu Flash, je na desce přítomen slot pro paměťovou MicroSD kartu, ze které je spouštěn operační systém a současně slouží pro ukládání programů a dat.



Obrázek 3-1 Raspberry Pi 3 Model B+

Tabulka 3-1 Parametry Raspberry Pi 3 B+

Chip	BroadCom BCM2837B0
Procesor	Cortex A-53, 64-bit
Počet jader	4
Frekvence CPU	1.4 GHz
GPU	VideoCore IV 400 MHz
Operační paměť	1GB LPDDR2 SDRAM
USB 2.0	4
WLAN	IEEE 802.11. b/g/n/ac
Ethernet	Gigabit Ethernet over USB 2.0 (300 Mbps)
Audio/Video výstup	HDMI, komponentní výstup, MIPI DSI display port
Video vstup	MIPI CSI camera port, USB
Úložiště	Micro SD
Napájecí napětí	5 V
Napájecí proud	300 mA až 1,5A
Způsob napájení	Micro USB, GPIO, PoE (Power over Ethernet)

3.1.1 GPIO

Kromě výše uvedených standardizovaných konektorů se na desce RPi nalézá ještě 40pinový GPIO (General Purpose Input/Output) port, který kromě logických vstupů a výstupů obsahuje také sběrnice UART, I2C a SPI.

Dále se na sběrnici nachází napájecí piny s napětím 3,3 a 5 V, připojení GND a 2 piny, které jsou určeny pro přístup k EEPROM paměti zařízení. Kompletní rozpis jednotlivých pinů a jejich funkcí uvádí Tabulka 3-2.

Při použití GPIO sběrnice je nutné respektovat jednotlivá omezení, která specifikuje výrobce. Například maximální napětí, které lze na jednotlivé GPIO piny připojit, je 3,3 V, použití vyššího napětí by mělo za důsledek zničení sběrnice. Dalším důležitým parametrem je maximální proudová zatížitelnost jednotlivých pinů, která je stanovena na 8 mA. Taktéž je nutné se vyvarovat použití komunikace na sběrnici, která využívá logické úrovně vyšší než 3,3 V. Při nastavení GPIO pinů jako vstupních, lze využít možnosti nastavení interních pull-up nebo pull-down rezistorů. Pouze piny GPIO2 a GPIO3 mají pevně nastaveno použití pull-up rezistorů. Pro účely práce bude využito pulzně šířkové modulace na výstupním pinu (PWM). Deska podporuje softwarovou PWM na všech výstupních pinech, pro hardwarovou PWM je nutné využít pinů GPIO12, 13, 18 nebo 19.

Tabulka 3-2 Popis funkce jednotlivých pinů GPIO sběrnice

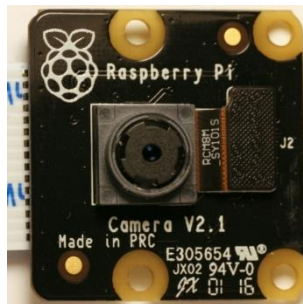
Označení	Funkce	Pin		Pin	Funkce	Označení
3V3	Power	1		2	Power	5 V
GPIO2	SDA I2C	3		4	Power	5 V
GPIO3	SCL I2C	5		6	-	Ground
GPIO4	-	7		8	UART0_TXD	GPIO14
Ground	-	9		10	UART0_RXD	GPIO15
GPIO17	-	11		12	PCM_CLK	GPIO18
GPIO27	-	13		14	-	Ground
GPIO22	-	15		16	-	GPIO23
3V3	Power	17		18	-	GPIO24
GPIO10	SPI0_MOSI	19		20	-	Ground
GPIO9	SPI0_MISO	21		22	-	GPIO25
GPIO11	SPI0_SCLK	23		24	SPI0_CE0_N	GPIO8
Ground	-	25		26	SPI0_CE1_N	GPIO7
ID_SM	I2C ID EEPROM	27		28	I2C ID EEPROM	ID_SC
GPIO5	-	29		30	-	Ground
GPIO6	-	31		32	-	GPIO12
GPIO13	-	33		34	-	Ground
GPIO19	-	35		36	-	GPIO16
GPIO26	-	37		38	-	GPIO20
Ground	-	39		40	-	GPIO21

3.2 Kamera

Vzhledem k požadavkům na přímou kompatibilitu a také rychlost zpracování obrazového signálu byl zvolen kamerový modul Raspberry Pi Camera Module v2. Tento modul o rozměrech 23,9x25x9 mm se připojuje do CSI camera portu pomocí plochého 15žilového kabelu. Jádrem modulu tvoří CMOS snímač Sony IMX219PQ s maximálním rozlišením 8 Megapixelů. Modul je z výroby osazen objektivem s pevnou ohniskovou vzdáleností $f = 3,04$ mm, při optické velikosti čipu $\frac{1}{4}$ “ činí horizontální úhel záběru $62,2^\circ$ (vertikální úhel záběru je $48,8^\circ$). Pro uvažovanou aplikaci by měl být uvedený vertikální úhel záběru dostačující. Vhodnou velikost ohniskové vzdálenosti lze ověřit jednoduchým výpočtem dle vzorce (3.1) [18]. Pro zvolenou aplikaci budeme uvažovat objekt ve vzdálenosti přibližně 5 metrů od kamery, jehož šířka je rovna 5 metrům. Z výsledku výpočtu je patrné, že uvedená ohnisková vzdálenost objektivu je pro danou aplikaci dostačující.

$$f = c \cdot \frac{d}{w} = 3,68 \cdot \frac{5 \cdot 10^3}{5 \cdot 10^3} = 3,68 \text{ mm} \quad (3.1)$$

Kde c označuje šířku snímače, d vzdálenost předmětu od kamery a w šířku objektu, přičemž vše je v jednotkách mm. Kompletní výčet technických parametrů snímače uvádí Tabulka 3-3.



Obrázek 3-2 Kamerový modul v2 pro Raspberry Pi

Tabulka 3-3 Parametry snímače Sony IMX219PQ

Velikost snímače	3,68 x 2,76 mm
Rozlišení snímače	3280 x 2464 pixelů
Velikost pixelu	1,12 x 1,12 μm
Optická velikost čipu	$\frac{1}{4}$ “
Max. rozlišení statických obrazů	8 Megapixelů
Rozlišení videosekvencí	1080p (30 sn/s), 720p (60 sn/s), 640 x 480 (60/90 sn/s)

3.3 Napájení

V souladu s technickou dokumentací zařízení je pro napájení požadován zdroj o nominálním napětí 5 V dodávající maximální proud 2,5 A. Přivedení napájení je možné realizovat prostřednictvím kabelu USB, který je zakončený microUSB portem. Jako zdroj napětí ve vozidle bude tedy použit standardizovaný USB napájecí zdroj do autozásuvky, který transformuje palubní napětí o nominální hodnotě 12 V na hodnotu 5 V a současně splňuje uvedenou proudovou zatížitelnost.

Toto řešení je výhodné zejména z hlediska snahy o minimalizaci rozměrů finálního zařízení, které tak nebude muset obsahovat přídatný obvod s regulátorem napětí. V praxi je toto řešení zpravidla využíváno u komerčně vyráběných autokamer a GPS navigací, které se dokupují jako příslušenství do vozidel.

Pro zamýšlený účel použití zařízení spolu s kamerovým modulem ve vozidle byla na základě výše uvedených požadavků zvolena kombinace napájecího zdroje do autozásuvky typu M8J068 a kabelu Nokia CA-179. Dle specifikace výrobce dodává napájecí zdroj napětí 5 V s výstupním proudem maximálně 3,4 A. Tato hodnota je dostačující pro napájení zařízení i při maximálním zatížení, a to včetně kamerového modulu, který dle dokumentace výrobce odebírá přibližně 200 mA.

Před začátkem testování zařízení je vhodné provést ověření, zda je navržené napájení dostatečné pro daný účel použití. Výrobce desky Raspberry Pi definuje jako hraniční hodnotu 4,70 V, pod kterou nesmí poklesnout napájecí napětí, a to zejména při vysokém zatížení procesoru zařízení. Pro tento účel bylo sestaveno jednoduché měřicí pracoviště, sestávající se z laboratorního zdroje Manson HCS-3202, testovaného zařízení a digitálního multimetru Owon B35T. Pomocí laboratorního zdroje bylo dodáváno napětí 12 V pro napájecí zdroj a současně pomocí multimetru měřeno napětí na pinech 2 a 6 GPIO konektoru. Na vývojové desce byl následně spuštěn operační systém a dále navržený algoritmus, který zatížil procesor zařízení na více než 65 %. Během několika provedených zkoušek nedošlo k poklesu napětí na měřených svorkách pod hodnotu 4,964 V. Lze tedy usuzovat, že navržený způsob napájení zařízení je pro daný účel použití dostačující.

3.4 Akustická signalizace

Pro varování řidiče před nebezpečím při detekci vybočení vozidla z jízdního pruhu je nutné zařízení opatřit vhodným akustickým zařízením. Jako dostačující se v tomto případě jeví použití standardního piezoměniče, jehož parametry vyhovují uvedenému účelu použití. Z uvedených důvodů byl vybrán piezoměnič PT-1540PM-PQ o rozměrech 17x7 mm s rezonanční frekvencí 4 KHz. Vybraný piezoměnič je napájen napětím o hodnotě 5 V a napájecím proudem maximálně 1,5 mA, což vyhovuje proudové zatížitelnosti výstupních pinů Raspberry Pi na GPIO sběrnici, která činí 8 mA.

Uvedený piezoměnič je připojen pomocí dvoupólového konektoru do 40pinové GPIO sběrnice. Na sběrnici se nachází celkem 4 piny (GPIO12, GPIO13, GPIO18 a GPIO19), které podporují hardwarové generování pulzně šířkové modulace (PWM) potřebné pro generování tónu v piezoměniči. Z hlediska umístění je výhodné použití pinu GPIO12, jelikož sousedí s pinem GND.

3.5 Návrh boxu pro zařízení

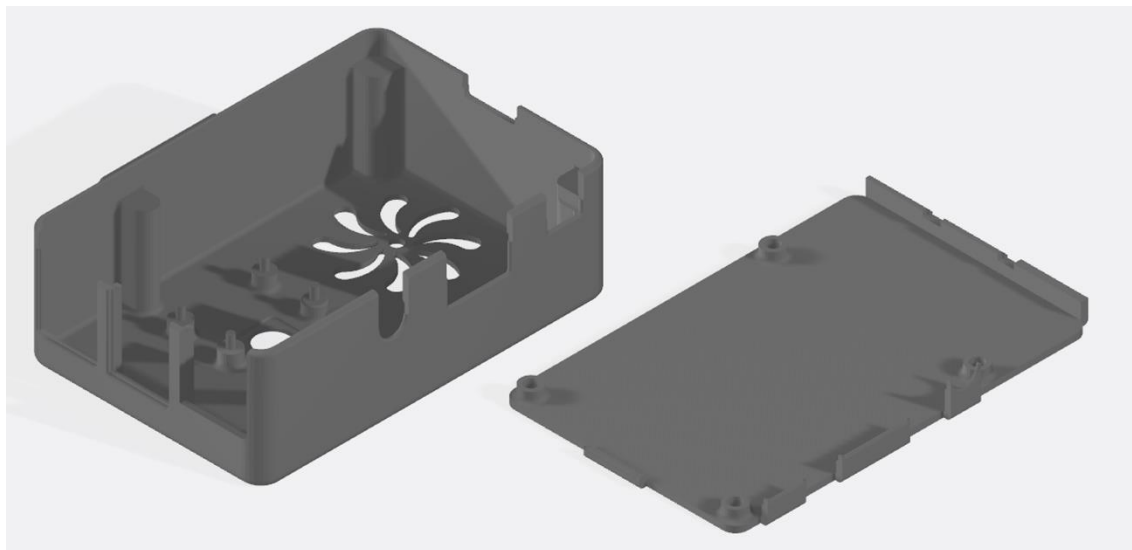
Vzhledem k tomu, že navržené zařízení musí tvořit jeden celek o co nejmenších rozměrech tak, aby bylo pohodlně umísitelné na čelní sklo vozidla v souladu s platnou legislativou, vznikla potřeba provedení individuálního návrhu krabičky pro Raspberry Pi spolu s kamerovým modulem. Navrženou krabičku bude možné spojit s nastavitelnou přísavkou pro vhodné umístění na čelním skle. Pro tento typ prototypové výroby se jeví jako ideální nyní velmi populární metoda 3D tisku. Pomocí programu Autodesk Fusion 360, jehož licence je pro studijní účely volně dostupná, byl dle výše specifikovaných požadavků navržen podrobný 3D model krabičky o finálních rozměrech 89x66x29 mm (viz. Obrázek 3-3). Výkres navrženého modelu je uveden v příloze této práce. Pro tisk vytvořeného modelu byl zvolen materiál PET-G (polyetyléntereftalát-glykol), který disponuje jak dostatečnou mechanickou odolností, tak odolností vůči teplotám do 70° C, což se jeví jako dostačující pro účely testování zařízení ve vozidle.

Tisk prototypu dle zhotoveného návrhu byl proveden na FDM (Fused Deposition Modeling) 3D tiskárně Original Prusa i3 MK3, která disponuje pracovním prostorem o rozměrech 250x210x210 mm a tryskou hotendu o průměru 0,4 mm, což je velikost vhodná spíše pro detailnější modely menších rozměrů. Zpracování navrženého 3D modelu ve formátu *.stl pro tisk na 3D tiskárně (potřebný formát souboru *.GCODE) byl použit program Slic3r Prusa Edition ve verzi 1.41.1. Jedná se o volně dostupný (opensource) software, který je výrobcem upraven pro použití s výše uvedenou 3D tiskárnou.

Dalším důležitým parametrem při 3D tisku je nastavení tzv. výšky vrstvy. Pro tisk z materiálu PET-G je k dispozici výběr výšky vrstvy v rozmezí 0,1 až 0,3 mm. Při volbě nejnižší výšky vrstvy nejsou na vytištěném modelu téměř patrné jednotlivé vrstvy tisku, výrazně však narůstá potřebný počet vrstev a stoupá časová náročnost výroby. Pro tisk navržené krabičky byla zvolena výška vrstvy 0,2 mm. Tato hodnota představuje kompromis mezi rychlostí tisku a výsledným vzhledem modelu. Při použití daného nastavení byla celková doba tisku krabičky rovna přibližně třem hodinám.

S ohledem na možnosti 3D tisku a požadavky na umístění zařízení byla navržena krabička skládající se ze dvou částí. Hlavní část s otvorem na kameru a držákem tištěného spoje kamery uvnitř a dále větracími otvory, aby bylo zajištěno pasivní chlazení vývojové desky Raspberry Pi. Přísavka, pomocí které se zařízení umístí na čelní sklo vozidla, se umísťuje do navržené konzole ve tvaru T, která se nachází na horní stěně krabičky.

Druhou část pak tvoří tvarovaná zadní stěna krabičky s otvory pro samořezné vruty o rozměru 2,5x7 mm, pomocí kterých jsou spojeny obě části, mezi nimiž je současně sevřena vývojová deska Raspberry Pi. Tím je zajištěno pevné a robustní spojení všech částí navrženého zařízení, které je však stále jednoduše rozebíratelné.



Obrázek 3-3 3D model krabičky pro zařízení

4. SOFTWARE

4.1 Operační systém Raspberry Pi

Pro úspěšné oživení a zprovoznění vývojové desky Raspberry Pi (RPi) je nutné nainstalovat vhodný operační systém. Oficiálním operačním systémem pro všechny modely desek RPi je operační systém Raspbian. Kromě tohoto systému lze na oficiálním webu výrobce [17] nalézt ještě další operační systémy jako jsou Ubuntu MATE, Ubuntu Core, Ubuntu server, Windows 10 IoT Core, PiNet, RISC OS a další. Pro účely této práce byl zvolen operační systém Raspbian. Jedná se o linuxovou distribuci operačního systému, odvozenou ze systému Debian. Jeho hlavní výhodou je, že se jedná o primární a současně nejvyužívanější operační systém pro platformu RPi. Tím je zajištěna kompatibilita použitého hardwaru spolu s použitými aplikacemi. Nespornou výhodou je také široká komunita uživatelů, která tento systém používá, a jejíž zkušenosti se dají při vývoji programu využít.

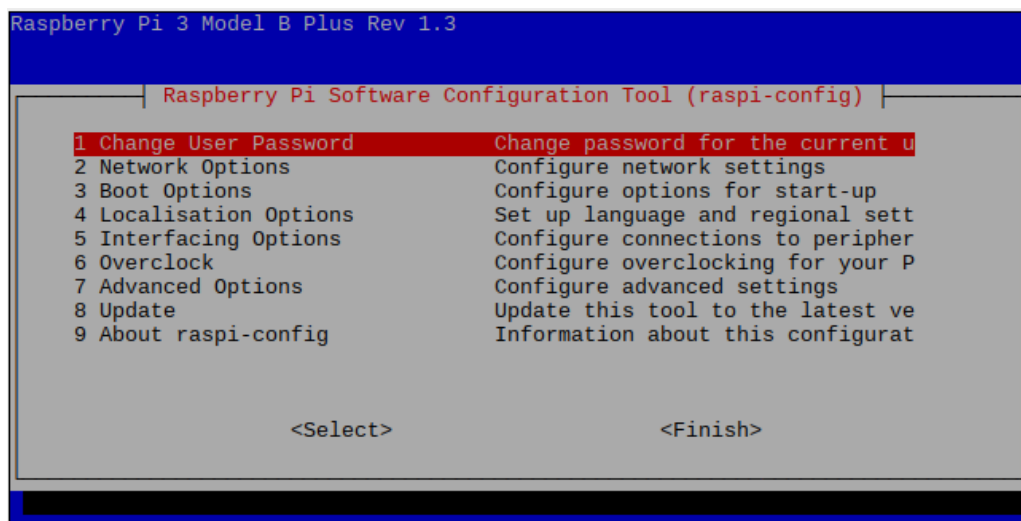
Operační systém RPi se instaluje na microSD kartu, přičemž výrobce doporučuje použití karty o minimální velikosti 8 GB s rychlostí zápisu třídy (class) 6, tedy o maximální rychlosti zápisu 6 MB/s. Pro účely této práce, kdy je nutné prostřednictvím kamery pořizovat záznam testovacích videosekvencí do testovací databáze, byla vybrána paměťová karta výrobce Kingston o kapacitě 32 GB s rychlostí zápisu třídy 10. Pro úspěšnou instalaci systému Raspbian s využitím PC s operačním systémem Windows 10 je nutné provést následující kroky. Prvním z nich je stažení příslušného image souboru (virtuálního obrazu disku) obsahující operační systém RPi z webu výrobce. Pro zápis image souboru na paměťovou kartu je výrobcem doporučeno použití programu *balenaEtcher*, který je k dispozici volně ke stažení pro platformu Windows (32 a 64 bit), Linux a macOS. Program obsahuje jednoduché grafické prostředí, ve kterém je nutné pouze vybrat image soubor určený zápisu na kartu ve formátu *.img nebo *.zip a následně vybrat jednotku, kam je vložena paměťová karta. Po kliknutí na tlačítko „Flash!“ začne program automaticky zapisovat operační systém na paměťovou kartu. Jakmile je zápis dokončen, je nutné paměťovou kartu vložit do příslušného slotu RPi a vývojovou desku oživit.

Pro oživení a úvodní konfiguraci desky RPi je nutné připojit pomocí HDMI kabelu monitor a pro ovládání klávesnicí a myš, které se připojí do USB portu na desce. V závislosti na továrním nastavení nemusí po spuštění RPi dojít k automatickému startu grafického prostředí, nýbrž pouze příkazové řádky. V souladu s pokyny výrobce k instalaci je tedy nutné nejprve zadat přihlašovací údaje (tovární uživatelské jméno: *pi*, heslo *raspberrypi*) a následně spustit konfigurační nástroj RPi pomocí příkazu:

```
> sudo raspi-config
```

V konfiguračním menu (viz. Obrázek 4-1) RPi je možné v nabídce „Boot Options“ nastavit zobrazení grafického prostředí po spuštění systému spolu s automatickým přihlášením uživatele. Dále je nutné aktivovat CSI slot pro připojení kamery, spolu s tím je vhodné povolit službu SSH (Secure Shell) pro vzdálený přístup k RPi. Obě nastavení jsou umístěna v nabídce „Interfacing Options“. Po uložení změn v konfiguraci RPi je nutné provést restart, poté by již mělo nastartovat grafické prostředí systému Raspbian. Po startu systému je vhodné pomocí terminálu provést aktualizaci systému zadáním následujících příkazů:

```
> sudo apt-get update
> sudo apt-get upgrade
```



Obrázek 4-1 Konfigurační menu Raspberry Pi 3

4.2 Vzdálený přístup

Vzhledem k tomu, že zařízení bude používáno ve vozidle, kde nebude možnost připojení monitoru, klávesnice ani myši, je nutné nakonfigurovat vzdálenou správu RPi. Jednou možností je využití protokolu SSH (Secure Shell), který zařízení nativně podporuje, je pouze nutné jej aktivovat v nastavení, jak je popsáno v předchozí kapitole této práce. Služba SSH využívá nativně portu 22. Protokol však umožňuje správu RPi pouze prostřednictvím příkazové řádky, což pro zamýšlený účel použití není dostačující. Z toho důvodu je výhodné mít možnost ovládat desku přes takzvanou vzdálenou plochu a pracovat tak v grafickém rozhraní systému Raspbian. Pro vzdálenou správu RPi pomocí vzdálené plochy byl použit program VNC (Virtual Network Computing) server. Instalace tohoto programu se provede pomocí terminálu s následujícím příkazem:

```
> sudo install tightvncserver
```

Po dokončení instalace se program spustí následujícím příkazem, přičemž číslo za dvojtečkou identifikuje port, na kterém server naslouchá připojení klienta:

```
> vncserver:1
```

Při prvním spuštění programu je uživatel vyzván k volbě hesla k zabezpečení vzdáleného přístupu. Po nastavení hesla je server spuštěn a připraven k připojení klienta. Problém však je, že po každém startu RPi je nutné spustit VNC server pomocí výše uvedeného příkazu. Řešením je sepsání následujícího jednoduchého skriptu, který zajistí, že po každém startu RPi bude spuštěn VNC server s daným nastavením. Tento skript je nutné umístit do složky `/home/pi/config/autostart`:

```
[Desktop Entry]
Type=Application
Name=TightVNC
Exec=vncserver:1
StartupNotify=false
```

Po úspěšném provedení tohoto kroku je již možné spravovat RPi vzdáleně pomocí programu VNC Viewer, který je k dispozici pro různé operační systémy. K připojení je nutné znát IP adresu RPi, použitý port, na kterém server běží, a také zvolené heslo pro vzdálenou správu. Pro případ, kdy nebude v dosahu lokální bezdrátová síť pro spojení PC a RPi je vhodné použít přímé spojení obou zařízení pomocí metalického síťového kabelu (UTP/STP). Za tímto účelem je nutné přidělit RPi statickou IP adresu pro rozhraní `eth0`, tedy ethernet port na desce. Toto nastavení je možné provést úpravou souboru `dhcpcd.conf`, který se nalézá ve složce `/etc/`. K tomu lze využít například integrovaný textový editor GNU nano a to následujícím příkazem:

```
> sudo nano /etc/dhcpcd.conf
```

4.3 Kamera Raspberry Pi

Kamerový modul RPi, který bude použit pro účely této práce a jehož parametry byly popsány v kapitole 3.2, je připojen do CSI sběrnice na vývojové desce. Pokud je kamerový modul korektně aktivován v nastavení, jenž bylo popsáno výše, je možné jej vyzkoušet zadáním jednoduchého příkazu v terminálu, který pořídí jeden snímek s názvem `still_test.jpg` a uloží jej do aktuální složky:

```
> raspistill -o still_test.jpg
```

V případě úspěchu v podobě pořízení testovacího snímku, lze otestovat pořízení krátkého videozáznamu pomocí kamery. Následující příkaz pořídí záznam o délce 5 s v rozlišení 1280x720 pixelů, ve formátu `*.h264` a uloží jej do aktuální složky pod názvem `video_test.h264`:

```
> raspivid -t 5000 -w 1280 -h 720 -o video_test.h264
```

Kromě příkazové řádky lze ke kameře přistupovat také pomocí programovacího jazyka Python (verze 2.7 a 3.2) a knihovny *picamera* v aktuální verzi 1.13.

4.4 Knihovna OpenCV

Pro účely této práce bylo rozhodnuto o použití knihovny OpenCV (Open Source Computer Vision Library) [19]. Jedná se o multiplatformní, svobodnou a otevřenou knihovnu pro manipulaci s obrazem. Tato knihovna obsahuje více než 2500 užitečných funkcí a algoritmů sloužících pro transformaci obrazu, dále kompletní analýzu, filtrování, převod formátů či detekci objektů v obraze. Jeho nespornou výhodou je jednoduchá komunikace se standardním hardwarem, ať už se jedná o webkamery či kamerový modul RPi. Knihovna je volně k použití pod BSD licencí a lze ji využít v programovacích jazycích C, C++, Python, Java a MATLAB, a to pod operačními systémy Windows, Linux, Android a Mac OS. OpenCV má modulární strukturu a skládá se z následujících základních modulů:

- Core functionality (core) – Modul definující základní datové struktury včetně vícerozměrného pole *Mat* a dále základní funkce, které jsou použity ostatními moduly.
- Image Processing (imgproc) – Modul pro zpracování obrazu zahrnující lineární a nelineární filtrování, geometrické transformace obrazu, konverze barevných formátů, histogramy a další.
- Video Analysis (video) – Modul pro analýzu videa obsahující pohybovou predikci, analýzu pozadí a algoritmy pro sledování objektů.
- Camera Calibration and 3D Reconstruction (calib3d) – Modul obsahující algoritmy pro vícekamerovou geometrii, kalibraci jedné či stereo kamery a dále prvky 3D rekonstrukce.
- 2D Features Framework (features2d) – Modul zahrnující nejvýznamnější prvky detektorů a dále deskriptory.
- Object Detection (objdetect) – Modul pro detekci objektů definovaných tříd (obličejů, očí, lidí, vozidel, ...).
- High – level GUI (higui) – Modul obsahující jednoduché prostředí pro vytvoření GUI – uživatelského rozhraní.
- Video I/O (videoio) – Modul obsahující funkce a kodeky pro pořizování snímků a videa.

Pro účely vývoje algoritmu pro tuto práci byla zvolena knihovna OpenCV ve verzi 3.4.6, pro její instalaci na zařízení RPi je nutné nejprve provést instalaci potřebných dodatečných součástí pomocí příkazu:

```
> sudo apt-get install build-essential cmake unzip pkg-config
```

A dále:

```
> sudo apt-get -y install libjpeg-dev libpng-dev libtiff-dev
libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
libxvidcore-dev libx264-dev libgtk-3-dev
```

Po úspěšném dokončení instalace dodatečných součástí je nutné stáhnout požadovanou verzi knihovny pomocí příkazu:

```
> wget -o opencv.zip
https://github.com/opencv/opencv/archive/3.4.6.zip
> wget -o opencv_contrib.zip
https://github.com/opencv/opencv\_contrib/archive/3.4.6.zip
```

Následuje rozbalení archivu a příprava složek pro kompilaci knihovny:

```
> unzip opencv.zip
> unzip opencv_contrib.zip
> cd opencv
> mkdir build
> cd build
```

Po tomto kroku je možné pomocí kompilátoru CMake přistoupit k nastavení a následné kompilaci knihovny pomocí příkazu:

```
> $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/.opencv_contrib/modules \
-D ENABLE_NEON=ON \
-D ENABLE_VFPV3=ON \
-D BUILD_TESTS=OFF \
-D OPENCV_ENABLE_NONFREE=ON \
-D INSTALL_PYTHON_EXAMPLES=OFF \
-D BUILD_EXAMPLES=OFF ..
> cmake ../
```

Kompilace knihovny dle výše uvedeného nastavení trvá na použité desce RPi přibližně dvě hodiny. Po úspěšném dokončení kompilace zbývá knihovnu již jen nainstalovat, což se provede zadáním následujících příkazů. Po jejich dokončení je knihovna připravena k použití a plně funkční.

```
> sudo make install
> sudo ldconfig
```

4.5 Vývojové prostředí

Hlavní aplikace pro detekci jízdních pruhů na vozovce a vybočení vozidla byla zpočátku vyvíjena na PC bez použití vývojové desky RPi. To je výhodné zejména díky vyššímu výpočetnímu výkonu PC a vyšší kapacitě operační paměti, což výrazně urychluje kompilace a ladění výsledného programu. Vzhledem k tomu, že PC neobsahuje kamerový modul, ani GPIO konektor, bylo nutné finální odladění programu provést přímo na desce RPi.

Pro vývoj aplikace na PC s operačním systémem Microsoft Windows 10 bylo použito vývojové prostředí Visual Studio 2017. Pro toto vývojové prostředí je možné stáhnout připravenou knihovnu OpenCV, která již obsahuje zkompileovaný build. Integraci knihovny do vývojového prostředí je pak možné jednoduše provést zadáním cesty ke knihovním souborům, pro vybraný překladač a sestavovací program (linker).

Pro finální doladění a testování navržené aplikace přímo na desce RPi bylo rozhodnuto o použití volně dostupného multiplatformního vývojového prostředí Code::Blocks ve verzi 17.12. Jedná se o komplexní vývojové prostředí určené pro jazyk C a C++. Instalace vývojového prostředí se provede zadáním jednoduchého příkazu v terminálu:

```
> sudo apt-get install codeblocks
```

Integrace knihovny OpenCV, jejíž kompilace a instalace na vývojovou desku RPi byla popsána v kapitole 4.4, do vývojového prostředí Code::Blocks se provede zápisem následujícího řetězce v nabídce nastavení „*Project build options*“, na kartě „*Linker settings*“:

```
`pkg-config -cflags -libs opencv`
```

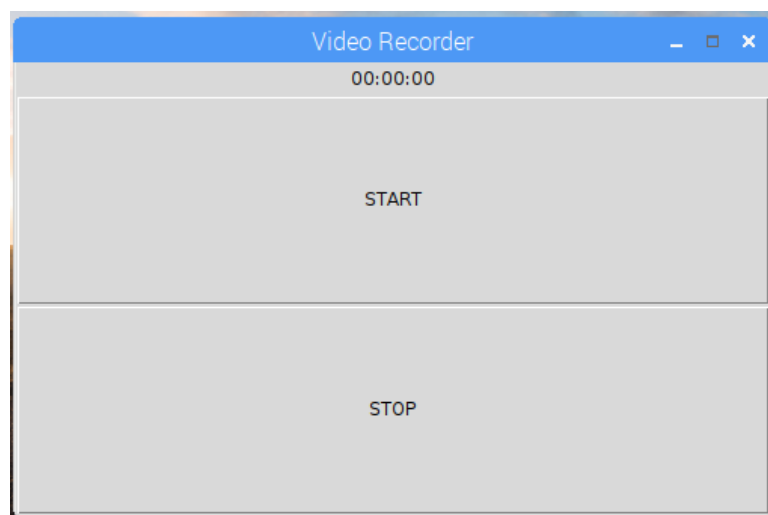
4.6 Test knihovny OpenCV

Za účelem ověření funkčnosti a správného nastavení knihovny OpenCV spolu s kamerovým modulem připojeným k desce RPi byl ve vývojovém prostředí Code::Blocks vytvořen jednoduchý program v jazyce C++. Tento program, jehož zdrojový kód je uveden v příloze této práce, po svém spuštění inicializuje připojenou kameru, otevře nové okno a zobrazí živý náhled videa z kamerového modulu. Po stisknutí jakékoli klávesy dojde k automatickému ukončení programu. Pokud se inicializace kamerového modulu nepodaří, je vypsána chybová hláška. V takovém případě je nutné ověřit, zda je kamerový modul správně připojen a zda kamera není aktuálně využívána jiným procesem.

4.7 Program Recorder.py

Pro účely testování funkčnosti a spolehlivosti programu detekujícího vybočení z jízdního pruhu bylo nutné pořízení sady testovacích videosekvencí zachycených prostřednictvím vývojové desky RPi umístěné na čelním skle ve vozidle. Pro tyto účely byl v programovacím jazyce Python vytvořen program s názvem Recorder.py. Zdrojový kód tohoto programu je uveden v příloze této práce. Programovací jazyk Python byl zvolen s ohledem na to, že v rámci systému Raspbian jsou k dispozici všechny potřebné součásti a knihovny, které jsou pro účel tohoto pomocného programu zapotřebí. Dalším důvodem je jednoduchost při vytváření uživatelského rozhraní (GUI) pomocí nativní knihovny *tkinter*. Cílem bylo vytvoření programu, kde je možné v jednoduchém uživatelském rozhraní pořizovat videosekvence a sledovat jejich stopáž.

Navržený program využívá kromě uvedené knihovny uživatelského rozhraní *tkinter* a dále knihovnu *picamera* pro ovládání kamerového modulu. Pro odečítání času nahrávání a automatické generování názvu videa dle aktuálního data a času knihovny *time* a *datetime*. Výsledný vzhled uživatelského rozhraní navrženého programu ilustruje Obrázek 4-2. Princip funkce programu je jednoduchý – po kliknutí na tlačítko „*START*“ dojde k inicializaci kamerového modulu a nastavení parametrů záznamu videa na rozlišení 1280x720 pixelů a snímkové rychlosti 24 snímků za vteřinu. Ve složce */home/pi/Videos/* je následně vytvořen soubor s názvem ve tvaru: „*ROK_DEN_MĚSÍC_HODINA-MINUTA-VTEŘINA*“ s koncovkou **.h264*. Současně se zahájením nahrávání videa do vytvořeného souboru se rozběhne časomíra v horní části okna, na základě které je možné sledovat stopáž dané sekvence. Po kliknutí na tlačítko „*STOP*“ je záznam ukončen a zastavena časomíra. K vynulování časomíry dojde při dalším spuštění nahrávání tlačítkem „*START*“.



Obrázek 4-2 Uživatelské rozhraní programu Recorder.py

Aby byl navržený skript spustitelný v operačním systému Raspbian, musí obsahovat takzvaný *Shebang* neboli řádek na začátku skriptu, pomocí kterého je vytvořený skript interpretován. V řádku je uvedena cesta a název programu, kterým má být skript interpretován v tomto případě:

```
#!/usr/bin/env python
```

Dále je nutné pomocí příkazu *chmod* (zkratka change mode) změnit přístupová práva tak, aby byl skript spustitelný – to je provedeno zadáním následujícího příkazu v terminálu:

```
> sudo chmod +x Recorder.py
```

Vzhledem k tomu, že navržená aplikace bude spouštěna pomocí laptopu ve vozidle, je vhodné si vytvořit zástupce programu na pracovní ploše, aby byla aplikace snadno dostupná. Za tímto účelem je nutné vytvoření nového souboru s příponou *.desktop ve složce */home/pi/Desktop*. V tomto souboru je nutné definovat název vytvořeného zástupce, cestu k ikoně aplikace a cestu ke spustitelnému skriptu. Obsah takového souboru může být následující:

```
[Desktop Entry]
Name=Recorder
Comment=Testing sequence recorder
Icon=/usr/share/pixmaps/openbox.xpm
Exec=/home/pi/Recorder.py
Encoding=UTF-8
Type=Application
Terminal=false
Categories=none
```

5. LDWS ALGORITHMUS

Při návrhu algoritmu je vhodné nejprve definovat podmínky, za kterých bude tento funkční. Účelem algoritmu je detekce vybočení vozidla z jízdního pruhu a následné akustické upozornění řidiče tak, aby mohl na rizikovou situaci adekvátně reagovat. Vybočení vozidla je detekováno prostřednictvím obrazového signálu kamery, která je umístěna na čelním skle vozidla a kontinuálně snímá prostor před vozidlem. Vzhledem k principu funkce algoritmu je jasné, že jeho funkčnost bude ovlivněna několika základními podmínkami, a to nejen světelnými, ale také výhledem kamery, přítomností a stavem vodorovného dopravního značení, ale také faktem, zda je vozovka zatáčkovitá či naopak rovinatá.

Na základě výše uvedených podmínek byl algoritmus navržen tak, aby byl funkční především na dálnicích a silnicích I. třídy, kde je zpravidla přítomno kvalitní a neporušené vodorovné dopravní značení jízdních pruhů. Výhoda tkví také v tom, že charakter těchto cest je převážně rovinatý a případné zatáčky mají velký poloměr. Toto tvrzení platí z části také pro silnice II. třídy. Z hlediska detekce jízdních pruhů jsou nejproblematictější silnice III. třídy a dále takzvané místní a účelové komunikace, kde často jízdní pruhy nejsou vůbec vyznačeny, a pokud ano, bývají poškozené či vybledlé. V takovém případě se řidič nemůže spoléhat ani na pomoc komerčních asistenčních systémů implementovaných do moderních vozidel. Jedinou možností detekce v tomto případě by bylo prahování barvy vozovky, která je odlišná od zbarvení krajnice, a tak detekovat případné vybočení z tělesa vozovky.

Dalším faktorem ovlivňujícím funkci algoritmu je viditelnost. Je-li snížena, např. za šera či tmy, je nutné počítat s jeho omezenou funkcí. V případě zvlášť nepříznivých podmínek, jako je mlha, sněžení či hustý déšť, je nutné počítat s tím, že varovný systém nebude funkční. Vzhledem k tomu, že scéna před vozidlem je snímána skrze čelní sklo, je nutné za běžných podmínek udržovat dobrý výhled. Znečištěné, zamlžené či namrzlé čelní sklo může výrazně omezit, nebo dokonce znemožnit správnou funkci algoritmu.

Při výběru vhodného algoritmu detekce jízdních pruhů je kladen důraz především na vysokou spolehlivost detekce a nízkou výpočetní náročnost, což jsou požadavky téměř protichůdné. Je tedy nutné zvolit vhodný kompromis, který bude vyhovovat výpočetnímu výkonu navrženého hardwaru a současně bude dostatečně spolehlivý a robustní.

5.1 Popis fungování algoritmu

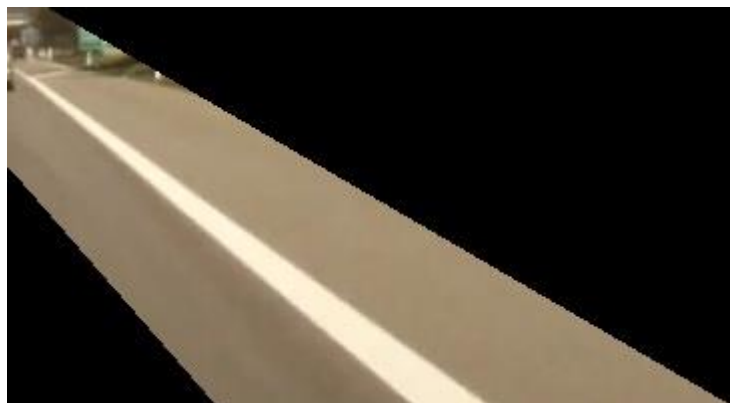
Na základě teoretické rešerše a experimentálního ověření funkce dostupných algoritmů bylo rozhodnuto o konečném výběru kombinace využití Cannyho hranového detektoru a následné Houghovy transformace pro detekci jízdních pruhů. Detailní princip fungování navrženého algoritmu ilustruje blokový diagram (Obrázek 5-11).

Veškeré funkce algoritmu, které byly vytvořeny pro zpracování obrazu, detekci jízdních pruhů a vybočení vozidla jsou součástí knihovny *Processing*. Soupis všech funkcí uvádí Tabulka 5-1.

Tabulka 5-1 Soupis funkcí knihovny *Processing*

Název funkce	Popis činnosti funkce
<code>colorTresholding</code>	Prahování bílé barvy na vozovce
<code>cluster_lines</code>	Třídění a sloučení detekovaných linií
<code>discard_lines</code>	Odstranění falešně pozitivních výsledků
<code>draw_lines</code>	Vykreslení detekovaných linií
<code>get_left_line</code>	Detekce levého jízdního pruhu
<code>get_means</code>	Načtení průměrné hodnoty z bufferu
<code>get_right_line</code>	Detekce pravého jízdního pruhu
<code>lane_departure</code>	Detekce vybočení vozidla
<code>limit_left_road_part</code>	Ořez ROI levého pruhu
<code>limit_right_road_part</code>	Ořez ROI pravého pruhu
<code>processImage</code>	Hlavní funkce
<code>update_line_parameters</code>	Uložení detekovaných hodnot do bufferu

Po spuštění algoritmu a úspěšném načtení zdrojového obrazu, jehož zdrojem může být statický snímek, uložená videosekvence či živý náhled z připojené kamery, je nejprve ve funkcích `get_right_line` a `get_left_line` vyříznuta požadovaná oblast zájmu (ROI). V té je očekávána přítomnost čar ohraničujících jízdní pruh. Velikost a umístění jednotlivých oblastí zájmu byly experimentálně zjištěny a poloha těchto regionů je v programu nastavena fixně. Pro snížení pravděpodobnosti falešné detekce a snížení výpočetní náročnosti algoritmu, je uvnitř každé oblasti ROI proveden ořez obrazu, pomocí funkcí `limit_left_road_part` a `limit_right_road_part` dle stanovených bodů. Výsledkem jsou připravené dva snímky ROI pro detekci dělicí (viz. Obrázek 5-2) a vodící (viz. Obrázek 5-1) čáry.



Obrázek 5-1 Ukázka ROI dělicí čáry



Obrázek 5-2 Ukázka ROI vodící čáry

Uvnitř každé oblasti je následně provedena detekce hran za použití Cannyho hranového detektoru. Pro zvýšení robustnosti algoritmu je paralelně prováděna rozšířená detekce prostřednictvím prahování bílé barvy v histogramu jednotlivých barevných složek obrazu. Vstupní barevný snímek je rozdělen do jednotlivých R (červená), G (zelená), B (modrá) složek s tím, že pro každou složku je vypočten histogram. Předpoklad je takový, že vodorovné značení je typické nejvyšším podílem bílé barvy v daném obraze. V každém histogramu je následně hledána první špičková hodnota. Tato maximální hodnota reprezentuje oblast s nejvyšší intenzitou bílých pixelů v daném obraze. Hodnota je nalezena, pokud splňuje podmínky:

1. $H(k) > 0$,
2. $H(k) > TV$,
3. $H(k) - 7 < RS$,
4. $H(k) - 7 < LS$,
5. $H(k) > TR$,

Kde $H(k)$ označuje četnost prvku v daném histogramu, TV označuje maximální zjištěnou četnost, TR je prahová hodnota označující minimální četnost hledané špičky a RS označuje sumu vypočtenou dle vztahu (5.1):

$$RS = \frac{1}{5} \sum_{k+1}^{k+5} H(k) \quad (5.1)$$

Suma LS je vypočtena z obdobného vztahu (5.2):

$$LS = \frac{1}{5} \sum_{k-1}^{k-5} H(k) \quad (5.2)$$

Dle prvotních experimentů při vývoji algoritmu bylo zjištěno, že pro korektní funkci prahování bílé barvy je nutné nalezení oblasti lokálního minima v histogramu. Lokální minimum je nalezeno dle následujícího pravidla:

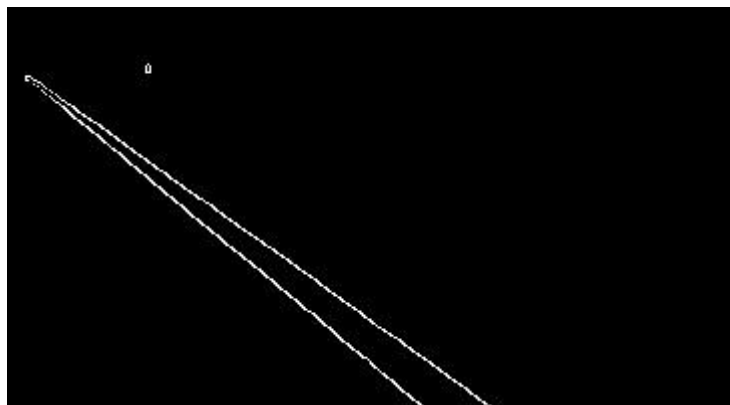
1. Musí být nalezena hodnota lokálního maxima
2. $H(k) - 9 < LS$

Jakmile jsou známy hodnoty obou prahů, je obraz převeden do binarizované podoby. Transformace probíhá tak, že všem pixelům, jejichž četnost v histogramu je vyšší než hodnota lokálního minima a současně menší než hodnota lokálního maxima +40, je přiřazena hodnota 1, všem ostatním pak hodnota 0. Uvedeným způsobem jsou vytvořeny binarizované obrazy pro každou barevnou složku a následně sloučeny do jednoho výsledného obrazu prostřednictvím logického operátoru AND. Takto získaný binarizovaný obraz (viz. Obrázek 5-3), ve kterém hodnota 1 označuje bílou barvu detekovaného dopravního značení, je přiveden na vstup Cannyho hranového detektoru.



Obrázek 5-3 Binarizovaný obraz pro detekci hran

Ve druhém, paralelním zpracování je barevný obraz převeden do režimu stupňů šedi (pomocí funkce `CV_RGB2GRAY`) a přiveden na vstup Cannyho detektoru hran. Výsledkem jsou dva binarizované obrazy s detekovanou čarou dopravního značení na vozovce: jeden pochází z funkce prahování bílé barvy (Obrázek 5-4), druhý ze snímku ve stupních šedi (Obrázek 5-5). Na oba snímky zvlášť je v následujícím kroku aplikována Houghova transformace, jejíž výsledky jsou posléze sloučeny do jednoho snímku.



Obrázek 5-4 Detekované hrany z binarizovaného snímku



Obrázek 5-5 Detekované hrany z obrazu ve stupních šedi

Na základě provedené Houghovy transformace je získáno mnoho parametrů detekovaného značení na silnici. Zpravidla je však nalezeno více detekovaných čar, cílem je proto zjednodušení na jednu finální linii. K tomu slouží funkce `cluster_lines`. Ta provádí výpočet srovnání úhlů a posuvů všech detekovaných čar. Pokud je zjištěn rozdíl úhlu menší než 5° a hodnota vzájemného posunutí nižší než 10 pixelů, jsou čáry považovány za shodné. Všechny ostatní výsledky, které nevyhovují uvedeným parametrům, jsou označeny za falešně pozitivní a nejsou dále zpracovávány, jejich odstranění provádí funkce `discard_lines`. Tato funkce se také stará o odstranění falešně pozitivních linií, které tvoří ořezové linie v obrazu ve stupních šedi (Obrázek 5-5). V případě úspěšné detekce dělicí i vodící čáry, jsou obě detekované linie vykresleny do původního snímku prostřednictvím funkce `draw_lines` (viz. Obrázek 5-6).



Obrázek 5-6 Vykreslení detekovaných čar do původního snímku

Vzhledem k vysoké dynamičnosti snímané scény před vozidlem a možnosti skokové změny podmínek detekce způsobených například razantní změnou světelných podmínek, dočasnou absencí dopravního značení na silnici či průjezdem zatáčkou s příliš velkým poloměrem, může dojít k náhlé ztrátě detekovaných linií a tím k dočasnému výpadku systému. Ve snaze o řešení tohoto problému byl v algoritmu navržen cyklický buffer, který uchovává parametry posledních deseti úspěšně detekovaných linií. V případě, že dojde k výpadku a linie nejsou v aktuálním snímku detekovány, je použit vážený průměr z nejnovějších dat. Tento vážený průměr je vypočten dle následujícího vztahu:

$$l_{0,x} = \frac{w_1 l_{1,x} + w_2 l_{2,x} + \dots + w_{10} l_{10,x}}{\sum_{i=1}^{10} w(i)} \quad (5.3)$$

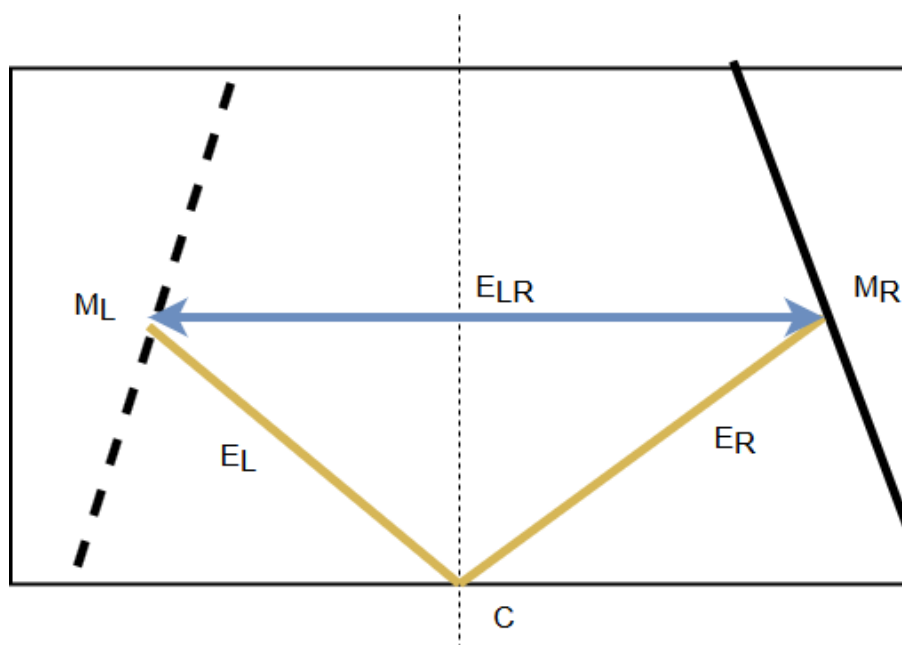
Kde $l_{0,x}$ označuje parametr čáry (posun případně úhel), w_i označuje váhu i -tého parametru čáry a $l_{i,x}$ uchovává parametry uložené v bufferu z předchozích detekcí. Váha jednotlivých parametrů v bufferu je stanovena tak, aby nejnížší váha byla přiřazena nejstarším parametrům v bufferu:

$$w_i = \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.2, 0.1, 0.05, 0.05\} \quad (5.4)$$

Aby bylo maximálně sníženo riziko chybné detekce, jsou vždy aktuálně získaná data porovnávána s obsahem bufferu a v případě, že tyto parametry nekorespondují, tj. aktuálně detekovaná linie má výrazně odlišné parametry než předchozí, pak nejsou aktuální data do bufferu zapsána.

Při úspěšné detekci obou čar na vozovce hlídá funkce `lane_departure` vybočení vozidla z jízdního pruhu. Princip této detekce ilustruje Obrázek 5-7. Počátek v bodě C je umístěn přibližně do středové osy vozidla tak, aby se nalézal přibližně uprostřed jízdního pruhu. Body M_L a M_R označují středový bod aktuálně detekovaných jízdních pruhů. Proměnné E_L a E_R označují euklidovské vzdálenosti mezi počátkem a bodem uprostřed detekované čáry. Horizontální vzdálenost mezi středy obou jízdních pruhů reprezentuje hodnota E_{LR} . Dalšími proměnnými tohoto algoritmu jsou experimentálně stanovené prahové hodnoty T_R , T_L a T_{LR} . Výsledný princip detekce vybočení vozidla je následující:

- Levé vybočení je detekováno, pokud $E_L < E_R$ a současně $E_L < T_L$
- Pravé vybočení systém detekuje, pokud $E_R < E_L$ a současně $E_L < T_L$
- V případě, že platí $E_{LR} < T_{LR}$ algoritmus detekuje opuštění jízdního pruhu



Obrázek 5-7 Princip detekce vybočení vozidla

Pro účely vývoje a vyhodnocení úspěšnosti algoritmu při detekci vybočení vozidla, jsou do výsledného obrazu vykresleny dva čtverce, které signalizují pravé a levé vybočení vozidla z jízdního pruhu. Pokud se vozidlo nachází uvnitř jízdního pruhu, jsou čtverce zbarveny zeleně (viz. Obrázek 5-8). V případě detekce levého nebo pravého vybočení se příslušný čtverec zbarví červenou barvou. Pokud je detekována změna jízdního pruhu, je barva obou čtverců žlutá.



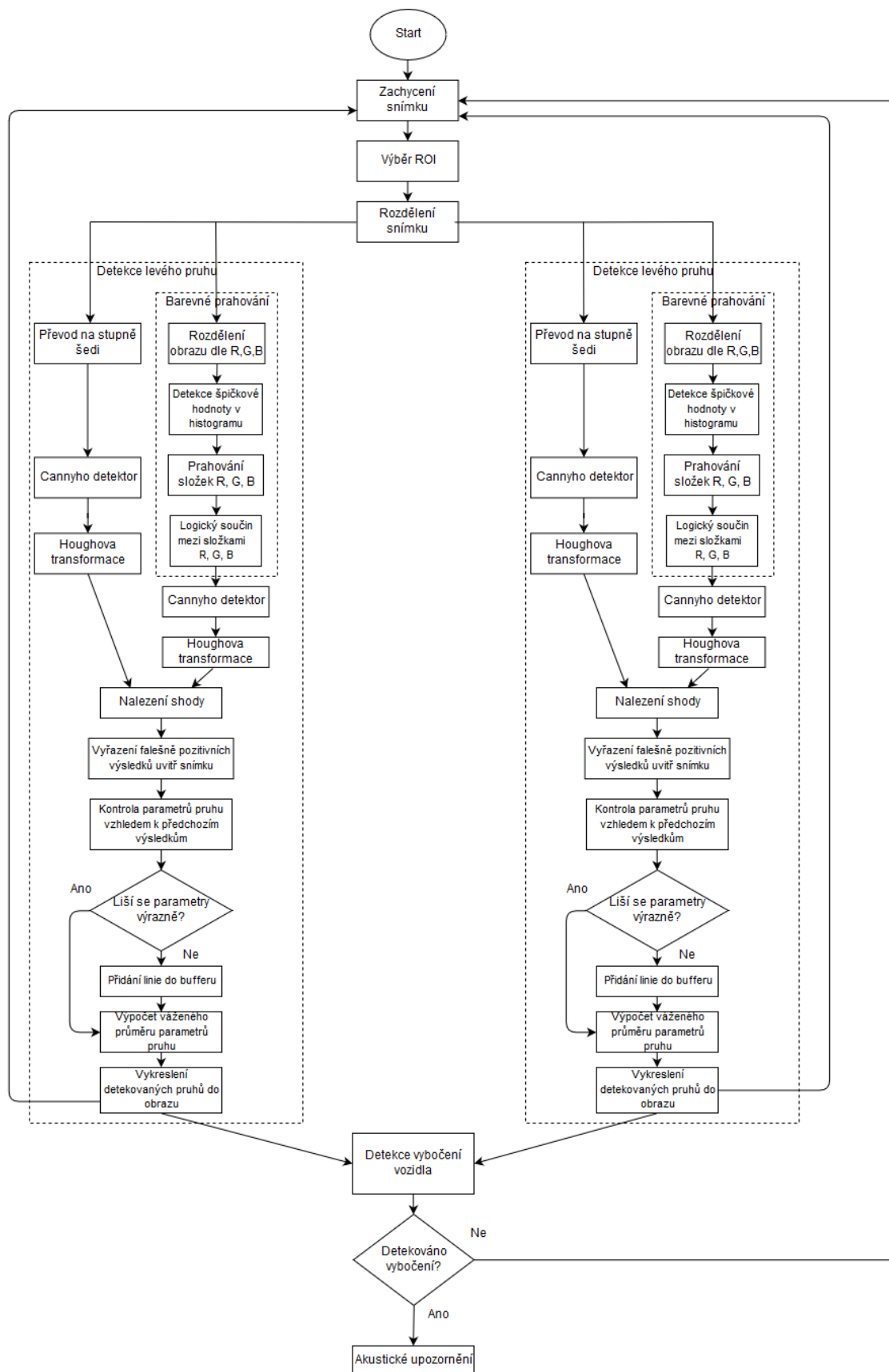
Obrázek 5-8 Detekce vozidla v pruhu



Obrázek 5-9 Ukázka detekce vybočení vlevo



Obrázek 5-10 Ukázka detekce vybočení vpravo



Obrázek 5-11 Blokový diagram navrženého algoritmu

6. VYHODNOCENÍ SPOLEHLIVOSTI ALGORITMU

6.1 Testovací databáze

Za účelem testování a vyhodnocení funkčnosti navrženého algoritmu byla pořízena databáze testovacích videosekvencí. Průměrná délka jednotlivých sekvencí činí 30 vteřin. Rozlišení videosekvencí je rovno 1280x720 pixelům (720p) při snímkové rychlosti 25 snímků za vteřinu. Video je komprimováno prostřednictvím kodeku h.264, který využívá hardwarové akcelerační zařízení a je vhodným kompromisem mezi potřebným výpočetním výkonem a nízkou bitovou rychlostí výsledných sekvencí. Videosekvence pro testovací databázi byly pořízeny pomocí navrženého zařízení s kamerovým modulem, které bylo umístěno na čelním skle vozidla. Pro přístup k ovládání zařízení a zobrazení živého náhledu byl ve vozidle umístěn také notebook, který byl s vývojovou deskou propojen krátkým metalickým síťovým STP kabelem, opatřeným konektory 8P8C (RJ – 45). Přístup k operačnímu systému Raspbian byl získán prostřednictvím aplikace pro vzdálenou plochu VNC Viewer, který se připojí k serveru běžícímu na RPi. Nahrávání testovacích videosekvencí je následně provedeno prostřednictvím aplikace Recorder, jejíž princip je popsán v kapitole 4.7.

Testovací sekvence byly pořízeny na základě předem vybraných kvalitativních parametrů; jednalo se o kvalitu vodorovného dopravního značení, různé světelné podmínky a také různé třídy pozemních komunikací. Přesný rozpis jednotlivých parametrů uvádí Tabulka 6-1. Pro každý kvalitativní parametr byly pořízeny alespoň 3 testovací sekvence o délce minimálně 30 vteřin, celkem tak bylo pořízeno 27 videosekvencí. Všechny testovací sekvence byly pořízeny na jihomoravských silnicích v první polovině roku 2019. Sekvence pro kategorii dálnic a silnic pro motorová vozidla byly pořízeny na dálnici D1 a na silnici pro motorová vozidla č. 43. Pro kategorii silnice I. třídy byla využita silnice č. 43 (E461). Kategorii silnic II. třídy reprezentují sekvence pořízené na silnici č. 385 a 379.

Tabulka 6-1 Návrh parametrů testovací databáze

Kvalita dopravního značení	Kvalitní, vysoce kontrastní jízdní pruhy
	Starší, zašlé jízdní pruhy
	Poškozené jízdní pruhy
Světelné podmínky	Jasno
	Zataženo
	Šero
Třída pozemní komunikace	Dálnice a silnice pro motorová vozidla
	Silnice prvních tříd
	Silnice druhých tříd

6.2 Vyhodnocení spolehlivosti algoritmu

Pro účely vyhodnocení spolehlivosti detekce vybočení vozidla z jízdního pruhu, prostřednictvím navrženého zařízení a vytvořeného algoritmu bylo analyzováno více než 700 snímků z databáze pořízených videosekvencí. V každém snímku jsou vykresleny detekované jízdní pruhy a indikátory, které označují, zda se vozidlo nachází v jízdním pruhu, nebo zda vybočuje. Snímky pro vyhodnocení byly vybírány tak, aby postihly všechny sledované kvalitativní parametry a současně bylo analyzováno každé provedené vybočení vozidla, jehož záznam byl pořízen. Každý z vybraných snímků byl následně analyzován a posouzen, zda se jedná o korektní detekci dopravního značení na vozovce, případně korektní detekci vybočení vozidla. Získaná data byla dále rozdělena dle třídy pozemní komunikace. Z těchto dat byla následně vypočtena procentuální úspěšnost detekce, jejíž výsledky uvádí tabulka.

Tabulka 6-2 Úspěšnost detekce pro jednotlivé třídy silnice

Třída pozemní komunikace	Úspěšnost detekce
Dálnice a silnice pro motorová vozidla	94 %
Silnice I. třídy	87 %
Silnice II. třídy	72 %

Z dosažených výsledků je patrné, že nejvyšší spolehlivosti detekce vybočení dosahuje zařízení na dálnicích a silnicích pro motorová vozidla. Hlavním důvodem je zpravidla celistvé dopravní značení, které je pravidelně obnovováno tak, aby nedošlo k jeho zániku. Dalším faktorem je dostatečná šířka jízdního pruhu a nízký počet zatáček.

Detekce je na tomto typu pozemní komunikace natolik robustní, že vliv sledovaných světelných podmínek byl shledán zanedbatelným.

Analogicky lze hodnotit také úspěšnost detekce na silnicích I. třídy, také zde díky udržovanému dopravnímu značení a dostatečné šířce jízdního pruhu, pracuje zařízení spolehlivě. Nižší hodnocení je v tomto případě způsobeno případy chybné detekce při průjezdu okolo šrafování (viz. Obrázek 6-1), a dále např. chybnou detekcí připojovacího pruhu (viz. Obrázek 6-2), místo požadované vodící čáry. V takovém případě zařízení není schopné detekovat vybočení vozidla z jízdního pruhu.

Výsledky analýzy použití zařízení na silnicích II. třídy reflektují stav vodorovného dopravního značení. Pokud kvalita dopravního značení vyhovuje prvním dvěma sledovaným parametrům (viz Tabulka 6-1) dosahuje úspěšnost detekce podobných hodnot, jako v případě silnic I. třídy. V případě silně poškozeného, či úplně chybějícího značení je pravděpodobnost úspěšné detekce nízká. Dalším faktorem snižujícím výslednou úspěšnost detekce je šířka vyznačeného jízdního pruhu, která je u silnic II. třídy znatelně menší. Toto způsobuje nižší rozhodovací úroveň algoritmu, při vyhodnocení stavu, kdy vozidlo vybočuje z jízdního pruhu. V důsledku toho je nutné počítat s možností falešné detekce vybočení. Posledním vlivem, se kterým je nutné počítat, jsou zatáčky, při jejichž průjezdu vozidlo zpravidla nejede středem jízdního pruhu. Také v tomto případě je nutné počítat s možností falešné detekce vybočení (viz. Obrázek 6-3).



Obrázek 6-1 Ukázka chybné detekce šrafování



Obrázek 6-2 Ukázka chybné detekce připojovacího pruhu



Obrázek 6-3 Ukázka falešně pozitivní detekce vybočení v zatáčce

6.3 Výpočet intervalu spolehlivosti

Vzhledem k tomu, že zkoumaná úspěšnost detekce je náhodnou veličinou s určitým rozdělením, lze získaná data chápat jako realizaci této náhodné veličiny. Z hlediska statistického vyhodnocení získaných výsledků je tedy vhodné provést výpočet intervalu spolehlivosti neboli konfidenčního intervalu. Pro daný účel bude uvažován interval s konfidenční hladinou 95 %, který se vypočte dle vzorce (6.1)[21]:

$$\left\langle P - z_{1-\frac{\alpha}{2}} \sqrt{\frac{P(1-P)}{n}}, P + z_{1-\alpha/2} \sqrt{\frac{P(1-P)}{n}} \right\rangle \quad (6.1)$$

Kde $z_{1-\alpha/2}$ je $1-\alpha/2$ kvantil normálního normovaného rozdělení a P je bodový odhad parametru π . Interval spolehlivosti byl vypočten zvlášť pro každou hodnotu úspěšnosti. Dále byla vyhodnocena celková úspěšnost detekce, a to včetně intervalu spolehlivosti. Výsledné hodnoty pro jednotlivé třídy pozemních komunikací uvádí Tabulka 6-3. Celková výsledná hodnota procentuální účinnosti detekce činí 83 % s konfidenčním intervalem velikosti 0,84.

Tabulka 6-3 Vypočtené intervaly spolehlivosti

Třída pozemní komunikace	Interval spolehlivosti
Dálnice a silnice pro motorová vozidla	1,57
Silnice I. třídy	1,53
Silnice II. třídy	1,3

6.4 Doporučení pro použití zařízení

Ze získaných výsledků testování lze dovodit, že navržené zařízení je vhodné zejména pro použití na dálnicích a silnicích I. třídy, neboť na tomto typu silnic dosahuje vysoké procentuální úspěšnosti detekce. Při použití na silnicích nižších tříd je nutné brát v potaz zvýšené riziko falešné detekce, či absence detekce, a to zejména v místech, kde není na vozovce kvalitní a celistvé vodorovné dopravní značení. Velký vliv na úspěšnou detekci jízdního pruhu, má korektní nastavení ROI, tedy oblastí, kde algoritmus vyhledává dělicí čáru. V navrženém algoritmu je umístění ROI dáno fixně, což klade zvýšené nároky na vhodné umístění zařízení do vozidla a zaměření kamery. Nad rámec této práce by bylo vhodné algoritmus modifikovat tak, aby na základě úspěšně detekovaných čar dynamicky měnil umístění ROI za účelem zvýšení robustnosti detekce. Pro použití na silnicích, kde není vyznačeno dopravní značení by jedinou alternativou mohla být detekce přechodu silnice - krajnice, na základě odlišné barvy či viditelné linie.

7. ZÁVĚR

Předkládaná diplomová práce byla věnována návrhu zařízení, které bude detekovat vybočení vozidla z jízdního pruhu prostřednictvím obrazového signálu získaného z připojeného obrazového snímače. Zařízení bylo nutné navrhnout tak, aby bylo možné jej umístit na čelní sklo vozidla v souladu s platnou legislativou a nerušilo výhled z vozu. Úvodní část práce byla věnována zmapování aktuálních způsobů detekce jízdních pruhů z obrazového signálu, dle dostupné literatury. Jednotlivé techniky byly podrobněji popsány v teoretické části práce. V další části práce byl proveden výběr vhodného hardwaru, na základě stanovených požadavků. Po pečlivém zvážení všech aspektů ovlivňujících funkčnost navrhovaného zařízení byl zvolen nejnovější model vývojové desky Raspberry Pi 3 Model B+, včetně kamerového modulu verze 2, který umožňuje pořizování videosekvencí ve vysokém rozlišení. Vybrané zařízení disponuje také logickou vstupně / výstupní sběrnici, která posloužila k připojení piezoměniče pro akustickou signalizaci. Vzhledem k požadavku, aby zařízení spolu s kamerovým modulem tvořilo jeden kompaktní celek přijatelných rozměrů, byl za tímto účelem navržen prototyp krabičky, který byl následně vytvořen pomocí technologie 3D tisku. Následující část práce byla věnována návrhu algoritmu, který byl implementován v jazyce C++. Jádrem navrženého algoritmu je analýza vybraných částí obrazu prostřednictvím Cannyho hranového detektoru a následné Houghovy transformace.

V experimentální části práce bylo navržené zařízení sestaveno a oživeno tak, aby s jeho pomocí mohla být pořízena databáze krátkých testovacích videosekvencí. Zařízení bylo opatřeno operačním systémem, vývojovým prostředím Code::Blocks a také potřebnými knihovnami pro implementaci algoritmu. Pro účely nahrávání videa byla vytvořena jednoduchá aplikace s uživatelským rozhraním. Testovací databáze byla navržena tak, aby reflektovala různé kvalitativní parametry jako jsou světelné podmínky, kvalita vodorovného dopravního značení a třída pozemní komunikace. Pro každý kvalitativní parametr bylo pořízeno několik testovacích videosekvencí. Pomocí těchto sekvencí bylo v závěru práce provedeno vyhodnocení úspěšnosti detekce vybočení vozidla. Pro účely hodnocení algoritmu je detekované dopravní značení vykresleno do obrazu a pomocí dvou indikátorů je zobrazováno detekované vybočení vozidla z jízdního pruhu. Závěrečná část práce byla věnována procentuálnímu vyhodnocení úspěšnosti detekce vybočení vozidla z jízdního pruhu. Výsledky byly rozděleny do skupin podle tříd pozemních komunikací. Nejvyššího hodnocení úspěšnosti detekce dosahuje zařízení na dálnicích a silnicích pro motorová vozidla, což je způsobeno zejména kvalitou dopravního značení. S tím úzce souvisí i účel použití navrženého zařízení, kterým je využití zejména na dálnicích a silnicích I. třídy. Na silnici II. třídy je nutné již počítat s nižší úspěšností detekce. Na silnicích III. třídy a místních a účelových komunikacích, které často postrádají konzistentní vodorovné značení, je nutné počítat pouze s omezenou funkcí zařízení.

Přes uvedená omezení však navržené zařízení skýtá stále velký potenciál v možném doplnění a rozšíření o další funkce. Pro další práci by bylo vhodné doplnit zařízení dotykovým LCD panelem vhodné velikosti a vytvořit jednoduché uživatelské rozhraní, pro základní konfiguraci zařízení. Kromě detekce vybočení by mohlo být zařízení použito také pro kontinuální záznam scény před vozidlem, pro případ dopravní nehody. Z hlediska detekce a využití knihovny OpenCV by bylo možné vhodně doplnit algoritmus tak, aby prováděl detekci překážek před vozidlem, což by dále velmi zvýšilo užitnou hodnotu zařízení.

Literatura

- [1] Jun Wang, Tao Mei, Bin Kong and Hu Wei, "An approach of lane detection based on Inverse Perspective Mapping," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 35-38. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=6957662&isnumber=6957655>
- [2] A. M. Muad, A. Hussain, S. A. Samad, M. M. Mustaffa and B. Y. Majlis, "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," *2004 IEEE Region 10 Conference TENCON 2004.*, Chiang Mai, 2004, pp. 207-210 Vol. 1. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=1414393&isnumber=30646>
- [3] Z. Ying and G. Li, "Robust lane marking detection using boundary-based inverse perspective mapping," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, 2016, pp. 1921-1925. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=7472011&isnumber=7471614>
- [4] P. N. Bhujbal and S. P. Narote, "Lane departure warning system based on Hough transform and Euclidean distance," *2015 Third International Conference on Image Information Processing (ICIIP)*, Wagnaghat, 2015, pp. 370-373. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=7414798&isnumber=7414696>
- [5] J. He, H. Rong, J. Gong and W. Huang, "A Lane Detection Method for Lane Departure Warning System," *2010 International Conference on Optoelectronics and Image Processing*, Haikou, 2010, pp. 28-31. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5663615&isnumber=5662379>
- [6] Computer Vision. *Multimedia Interactive Didactic System* [online]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/LPOV_Exercise05/content_cz.php
- [7] P. Sthitpattanapongsa and T. Srinark, "A two-stage Otsu'S thresholding based method on a 2D histogram," *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, 2011, pp. 345-348. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=6047894&isnumber=6047826>

- [8] Sobel Edge Detector [online]. Copyright ©2003 R. [cit. 11.12.2018]. Dostupné z: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>
- [9] G. N. Chaple, R. D. Daruwala and M. S. Gofane, "Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA," *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, Mumbai, 2015, pp. 1-4. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=7095920&isnumber=7095833>
- [10] Laplacian of Gaussian [online]. Copyright ©2003 R. [cit. 11.12.2018]. Dostupné z: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>
- [11] Canny Edge Detector [online]. Copyright ©2003 R. [cit. 11.12.2018]. Dostupné z: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm>
- [12] Duda, Richard, O., Hart, Peter, E. *Use of the Hough Transformation To Detect Lines and Curves in Pictures*. Stanford Research Institute, California, 1972.
- [13] Dobeš, M., Závodný, M. *Vyhledávání v obraze a Houghova transformace*. Jednota českých matematiků a fyziků, Olomouc, 2010. [cit. 10.12.2018]. Dostupné z: https://dml.cz/bitstream/handle/10338.dmlcz/146342/Rozhledy_085-2010-1_3.pdf
- [14] V. Gaikwad and S. Lokhande, "An improved lane departure method for Advanced Driver Assistance System," *2012 International Conference on Computing, Communication and Applications*, Dindigul, Tamilnadu, 2012, pp. 1-5. [cit. 10.12.2018]. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=6179148&isnumber=6179125>
- [15] S. Zoican, M. Vochin, R. Zoican and D. Galațchi, "Lane departure warning system implementation using the Blackfin microcomputer," *2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, 2016, pp. 7-10. [cit. 10.12.2018] Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?tp=&arnumber=7781044&isnumber=7781033>
- [16] MUDROVÁ, M. Histogram a jeho zpracování. Ústav počítačové a řídicí techniky. Vysoká škola chemicko – technologická:2004. Dostupné z: <http://uprt.vscht.cz/mudrova/zob/prednasky/03-HISTOGRAM/histogramcz-tisk.pdf>

- [17] SEIDL, A. Revize TP 133 Zásady pro vodorovné značení na pozemních komunikacích. Praha : Ministerstvo dopravy odbor pozemních komunikací, 2005. Dostupné z:
https://www.dopravniznaceni.com/fotky50652/ke_stazeni/technicke_podminky.pdf
- [18] RASPBERRY PI FOUNDATION UK REGISTERED CHARITY 1129409. [online] Dostupné z: <https://www.raspberrypi.org/downloads/>
- [19] CCTV Lens Guide [online] Dostupné z:
<https://www.dtsdigitalcctv.co.uk/CCTV-Camera-Lens-Guide.asp>
- [20] OpenCV library. *OpenCV library* [online]. Copyright © Copyright 2018, OpenCV team [cit. 12.12.2018]. Dostupné z: <https://opencv.org/>
- [21] STEHLIKOVA, B., TIRPAKOVA, A., POMENKOVA, J., et al. Research Methodology and Statistical Inference. 1st ed. Brno (Czech republic): Mendel University, 2009. ISBN 978-80-7375-362-7

Seznam příloh

Příloha 1 - Program pro zkoušku knihovny OpenCV	60
Příloha 2 - Obsah přiloženého CD	61

Příloha 1 - Program pro zkoušku knihovny OpenCV

```
#include <stdio.h>
#include <opencv2/opencv.hpp>
#include <opencv2/highgui.hpp>

using namespace cv;
using namespace std;

int main(int argc, char ** argv)
{
    VideoCapture cap(0);
    if (!cap.isOpened()) {
        cerr << "ERROR: Unable to initialize the camera" <<
endl;
        return 0;
    }
    Mat frame;
    cout << "Start preview, press any key to end the program" <<
endl;
    while (1) {
        cap >> frame;
        if (frame.empty()) {
            cerr << "ERROR: Unable to capture frame" <<
endl;
            break;
        }
        imshow("Live preview", frame);
        if (waitKey(10) >= 0)
        {
            break;
        }
    }
    cout << "Closing the camera" << endl;
    cap.release();
    destroyAllWindows();
    cout << "END" << endl;
    return 0;
}
```

Příloha 2 - Obsah přiloženého CD

- Elektronická verze diplomové práce
- 3D model navržené krabice ve formátu *.stl
- Výkres navržené krabice
- Zdrojový kód programu Recorder.py
- Zdrojové kódy navrženého algoritmu