



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

RE-IDENTIFIKACE AUTOMOBILŮ POMOCÍ REGISTRAČNÍ ZNAČKY

VEHICLES RE-IDENTIFICATION WITH LICENSE PLATE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ HAŽMUKA

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAKUB SOCHOR

BRNO 2016

Abstrakt

Tato práce se zabývá návrhem a tvorbou aplikace pro reidentifikaci vozidel ve dvou videích. Práce popisuje metody zpracování obrazu a detekci objektů v něm a identifikaci symbolů. Vytvořená aplikace používá knihovnu OpenCV k velkému množství úkonů. Aplikace je napsaná v jazyce C++, standart C++11. Tato práce se dále popisuje otestování aplikace na vstupních datech včetně natrénování referenčních dat.

Abstract

This thesis describes design and development of an application for reidentification of vehicles in multiple camera streams. It explains methods of image processing, object detection and identification of symbols. Created application uses OpenCV library. It is written in the C++ language, standart C++14. Furthemore, this thesis describes testing of application on input data including training of the reference data.

Klíčová slova

OpenCV, SPZ, identifikace, reidentifikace, vozidla, detekce, sledování

Keywords

OpenCV, license plates, identification, reidentification, vehicle, detection, tracking

Citace

HAŽMUKA, Tomáš. *Re-identifikace automobilů pomocí registrační značky*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Sochor Jakub.

Re-identifikace automobilů pomocí registrační značky

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jakuba Sochora. Uvedl jsem všechny literární prameny a publikace, z nichž jsem čerpal.

.....
Tomáš Hažmuka
18. května 2016

Poděkování

Rád bych poděkoval mé rodině za psychickou podporu a vedoucímu Ing. Jakubu Sochorovi za vedení práce, rady, trpělivost a shovívavost.

© Tomáš Hažmuka, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Zpracování obrazu	4
2.1 Snímání, digitalizace a uložení obrazu v počítači	4
2.2 Předzpracování	5
2.3 Segmentace obrazu	9
3 Detekce celých oblastí	10
3.1 Prahování	10
4 Registrační značka	12
4.1 Poškození RZ	13
4.2 Využití identifikace RZ	13
5 Přístupy detekce, identifikace, re-identifikace vozidel	16
5.1 Dopravní detektory	16
5.2 Re-identifikace vozidel	17
6 Návrh řešení	19
6.1 Detekce státní poznávací značky	19
6.1.1 Sledování vozidel	19
6.1.2 Zpracování SPZ	20
6.2 Rozpoznání jednotlivých znaků	20
6.3 Reidentifikace	20
7 Implementace	21
7.1 Získání SPZ	21
7.1.1 Sledování vozidel	21
7.1.2 Dynamická změna pozadí	23
7.1.3 Odečtení pozadí	23
7.2 Segmentace SPZ	23
7.2.1 Získání plochy SPZ	24
7.3 Segmentace značky na jednotlivé znaky	26
7.3.1 Hlavní mezery SPZ	26
7.3.2 Mezery mezi znaky	27
7.4 Rozpoznání znaků	28
7.5 (Re)Identifikace	28

8	Vyhodnocení	29
8.1	Základní problém	29
8.2	Výsledky	30
9	Závěr	32
	Literatura	33
	Přílohy	35
	Seznam příloh	36
A	Obsah DVD	37

Kapitola 1

Úvod

Představte si, že stojíte venku a před vámi stojí rozkvetlá jabloň. Slyšíte zvuk kosa a hledáte, kde se nachází. Ve vaší scéně se pohybují listy, pohupující větve stromu, díky větru, který zrovna fouká. Vy ale stejně kosa objevíte na základě jeho typické barvy a odlišného pohybu od všeho ostatního a uvědomíte si, kde se ve scéně nachází[26].

Kapitola 2

Zpracování obrazu

Aby člověk byl schopen takovéto analýzy, musí mít nějaké předešlé znalosti věci, kterou zkoumá. Této části počítačového vidění se říká **vyšší úroveň**. Druhou částí je **nižší úroveň**, jinak také **zpracování obrazu**, která analyzuje dvojrozměrná vstupní obrazová data. Ty se poté dají použít například pro zpracování objektů v obraze[26].

Řekli jsme si, že je člověk obdařen schopností interpretovat velké množství informací okolo sebe jako jsou barvy, různé tvary či textury na základě zkušeností a znalostí.

Počítač tyto schopnosti nemá a je velice náročné mu tyto schopnosti předat, aby byl schopen zpracovávat digitální obrazové data z reálného světa, například z fotoaparátů nebo kamer. Obraz může být pořízen v různých oblastech záření jako je oblast viditelného spektra (denní světlo) nebo v oblastech, které lidské oko nevidí jako jsou například infračervené snímky nebo rentgenové snímky.

Cílem je více porozumět obsahu obrazu. Zpracování obrazu je jednou z perspektivních disciplín v oblasti počítačové grafiky[27][11][15].

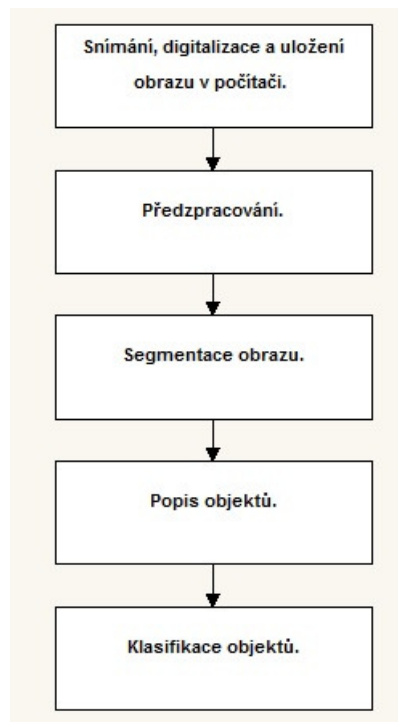
2.1 Snímání, digitalizace a uložení obrazu v počítači

Při digitalizaci obrazu dochází k převedení analogového (spojitého) signálu reprezentujícího jas na diskrétní hodnoty. K převodu dochází kvůli tomu, že neumíme zaznamenat a uložit spojité hodnoty naproti diskrétním hodnotám. Mezi nejběžnější způsoby digitalizace je snímání digitálním fotoaparátem a skenování. Nejrozšířenějšími snímači obrazu jsou:

- **CCD** (Charge Coupled Device)
- **CMOS** (Complementary metal-oxide-semiconductor)

Formáty pro ukládání obrazu dělíme na **rastrové** a **vektorové**. Rastrové jsou s kompresí nebo bez komprese. Kompresí slouží ke zmenšení objemu uložených dat a dělí se na:

- **Bezeztrátová komprese** - je definována způsobem, že v komprimovaném obraze zachovává všechny informace o původním obraze a při zpětné rekonstrukci dostaneme zpět původní obraz. Pro tento způsob komprese se využívá například formát **RAW** nebo **PNG**.
- **Ztrátová komprimace** - je založena na nedokonalosti lidského oka. Dosáhneme větší komprimace než u bezeztrátové komprese, ale už při zpětné rekonstrukci nedosáhneme původního obrazu, tuto kompresi využívá formát **JPEG**.



Obrázek 2.1: Zpracování a rozpoznání obrazu reálného světa

Rozdělení příslušných formátů:

- **JPEG** (Joint Photographic Expert Group) - používá ztrátovou kompresi založenou na separaci jasové a barvonosné složky obrazu, na které je aplikovaná diskrétní kosinová transformace (DCT).
- **RAW** („Surový formát“) - dosahuje maximální kvality snímků, jelikož se jedná o zápis hodnot ze snímače, tak jak byly naměřeny. Používá se pro profesionální fotografie.
- **PNG** (Portable Network Graphics) - patří mezi velice oblíbené formáty užívané zejména u medicínských a astronomických obrazů. Umožňuje použít informace o průhlednosti – alfa kanál[15]

2.2 Předzpracování

Na výsledný snímek působí mnoho faktorů jako je správná expozice, protisvětlo, velké kontrasty stínu a světlé části scény. Při působení těchto vstupních vlivů dostaneme obraz, ne v takové kvalitě, jak jsme očekávali. Naštěstí můžeme použít úpravy vedoucí ke zlepšení vzhledu obrazu (**Image Enhancement**). Základní rozdělení metod předzpracování obrazu:

- **Jasová transformace**
- **Histogram a transformace založené na změně histogramu**
- **Geometrické transformace a interpolace jasu**



Obrázek 2.2: Vstupní obraz v odstínech šedi a převedený obraz do negativu

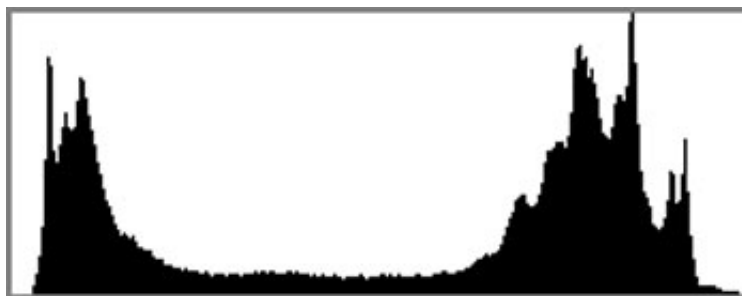
Jasová transformace

Jednou z nejjednodušších jasových transformací je tzv. **Negativ**. Nevyužívá informací z předchozích analýz obrazu. Intenzita jasu každého pixelu v obraze je nahrazena novou hodnotou intenzity jasu $g(x,y) = 1 - f(x,y)$ pro všechny x, y v obraze. Negativ je dán předpisem $g = 1 - f$. V případě, kdy používáme L diskretních jasových úrovní $\langle 0, L - 1 \rangle$, se vzorec změní na $g = L - 1 - f$. Další z jasových transformací je **Gama korekce**.

Důvodem jeho vzniku byla nelineární závislost televizních obrazovek. Problém byl u klasických obrazovek ze žhavou katodou CRT, které nezobrazovaly vstupní napětí lineárně. Využití Gama korekce je také v oblasti opravy špatné expozice, kdy je požadovaná část objektu ukryta v příliš tmavé nebo naopak v příliš světlé části obrazu.

Histogram a transformace založené na změně histogramu

Jedná se o grafické zobrazení jasových úrovní od bílé po černou. Vodorovná plocha značí počet odstínů ve škále od 0-255 a svislá plocha značí velikost plochy, kterou jednotlivé odstíny zabírají.

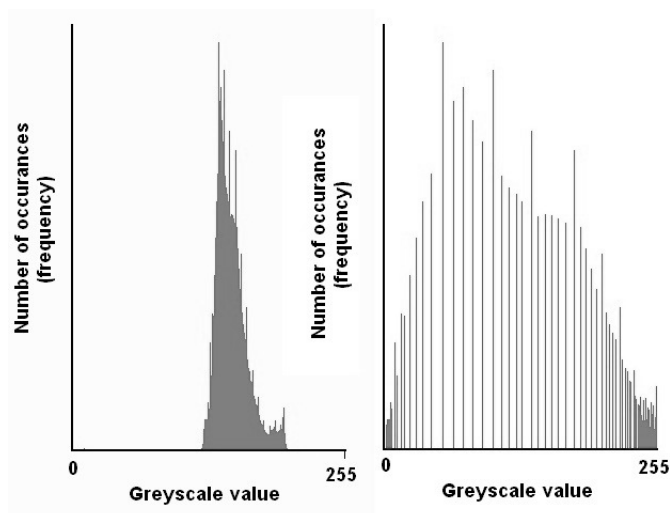


Obrázek 2.3: Histogram

- **Metoda vyrovnání histogramu** - anglicky Histogram Equalization, pokud máme nevhodně exponovaný obraz, to znamená, že obraz je nekонтastní a jednotlivé jasové složky se vyskytují nerovnoměrně. Metoda vyrovnání histogramu upravuje kontrast rastrového obrazu s využitím histogramu, dojde k tomu, že četnost výskytu jasových složek je rovnoměrnější a detaily v obraze jsou lépe rozlišitelné a kontrastnější. Tato metoda nachází nejlepší využití v obrazech, které jsou příliš světlé nebo příliš tmavé[15][22].



Obrázek 2.4: Vyrovnání histogramu by Phillip Capper by wiki



Obrázek 2.5: Vyrovnání histogramu by Konstale by wiki

Geometrické transformace

Jedny z častých úprav obrazů se nazývají geometrické transformace. Používají se k opravě zdeformovaných objektů nebo naopak k samotné deformaci. Mezi základní transformace patří posunutí, otočení, změna měřítka nebo zkosení[15].

Morfologická transformace

Morfologie se využívá v biologii a jazykovědě, ale nás zajímá její použití v digitálním zpracování obrazu, kde funguje jako matematický nástroj pro předzpracování, segmentaci i finální úpravy obrazů. Obvykle pracujeme s binárním obrazem nebo můžeme použít šedotónové obrazy.

- **Eroze** - je definována jako průnik množiny obrazu A s množinou všech posunů masky B. Používá se ke zjednodušení struktury objektů (zmenšení objektů, odstranění pruhů podél hranic objektů, vyhlazení hran, odstranění izolovaných malých objektů).
- **Dilatace** - je sjednocení množiny obrazu A s množinou všech posunů masky B. Používá se k zaplnění malých děr a úzkých zálivů v objektech, dále zvětšuje objekty a vyhlazuje hrany.



Obrázek 2.6: Geometrická transformace typu rotace



Obrázek 2.7: Původní snímek a snímek po erozi



Obrázek 2.8: Původní snímek a snímek po dilataci

- **Binární otevření a zavření** - binární otevření a zavření jsou operace založené na dilataci a erozi, obě operace vyhlazují objekty, otevření filtruje objekty zvenčí a

zavření zevnitř. Otevření je eroze následována dilatací. Zavření je dilatace následována erozí[17][13].



Obrázek 2.9: Převod původního snímku na snímek za pomoci otevření a zavření

2.3 Segmentace obrazu

Jednou z hlavních a klíčových oblastí zpracování obrazu je **segmentace obrazu**. Jejím úkolem je rozlišit jednotlivé objekty v obraze jako rozpoznávání či klasifikace objektů. Najdeme algoritmy, které můžeme zařadit mezi naprosto triviální až po velmi složité[19]. Za objekty se dá považovat část obrazu o kterou se zajímáme a kterou chceme dáme zpracovávat.

Segmentace je soubor oblastí a pokud odpovídá objektům ve vstupním obraze, tak se jedná o **kompletní segmentaci**. A když neodpovídá, tak se jedná o **částečnou segmentaci**, který je založena na vlastnostech jako jas nebo barva uvnitř segmentu[16]. Segmentace je nejčastěji založena na detekci kontur(hran) ohraničujících jednotlivé objekty nebo na detekci celých oblastí, kterými jsou jednotlivé objekty v obraze reprezentovány[19].

Kapitola 3

Detekce celých oblastí

K detekování oblasti dochází na základě homogenních kritérií, jako jsou stejné nebo velmi blízké hodnoty jasu, barvy a nebo pokrytí stejnou texturou. Metody detekce oblastí lze rozdělit do tří skupin[19]:

- detekce prahováním (**thresholding**)
- detekce narůstáním oblastí
- detekce dělením oblastí

3.1 Prahování

Prahování (Thresholding) je nelineární operace převádějící obraz v barvě šedi na binární obraz, který nabývá dvou úrovní přiřazených pixelů - nad úrovní zadaného prahu a pod její úrovní. Pokud bychom chtěli používat pouze jedinou hodnotu zvoleného prahu, tak by se naše metoda hodila jen na velice jednoduché aplikace.

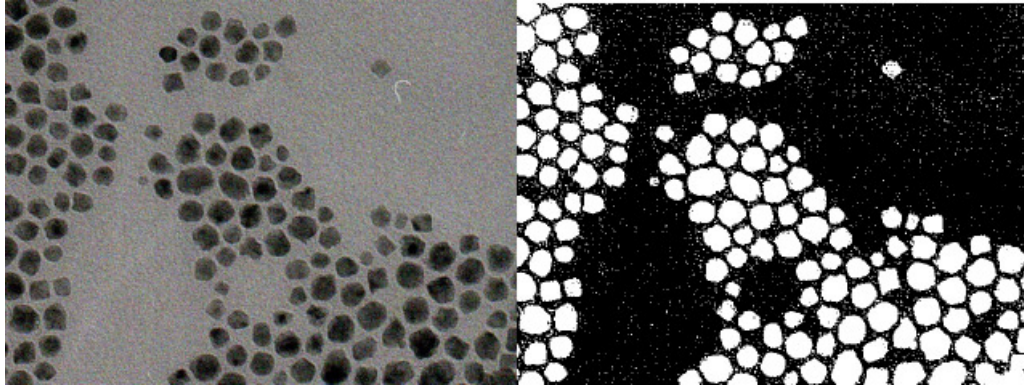
Proto se častěji volí automatický práh za pomoci například metod standardního histogramu, metod procentního prahu nebo metod váženého/vysokého gradientu[18].

Globální prahování

Jedná se o nejjednodušší případ použití jedné prahové úrovně. Segmentace je potom provedena tak, že se prochází pixel po pixelu a zjišťuje se zda se jedná o bod objektu nebo o bod pozadí, tzn. závisí to na tom zda je hodnota pixelu větší nebo menší než zvolená prahová úroveň. Příklady globálního prahování:

- **Threshold Binary** - hodnoty pixelů jsou rozděleny dle úrovně jejich jasu. Hodnoty, které tento práh překročí se stanou "1" a hodnoty pod prahem "0".
- **Truncate** - funguje na principu, že maximální hodnoty intenzity jasu jako nastavený práh, pokud má úroveň pixelů vyšší hodnotu, tak je jejich hodnota zkrácena na úroveň nastaveného prahu. Pokud mají hodnoty pixelů nižší hodnotu, tak zůstanou nezměněny.
- **Threshold to Zero** - další metodou je prahování k nule, která funguje na principu, že pokud je hodnota pixelu menší než práh, tak je jejich hodnota nastavena na "0" a

v případě vyšší hodnoty je její stav nezměněn. Existuje i její inverzní verze **Inverted Threshold to Zero** u níž hodnoty pixelů menší než hodnota prahu zůstanou nezměněny a hodnoty vyšší nastaveny na "0".

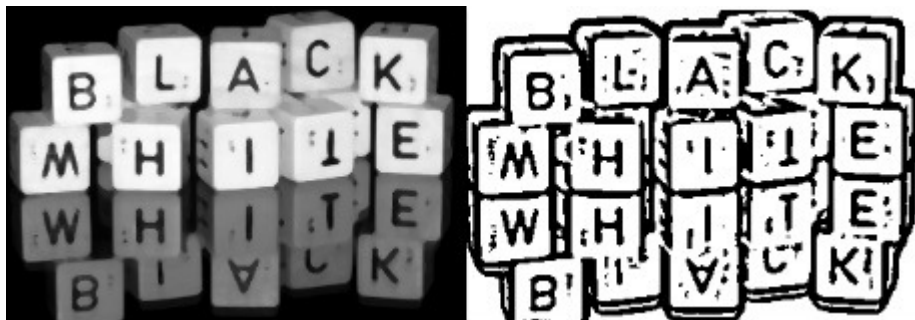


Obrázek 3.1: Binární prahování

Lokální prahování

Použití globálního prahování na objekt, který má ve svých různých hodnoty jasu nebo na nerovnoměrně osvětlený obraz nedává příliš dobré výsledky. Lokální prahování respektuje různé hodnoty jasu v různých částech obrazu a dává lepší výsledky.

Princip spočívá v tom, že je postupně zkoumáno okolí každého pixelu v obraze a hodnota prahu je stanovena na základě informace z tohoto okolí[15].



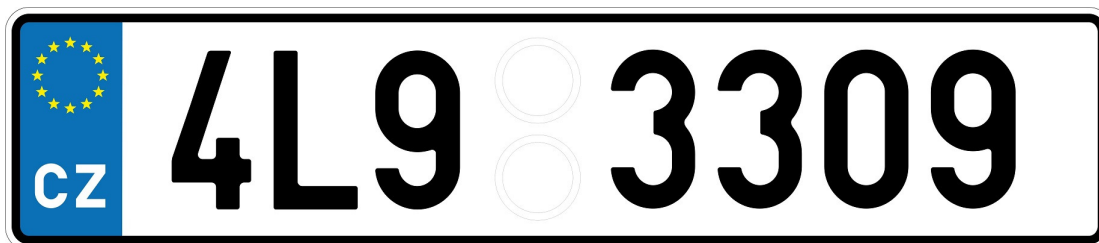
Obrázek 3.2: Adaptivní prahování

Kapitola 4

Registrační značka

Státní poznávací značka (**SPZ**) je unikátní identifikátor každého motorového vozidla, přívozu či návěsu. 29.června 2001 přinesl zákon č.56 a prováděcí vyhláška č.243/2001 Sb. podobu registračních značek, jak je známe dnes. Došlo ke změně názvosloví ze "státní poznávací značky (**SPZ**)" na "registrační značku (**RZ**)". Po zrušení okresních úřadů byl vydávání RZ převedeno na obce s rozšířenou působností (**ORP**). Základní rozlišení je rozděleno podle krajů jako například:

- **A**-Praha
- **B**-Jihomoravský kraj (Brno)
- **J**-Vysočina (Jihlava)



Obrázek 4.1: Česká registrační značky

Nové značky obsahují 5 až 7 znaků (číslíce a písmena bez diakritiky s výjimkou G,O,Q,W, u kterých by mohlo dojít k záměně s písmeny C,V a číslicí 0. Dále musí RZ obsahovat alespoň jedno písmeno a jednu číslici. První znak zleva označuje příslušný kraj. Kromě rozlišení pomocí čísel a písmen se používá rozlišení pomocí barev jako jsou například:

- Černá písmena na bílém značce (pro všechna osobní, nákladní vozidla, autobusy, přívozy a motocykly)
- Modrá písmena na bílé značce (auta diplomatických a konzultárních úředníků)
- Černé písmena na zelené papírové značce (jednorázové použití s omezenou platností, například při převozu z autosalonu domů)

Od ledna roku 2016 je možné zažádat o vlastní registrační značku. Značka může obsahovat 7 až 8 znaků, na základě, že se jedná o automobil nebo motorku. Velké množství výrazů

je však zakázáno. Jedná se o vulgární, rasistické výrazy, názvy úřadů, policie a spousta dalších.

1. května 2004 po vstupu ČR do Evropské unie došlo k poslední úpravě RZ a to takové, že se do levé části značky umístil modrý kruh Evropské unie s rozlišovací značkou České republiky(CZ). Kromě RZ České republiky existuje velká škála cizích značek rozlišující se nejenom množstvím znaků, čísel na značce, ale i odlišnou barvou a rozměry viz. Obrázek 4.2[24][9][6][8].



Obrázek 4.2: Česká registrační značka

4.1 Poškození RZ

Pokud se zaměříme na RZ z pohledu jejich identifikace, tak při implementaci rozpoznání RZ musí programátor počítat s problémy jako:

- Mechanické poškození registrační značky jako jsou praskliny či škrábance, které mohou nastat při situacích jako jsou menší autonehody.
- Znečištění registrační značky způsobené jízdou za nepříznivých podmínek jako je například prach nebo bahno nalepený na značce automobilu.

Tyto defekty se mohou pohybovat od lehkých, které pouze znevýhodní identifikaci až po těžké, které identifikace znemožní. Jako příklad může být uvedena úplná ztráta RZ vlivem mechanického poškození.

4.2 Využití identifikace RZ

Mezi nejčastější využití identifikace RZ patří jejich použití v oblastech jako je měření rychlosti, jízda na červenou nebo jízda v protisměru.

Detekce jízdy na červenou

Na sledovaném místě (křižovatce) je umístěna jedna přehledová kamera pro 1 až 3 jízdní pruhy a jedna detailová kamera pro každý měřený jízdní pruh. Pomocí přehledové kamery



Obrázek 4.3: Znečištěná registrační značka

se zjišťuje fáze semaforu, dále celkové situace na křižovatce, historie vjezdu vozidel do křižovatky a pořizuje se videozáznam celé situace.

Detailová kamera slouží k pořízení snímku vozidla včetně registrační značky a tváře řidiče. Počítač zpracovává situaci detekovanou jednotlivými kamerami, uchovává údaje o přestupcích a zaznamenává situaci na křižovatce[1]

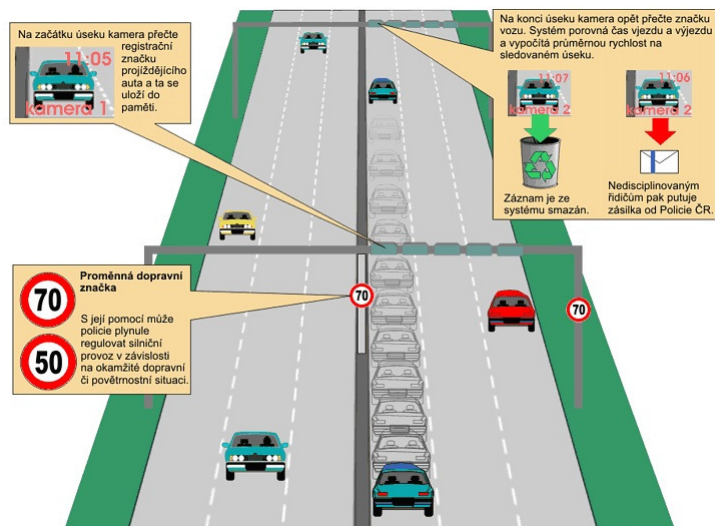
Měření rychlosti

. V ČR se pro měření rychlosti používají radarové či laserové měřiče. Úsekové měření rychlosti a stacionární radary kontrolují dodržování rychlosti jízdy v určitém úseku. Každý systém, ale funguje na jiném principu.

Měřené rychlosti radarem využívá **Dopplerova jevu** - porovnává frekvenci vysílaného a odraženého signálu - čím rychleji auto jede, tím větší je rozdíl obou frekvencí.

- **Stacionární radary** - měří aktuální rychlost vozidla pouze na jednom konkrétním místě.
- **Úsekové měření rychlosti** - na počátku a na konci měřeného úseku je vozidlo vyfotografováno s přesně uvedeným časem. Automaticky je spočtena průměrná rychlost vozidla v měřeném úseku. Pokud je vyšší než povolena, údaje o vozidlu jsou poslány policii nebo městskému úřadu.
- **Mobilní radary** - buď se jedná o ruční přenosné radary a nebo o radary zabudované v policejních autech.

V případě **překročení povolené rychlosti** musí radary počítat s odchylkou. V ČR je tato odchylka jednotně stanovena na $\pm 3\text{km/h}$ při zjištěné rychlosti do 100km/h , respektive 3% při zjištění rychlosti nad 100km/h . Jako příklad si můžeme uvést, že pokud vozidlo překročí povolenou rychlost v obci 50km/h , můžeme jí spolehlivě prokázat až se na radaru objeví minimálně 54km/h .



Obrázek 4.4: Příklad probíhání úsekového měření

Velmi kontroverzní kapitolou měření rychlosti je použití tzv. "**antiradarů**". Pokud se jedná o pasivní radary, tak je zákon nezakazuje, protože pouze upozorňují na probíhající měření. To neplatí pro tzv. "**rušičky**", které policejním radarům znemožňují měření.

Měření může provádět pouze Policie České republiky. Městská policie má také oprávnění měřit rychlost vozidel, ale musí splnit některé omezení. Smí měřit pouze na místech určených Policií ČR a tato místa musí být označena značkou "Měření rychlosti" [7][3][5].

Ochrana osobních údajů

Jednou z další oblastí identifikace vozidel potažmo RZ je oblast ochrany osobních údajů. Většina z nás zná aplikaci **Google Street View**, která nabízí panoramatické pohledy. V případě, že se v záběru objeví automobil, dojde k rozmazání RZ [23].

Další ukázkou použití může být systém **UnicamPRIVACY**, který slouží k automatickému rozostření spolujezdce při vyfocení vozidla. [4]



Obrázek 4.5: Rozmazání RZ v Google Street View

Kapitola 5

Přístupy detekce, identifikace, re-identifikace vozidel

5.1 Dopravní detektory

Jedná se o zařízení sledující aktuální dopravní situaci v daném místě. Díky detektorům lze zajistit dynamické řízení dopravy a získávání dopravních informací. Detektory jsou umístěny buď vedle komunikace, nad komunikací, do vozovky nebo na povrch vozovky. Detektory slouží k získávání informací o vozidlech nebo chodcích například v prostoru křižovatek, na přechodech pro chodce, na dálnicích nebo na významných silničních komunikacích. Získané informace lze využít pro predikci cestovních časů vozidel, optimalizaci řízení dopravních toků. Mezi nejčastěji v praxi používané detektory jsou:

- **Indukční smyčky**
- **Videodetekce**
- **Mikrovlnné detektory**

Ale můžeme se setkat i s infračervenými, laserovými a ultrazvukovými detektory.[10][2]

Indukční smyčka

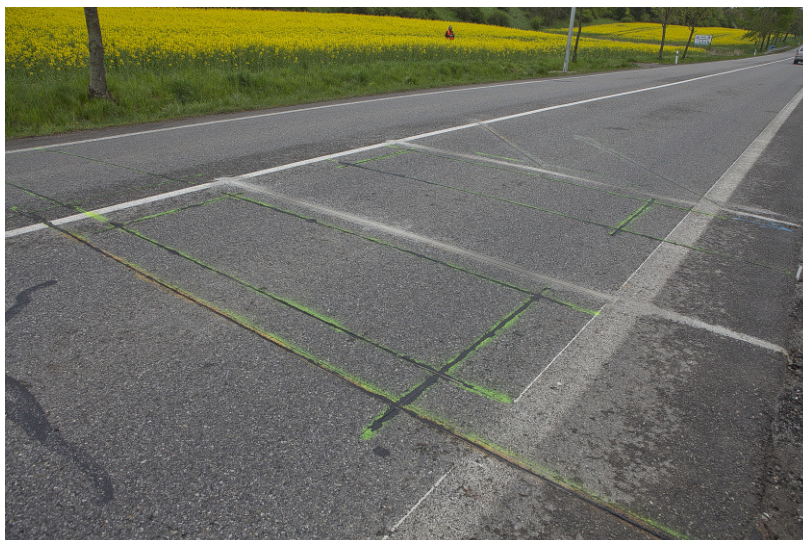
Je jedním z nejběžnějších dopravních detektorů určených řízení dopravy. V případě umístění jedné smyčky může sloužit ke sčítání vozidel, světelné signalizaci a pokud vhodně umístíme více smyček za sebe můžeme měřit rychlost vozidel. Indukční detektor využívá změny indukčnosti smyčky, která je umístěna ve vyřezané spáře v asfaltu v jízdním pruhu.

Princip fungování indukčních smyčky je takový, že pokud vozidlo přejede přes smyčku, dojde ke snížení její indukčnosti a naopak zvýšení frekvence oscilátoru. Pokud změna frekvence přesáhne daný práh, je to vnímáno jako přítomnost vozidla na smyčce.

Velkou výhodou smyček jsou jejich poměrně nízké náklady a jejich nezávislost na počasí, nevýhodou je jejich složitější instalace do vozovky[21][10].

Mikrovlnné detektory

Jedná se o zařízení založené na principu vysílání a následném příjmu mikrovlnných paprsků. Díky svým malým rozměrům a jednoduché montáži na stožár jsou tyto detektory možnou náhradou za indukční smyčky v místech, kde není možný zásah do povrchu vozovky[10].



Obrázek 5.1: Indukční smyčka

Videodetekce

Měření toku dopravy a automatická detekce pomocí kamer je další formou detekce vozidel. Systémy založené na videodetekci, které se používají pro automatické rozpoznání RZ, nezahrnují instalaci komponentů přímo do povrchu vozovky, takže jsou známé jako "nedestruktivní" metody detekce provozu.

Video z kamer je převáděno do procesorů, které analyzují měnící se vlastnosti obrazu a procházejících vozidel. Kamery jsou obvykle namontovány na sloupy nebo konstrukce nad nebo v blízkosti vozovky. Jediný procesor videodetekce dokáže detekovat provoz současně až z osmi kamer, v závislosti na značce a modelu.

Typickým výstupem systému je rychlost vozidel, jejich počet a obsazení pruhů. Systémy se dají využít na místech, na kterých není možné použít indukční smyčky (oblast tramvajové dopravy, historická centra měst)[10][20].

5.2 Re-identifikace vozidel

Re-identifikace vozidel vyžaduje sady detektorů namontovaných podél silnice. Dochází k detekci RZ vozidla v jednom místě a její znovu zjištění (re-identifikace) v dalším bodě dále po silnici. Doba cesty vozidla a rychlost jsou vypočteny na základě porovnání časů, kdy došlo k detekci dvojicí senzorů.

Využití přístupu re-identifikace vozidel najdeme ve městech, kde pomocí videokamer získáváme údaje o provozu. Při použití stacionárních kamer připevněných na budovách získáváme data 24hodin denně za každého počasí. Data mohou být použita pro optimalizaci dopravního toku - dynamické řízení semaforů.

V případě stacionárních kamer s cílem minimalizovat jejich počet je vhodné re-identifikovat vozidlo opouštějící jednu sledovanou oblast a poté vstupující do zorného pole další kamery. Koncept je založený na rozpoznání vozidel z libovolných úhlů pomocí přístupu založeném na 3D modelu.

Algoritmus je schopen získat střední část horní roviny objektu a z této části jsou extrahovány odpovídající barevné hodnoty, poté převedeny na **HSV** (odstín, sytost, hodnota

barevného prostoru) a použity jako parametr pro re-identifikaci vozidel.

Dochází ke korigování stínů a odrazů světla na mokré ulici a proto je dosaženo vysoké přesnosti rozpoznání, která je nutná najít v horní polovině vozidla. Algoritmus je vhodný pro aplikace v reálném čase[20][25].

Kapitola 6

Návrh řešení

Cílem práce je vytvořit systém, který analyzuje data ze dvou kamer umístěných nad stejnou silnicí v určité vzdálenosti od sebe, rozpoznává státní poznávací značky a identifikuje vozidla ve dvou pruzích v jednom směru.

Problém bych rozdělil na dvě části.

1. Analýza jednoho videa, která vygeneruje seznam detekovaných SPZ vozidel.
2. Korelace výstupu z obou kamer a reidentifikace.

Při dalším průchodu programem na jiném videu spuštěném s parametrem reidentifikace, program vypíše všechny SPZ, které byly rozpoznány v obou testovaných videích. Program bude rozdělen na tři základní části:

1. detekce SPZ
2. rozpoznání znaků na SPZ
3. reidentifikace

6.1 Detekce státní poznávací značky

Detekce SPZ bude provedena pomocí klasifikátoru Haarových příznaků (Haar Cascades)¹, který, pokud je dobře natrénován, vykazuje velice dobré výsledky a právě proto se tento přístup využije. Protože SPZ jsou mezi sebou velice podobné, tak jejich natrénování není tak složité jako natrénování aut nebo například lidí.

Poznávací značka bude vyhledávána v části každého snímku videa. Z optimalizačních důvodů omezím plochu pro vyhledávání SPZ na kontury vozidel, které budou získány díky detekci pohybu (stálé změny vůči silnici) a z plochy vozidla lze využít jen spodní část (obsahuje SPZ). Navíc lze očekávat snížení počtu falešně detekovaných SPZ.

6.1.1 Sledování vozidel

Pro sledování vozidel se využije pozice detekovaných SPZ, proto bude vhodné uložit pozici SPZ do struktury, která bude také obsahovat její identifikátor a čas jeho posledního výskytu.

Pokud bude nalezená SPZ dostatečně blízko jiné, nalezené v některém z předchozích záběrů (ne však příliš starém), lze tuto SPZ přisoudit stejnému autu. V opačném případě se jedná o nové auto, které se dostalo do záběru, a je třeba na to adekvátně zareagovat.

¹http://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_face_detection.html Haar Cascades

6.1.2 Zpracování SPZ

Po úspěšné detekci bude třeba plochu SPZ ořezat, abychom získali pouze značku ohraničenou jejím černými okraji. Okolní pozadí může být pro úspěšnou identifikaci nežádoucí.

6.2 Rozpoznání jednotlivých znaků

Další fází bude oddělení jednotlivých znaků, které následně za použití KNN algoritmu (K-nearest neighbors) ² porovnáme s předpřipravenými vzory jednotlivých znaků. V případě shody zjistíme rozpoznávaný znak. Toto bude provedeno pro každou pozici na SPZ, z čehož vznikne sada sedmi znaků příslušející k jednotlivým pozicím. Výsledek bude rozpoznávaný kandidát státní poznávací značky.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Obrázek 6.1: Znak

1234567890

Obrázek 6.2: Číslo

Jelikož jedno vozidlo se vyskytuje na několika po sobě následujících snímcích, program získá několik SPZ. Nabízí se dva přístupy určení konečné SPZ. Prvním přístupem může být vybrání nejčastější značky jako celku ze všech detekovaných. Druhým pak je vybrání nejčastějšího znaku v jednotlivých sloupcích, což vede k určení nejlepšího možné SPZ daného auta.

6.3 Reidentifikace

Vzhledem k tomu, že program je koncipován tak, aby videa analyzoval zpětně, je třeba nalezené SPZ uložit do souboru. Při reidentifikaci pak aplikace načte seznam SPZ ze zadaného souboru a bude v něm vyhledávat. Pokud dojde ke shodě, aktuálně porovnávaná SPZ bude prohlášena za reidentifikovanou, uložena do výstupního souboru a odstraněna z načteného seznamu.

²https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm KNN algoritmus

Kapitola 7

Implementace

Program implementuji v jazyce C++, zejména kvůli knihovně OpenCV.

Algoritmus programu je následující:

1. načtení hodnot pro porovnání
2. inicializace všech datových struktur
3. iterace nad snímky videa:
 - (a) převedení obrazu do odstínů šedi
 - (b) snížení rozlišení snímku
 - (c) na základě současného snímku s váhou 0.001 uprav pozadí
 - (d) na základě rozdílu pozadí a snímku získej pohybující se objekty
 - (e) získej SPZ všech vozidel v obraze

7.1 Získání SPZ

7.1.1 Sledování vozidel

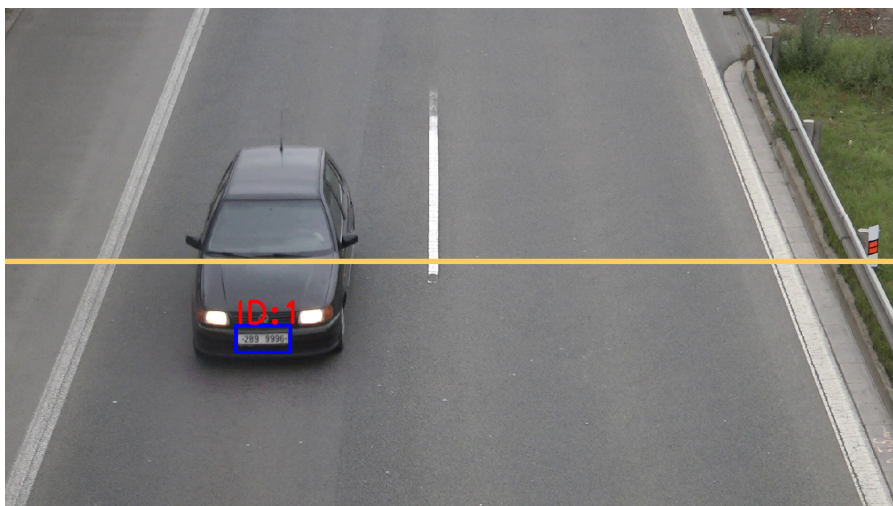
K tomuto účelu slouží funkce *detect_spz()*. Její nejdůležitější částí je seznam *cars*, který obsahuje informace o zaznamenaných značkách. Prvky tohoto seznamu jsou struktury, obsahující:

- **id** - id položky, slouží spíše k debugovacím účelům
- **frame** - pořadové číslo posledního snímku, na kterém se vozidlo vyskytlo
- **spz** - obdelník obsahující část snímku s SPZ

Algoritmus metody *detect_spz()* je následující:

1. odstraň ze seznamu *cars* všechny položky, kde rozdíl jejich snímku a čísla aktuálního snímku je vyšší než 25 (což je 0.5 sekundy pro kameru s 50 FPS),
2. zařaď pohybující se objekty do seznamu,
3. získej boxy ohraničující jednotlivé SPZ a pro každý z nich:

- prohledej všechny položky seznamu cars, pokud je nalezen box, jehož horní levý roh je vzdálen maximálně 100px¹ od levého horního rohu aktuálního boxu, lze říct, že se jedná o stejné auto a položky *frame* a *spz* jsou aktualizovány
- v opačném případě - pokud není nalezeno žádné auto - je vytvořena nová struktura, naplněna a vložena do seznamu cars



Obrázek 7.1: Sledování vozidla a detekce SPZ

Funkce detectMultiScale

Funkce `detect_spz()` dostane na vstup masku, která určuje místa, kde se nachází pohybující se objekty. Zjistí se jejich kontury a pokud jsou větší než je stanovený limit, aby byly odfiltrovány nežádoucí objekty jako například motorky. Protože na začátku programu z důvodu zvýšení rychlosti detekce aut, která nepotřebuje rozlišení 1920x1080, musí být teď box přepočítán zpátky na výchozí velikost. Poté jsou jednotlivé boxy uloženy a mohou jít na vstup kaskády, která vrátí pozici SPZ ve snímku.

Tento přístup je daleko rychlejší než kdyby byla detekovaná SPZ z celého snímku, dalším efektem je snížený počet detekcí falešných kandidátů SPZ. Nakonec musíme dopočítat detekovaný box zpátky do celého snímku, tak aby se jeho souřadnice vztahovaly k původnímu obrázku, který na konci programu zobrazuje. Detekce probíhá jen v dolní části snímku, což vede k dalšímu zrychlení programu (viz kapitola 8.1).

Změna velikost rozlišení

Jak už bylo zmíněno v návrhu řešení První krátkou, ale za to důležitou částí, je změna rozlišení z výchozího 1920x1080 na rozlišení 25x menší. Kvůli tomu, aby byl zrychlen chod programu.

Samotné rozpoznání SPZ

Při průběžném rozpoznávání SPZ se výsledky ukládají do dalšího, globálního seznamu - *list_id_spz*, ve kterém jsou uloženy dvojice id auto - rozpoznaná SPZ. Jakmile dojde k

¹tuto hodnotu jsem zvolil experimentálně na základě testování

odstranění auta ze seznamu *cars*, vezmou se všechny odpovídající záznamy ze seznamu *list_id_spz*. Výsledná SPZ přiřazená k právě odstraněnému auto se pak skládá ze sekvence nejčastějšího znaku na každé pozici (na základě záznamů vyjmutých z *list_id_spz*).

7.1.2 Dynamická změna pozadí

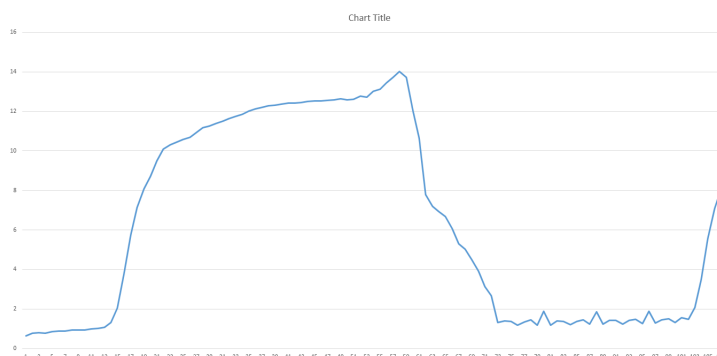
Při prvním průchodu načte připravené prázdné pozadí (silnice bez vozidel) z důvodu možnosti, že by první snímek už něco obsahoval (auta, chodci, motorky,..). Po změně velikosti pozadí na aktuální velikost vstupního snímku se nastaví příslušné pozadí jako referenční.

Každý deset sekund přichází nový snímek, který je porovnán s maskou pozadí. Poměr pozadí je nastaven na 0.001, to znamená, že by musel stejný obrázek přijít 1000x za sebou, aby se na něj kompletně změnilo pozadí. Jeho cílem je reagovat na dlouhodobé změny.

7.1.3 Odečtení pozadí

Nezbytnou součástí programu je využití funkce `absdiff()`, která odečte aktuální snímek od připravené masky pozadí (vygenerovanou funkcí `background_substraction()`) a vrátí snímek pouze s autem a zbylým šumem.

K odstranění zbylého šumu používám Otsu prahování² s následnými morfologickými operacemi. Protože tento způsob prahování lze správně využít pouze pokud je na aktuálním snímku vozidlo (histogram má dva vrcholy, naproti tomu, když je na snímku jen silnice, tak má jen jeden), takže pomocí funkce `meanStdDev()` zjistím jaká je odchylka od průměrné hodnoty snímku, pokud je hodnota vyšší než 6 7.2, tak do snímku vstupuje vozidlo a použiji Otsu prahování.



Obrázek 7.2: Průměrná hodnota snímku

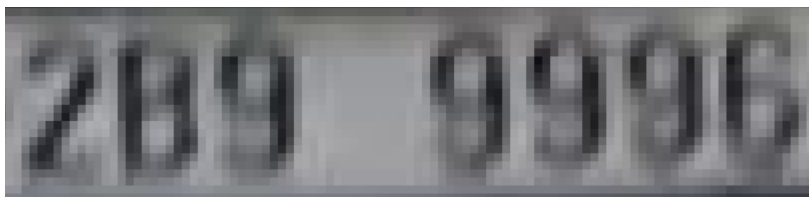
Protože vozidla nejezdí přes celou plochu snímku, byla vytvořena maska oblastí, kde se vozidla nevyskytují a pomocí `bitwise_and()` aplikovaná na vstupní snímek. Nakonec funkce `findContours()` najde kontury vozidel, `convexHull()` je spojí a `drawContours()` je vyplní.

7.2 Segmentace SPZ

Kvalita SPZ

Jak je vidět níže, kvalita vstupní poznávacích značek je nízká. Velikost znaku je okolo 20px a šířka okolo okolo 10px. Proto nebylo její zpracování a následná identifikace triviální.

²<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html> Otsu Thresholding



Obrázek 7.3: Kvalita SPZ

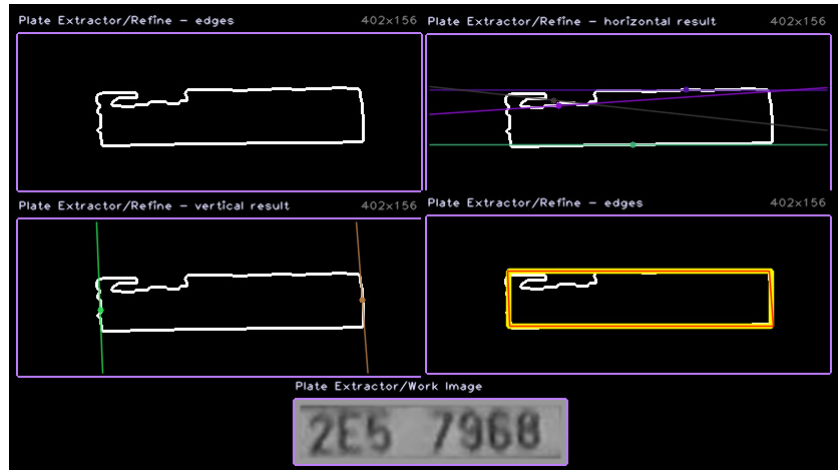


Obrázek 7.4: Práce v HSV modelu

7.2.1 Získání plochy SPZ

V této fázi je cílem ořezat vstupní SPZ tak, abych se zbavil přebytečného pozadí a také odstranění černých okrajů SPZ, které splývají se samotnými znaky.

- provedu zvětšení vstupního snímku abych přešel slévání jednotlivých kontur
- ve funkci *make_plate_matching()* převedu vstupní snímek do **HSV** modelu pomocí, kterého vytvořím dvě masky:
 - masku obsahující místa, které jsou hodně světlá
 - masku obsahující místa, která mají málo barvy
 - jako výsledek dostanu masku světlých málo barevných míst (čímž se zbavím například modrého pruhu Evropské unie v levé části) (viz 7.4)
- následně ve funkci *find_plate_matches()* najdu všechna místa, která by mohla být prohlášena za SPZ
- detekuji kontury v masce a odstraním šum, tím, že odstraním příliš malé kontury
- poté vyberu pravděpodobné kontury dle vzdálenosti těžiště od středu, předpokladem je, že je značka vždy uprostřed což bylo zajištěno předchozí fází.

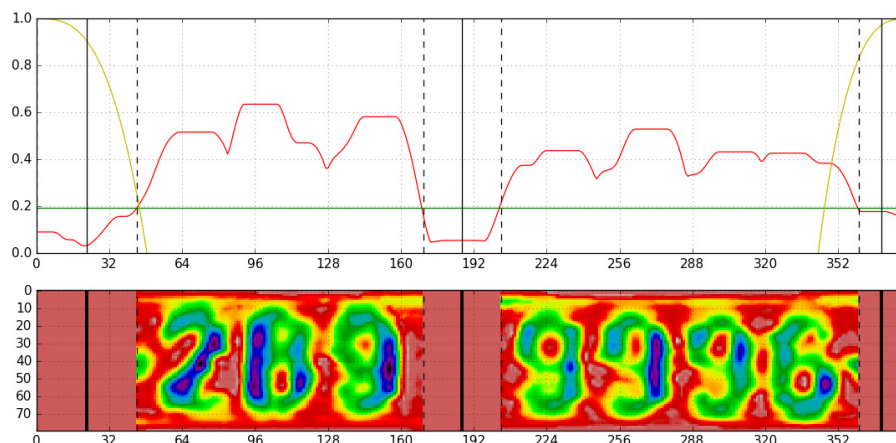


Obrázek 7.5: Nalezená kontura, horizontální/vertikální čáry, výsledný obdélník a extrahovaná SPZ

- touto fází získáme jednu, či více hrubých masek, definujících potenciální místa SPZ
- tyto masky mohou stále obsahovat mezery, často v místech, kde se nachází symboly, cílem další fáze je odstranit tyto mezery pomocí morfologických operací `refine_plate_matches()`
- i potom ale maska nemá obdelníkový tvar
- abych získal obdelníkovou masku, která nejlépe odpovídá tvaru a pozici značky, použiji **Canny**³ k vyhledání obrysu
- v obrysu pote vyhledám rovné segmenty (čáry) pomocí **HoughLinesP**⁴
- čáry sjednotím do skupin *partition* tak, že v každé skupině jsou čáry s podobným úhlem a umístěním
- každou skupinu podobných čar poté redukuji na jednu čáru
- po redukci zbývající čáry v každém ze dvou směrů (horizontální a vertikální) setřídíme podle pozice a vybereme nejvíce vzdálený pár jako okraje značky
- poté oba páry transformujeme tak, aby delší hrany byly paralelní a kratší vůči nim v pravém úhlu, čímž získáme obdélník
- nakonec vypočítáme průsečíky těchto čar, čímž získáme souřadnice souřadnice obdelníku – pozice značky
- poslední fází je extrakce **ROI** (region obrázku, který mě zajímá) se značkou spojena s transformací (rotací) značky tak, aby byla vodorovná **7.5**

³http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html Canny Edge Detector

⁴http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html Hough Line Transform



Obrázek 7.6: Řezy mezer mezi skupinami znaků

- abychom se vyhnuli ostrému přechodu na okraji (a potlačili stíny, které se tam často vyskytují), tak vytvoříme plynulý přechod rozmazání masky pomocí **GaussianBlur**
- poté použijeme techniku **alpha blending**⁵ k vytvoření obrázku oříznuté značky

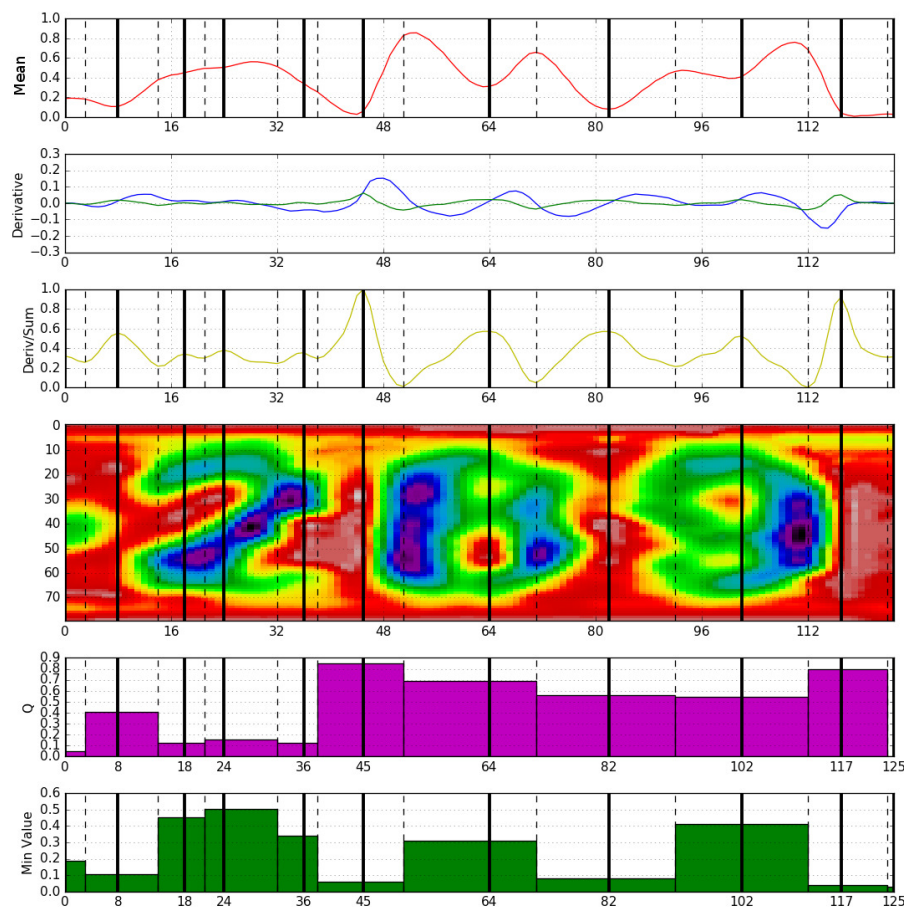
7.3 Segmentace značky na jednotlivé znaky

Cílem další fáze algoritmu je segmentace SPZ na jednotlivé znaky. Segmentaci provádím po částech s ohledem na to, že v pozdější fázi algoritmu budu porovnávat znaky se vzory pomocí **KNN**.

7.3.1 Hlavní mezery SPZ

- účelem první části *split_plate_level_1()* je rozdělit značku na dvě části obsahující souvislé sekvence znaků vyhledáním široké mezery nacházející se přibližně uprostřed značky
- abych potlačil vliv úzkých mezer mezi znaky a jiné lokální nepravidelnosti, rozostřím obraz v obou osách
- mezery jsou vyhledány analýzou vertikální projekce takto upraveného obrazu^[14]
- obrázek redukuji do jednoho rozměru obsahující průměrnou intenzitu každého sloupce
- v křivce opět potlačím úzké mezery a rozšířím vrcholy za použití morfologických operací 7.6
- porovnáním s prahovou hodnotou vytvořím masku identifikující jednotlivé znaky a mezery a poté s využitím prvního derivativu (funkce *filter2D()*) najdu hranice těchto oblastí
- jako střed poté vyberu nejširší mezeru, jejíž střed se nachází nejbližší středu značky

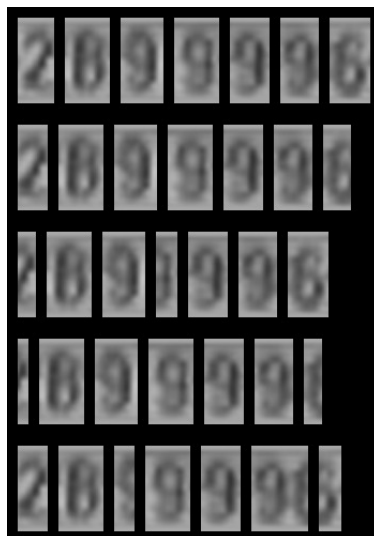
⁵https://en.wikipedia.org/wiki/Alpha_compositing Alpha blending



Obrázek 7.7: Potencionální místa řezu znaků

7.3.2 Mezery mezi znaky

- v další části opět hledám mezery, ale tentokrát pracuji se skupinou 4 nebo 3 znaků (*split_plate_level_2()*) vedle sebe a opět hledám mezery pomocí vertikální projekce
- v obou částech se snažím najít potencionální místa k řezu
- ke každému místu řezu přiřadím váhu vypočítanou na základě hodnot křivky v těchto místech (maximech a minimech křivky)
- potom vypočítám všechny možné kombinace těchto řezů
- každé kombinaci přiřadím celkovou váhu na základě vah jednotlivých řezů, jejich relativního rozmístění v obrázku a celkové šířky vyřiznutého dílu tak, aby nejvyšší váhu měly široké, pravidelně rozmístěné kombinace
- na grafu je také znázorněna projekce (průměrná hodnota sloupců), druhý derivativ a poměr druhého derivativu vůči křivce 7.7
- nakonec udělám kombinace všech řezů na základě váhy, které setřídím a vyberu nejlepší možnosti, v mém případě pět (*split_plate_level_3()*) 7.8



Obrázek 7.8: Možnosti řezů

7.4 Rozpoznání znaků

Po vyřazení možností, které nesplňují dané kritéria (např. šířka řezu), je každý sloupec (pozice znaku na SPZ) prohnán identifikačním algoritmem a je vybrána jediná možnost na základě nejvyššího výskytu, popřípadě váhy znaku. Toto proběhne pro levou a pravou stranu SPZ, výsledek spojíme a dostaneme jednoho z kandidátů na výslednou SPZ.

Rozpoznání znaků je následující:

- příslušný box se znakem zvětším, provedu vyrovnání histogramu a následné morfologické operace
- zjistím všechny kontury v boxu a pokud nevyhovují specifikaci (jsou příliš malé), tak je zahodím.
- následně začne probíhat porovnání se vzorovými znaky pomocí KNN klasifikátoru
 - porovnání je rozděleno podle typu SPZ na starý typ (podle obcí) a nový typ (dle krajů)
 - byly natrénovány znaky pro každou situaci, která může na značce nastat
 - díky tomu například stačí na posledních čtyřech pozicích porovnávat detekované znaky jen se vzorovými čísly viz. příklad níže
 - $p = knnx[35] \rightarrow findNearest(small_char_linear_1, tmp);$

7.5 (Re)Identifikace

Protože pro každé auto je vygenerováno více SPZ, je na základě podobného algoritmu jako pro výběr znaku, vybrána konečná SPZ pro dané auto na základě nejvyššího výskytu daného znaku v každé pozici SPZ.

Reidentifikace probíhá na principu popsaném výše v návrhu řešení a není třeba jeho popis znovu opakovat.

Kapitola 8

Vyhodnocení

Pro vyhodnocení aplikace byli pořízeny čtyři zhruba 15-minutová videa. Video byla natočena na **FULLHD** kameru s 50 snímků za sekundu, pořízena na dvou mostech v Brně (na každém vždy dvě videa). Souřadnice mostů jsou: **N49.226923, E16.599212** a **N49.242797, E16.589049**.

Jde mi o to, abych po identifikaci SPZ z jednoho videa na jednom mostě, mohl SPZ z videa na druhém mostě reidentifikovat. Jak bylo uvedeno výše, pro úspěšnou detekci byl použit klasifikátor **Haar Cascade**. Bylo natrénováno 1137 pozitivních snímků (snímků obsahující pouze SPZ). Tato práce cílila na české SPZ a od toho se odvíjela skladba typů pozitivních snímků. Do datasetu byli vloženi převážně standardní SPZ, starší bílé a novější s modrým eu pruhem v levé části značky.

Bylo také pořízeno 1185 negativních snímků obsahující vše až na SPZ (vozidla (bez SPZ), silnice, různá pozadí). Klasifikátor byl natrénován na 20 stage (částí - počet ovlivňující konečnou kvalitu klasifikátoru). Zpracování trvalo zhruba 4 dny čistého času.

Dále jsem pomocí vytvořeného skriptu v **Pythonu** natrénovat vzory jednotlivých znaků na SPZ, jejichž hodnoty jsem uložil do cvs formátu a využil při identifikaci znaků v systému. Celkem bylo vytvořeno 37 různých sad s různými kombinacemi znaků.

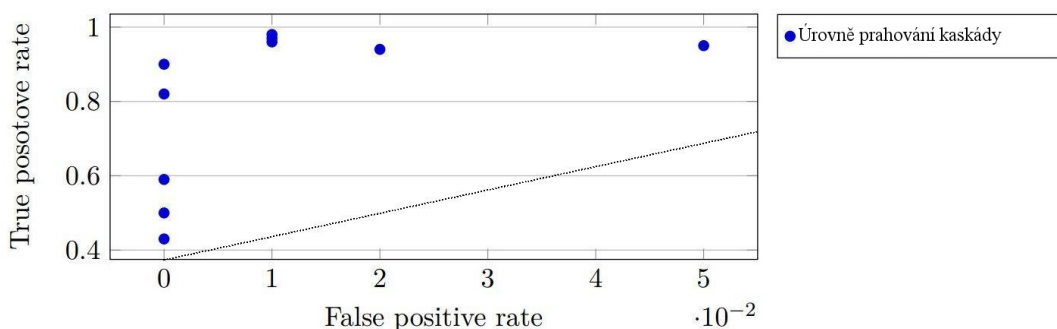
Poslední předem připravenou částí bylo vytvoření dvou masek pozadí (pro videa z obou mostů) a dvou masek, které redukuje místa zpracování videí jen na silnice.

8.1 Základní problém

Od začátku jsem se potýkal s jedním velkým problémem a tím, že kvalita SPZ je nízká, v horní části, tak nízká, že identifikace tam neměla smysl, proto vyhodnocení začalo probíhat, až od poloviny snímky (ve videu označeno dělicí čarou). Toto řešení má také obrovský vliv na rychlost aplikace (což je patrné v přiloženém videu).

Mnoho studií se zabývalo detekcí SPZ s rozlišením od 10px do 16px. Bohužel toto nízké rozlišení neumožňovalo následnou identifikaci a proto byly tyto studie zaměřené pouze na detekci[12].

Protože nízká kvalita SPZ znemožňuje v horní části snímku pořádnou identifikaci, celý systém provádí vyhodnocení jen v dolní části, kde dosahují jednotlivé znaky rozlišení okolo 20px. Při této úrovni rozlišení je identifikace v omezené míře možná. Kromě samotného rozlišení bylo potřeba zohlednit typ kamery, úhel sklonu k silnici při natáčení videí, vliv ztrátové komprese a občasné automatické ostření). Mnoho studií využívalo černobílou kameru, či IR přisvětlení, což jsou možnosti jak zlepšit kvalitu identifikace SPZ.[12]



Obrázek 8.1: ROC křivka

Vzhledem k zanedbatelnému množství modrých a převozních SPZ a neexistence SPZ na přání (kterých je na videu řádově jednotky) v začátcích implementace této aplikace, jsou tyto druhy SPZ ignorovány.

8.2 Výsledky

Na 1001 snímcích z videa bylo otestováno 10 úrovní tresholdu na natrénované kaskádě. Nastavováním těchto různých prahových hodnot hledáme na ROC křivce kompromis mezi množstvím falešně pozitivních a falešně negativních pozorování. Ideální ROC křivka nejprve stoupá téměř svisle vzhůru (úspěšnost se blíží 100 %, zatímco míra chyb zpočátku zůstává blízko 0 %), teprve pak se zvyšuje míra falešné positivity. Naproti tomu, když ROC křivka stoupá po úhlopříčce, znamená to, že každé zlepšení senzitivity je zapláceno stejně významným zhoršením specificity a test není dobře navržen.¹

Na základě tohoto předpokladu mohou říct, že ROC křivka je téměř ideální.

V tabulce 8.1 jsou zobrazeny výsledky z natočených videí včetně vstupních informací. Jak je vidět z výsledků, úroveň detekce se pohybuje od 91% do 98%, tudíž úroveň detekce je velmi úspěšná, což potvrzuje i výsledná ROC křivka. Průměrná úspěšnost správného rozpoznání SPZ je okolo 65%, vzhledem ke kvalitě SPZ je tento výsledek očekávatelný. Následná reidentifikace v našem případě dosahuje průměru okolo 67%, ale je přímo závislá na úspěšné identifikaci, lze tedy očekávat, že v případě vyšší kvality vstupních SPZ, dosáhne reidentifikace daleko lepších výsledků.

	Počet aut	Detekce	Rozpoznávání	Reidentifikace
A1	437	403 (92%)	231 (52%)	224/177 (79%)
A2	369	337 (91%)	181 (54%)	229/128 (55%)
B1	620	610 (98%)	505 (82%)	(viz A1)
B2	528	510 (96%)	374 (73%)	(viz A2)

Tabulka 8.1: Úspěšnost správného rozpoznání textu rozpoznávači.

Na základně nízké kvality SPZ je úroveň podobnosti jednotlivých znaků mnohem vyšší. V tabulce 8.2, která byla vytvořena na základě 403 značek z jednoho videa, kde se došlo

¹ČVUT FEL. ROC křivka <http://cyber.felk.cvut.cz/gerstner/teaching/zbd/biostat3.pdf>

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	H	I	J	K	L	M	N	O	P	R	S	T	U	V	X	Y	Z					
0	148		2	1	1	2	1		2	2																												
1	4	186	21	9			1	1		3																												
2			177		1				1	1																												
3	2	1		146	1		1	1		1									1																			
4	2	2	1	1	233			1		1																												
5			2			114	6																															
6			1		1	40	183			1										1																		
7			9			2		208																														
8	2			2	1	1	10		182	3		1																										
9	1	1	1	2		1		1	1	190																		3										
A				2	2		1				32	12			1		4																	1				
B									2	1		228			1																							
C	2					1							10																									
D	5	1												0										1														
E												1			5						1																	
F																																						
H																			0																			
I	1	1				2	1		3	1	2	4			1	1		4		4	4	3	2	1											1			
J																				10					1													
K					1							2			1						1				1													
L															1																							
M												3																										
N								1										1																				
O												3													1													
P				2											1											3												
R		2										1																										
S												1																										
T								1				2		1																					4			
U												1													1													
V		1										1																								1		
X																																						
Y																																						
Z			1					2																													6	

Tabulka 8.2: Tabulka záměn znaků

k porovnání vzorových značek se značkami detekovanými. Na tomto základě můžeme vyhodnotit všech kombinací znaků. Z tabulky je patrné, že nejvíce zaměňovaná dvojice znaků je 5 a 6. Zajímavé je naopak to, že byť písmeno B číslo 8 jsou si velmi podobné, k jejich záměnám naopak docházelo velmi zřídka. To je díky tomu, že algoritmus hlídá povolené kombinace hodnot písmen SPZ, tedy např. poslední čtyři znaky každé SPZ musí být čísla. Z tabulky je patrné, že její zdrojové video bylo pořízeno v Brně - co se týče písmen má znak B drtivou převahu.

Během testování² jsem zkoušel měřit čas, jak dlouho aplikaci trvá zpracovat video určité délky. Měřil jsem čas, jak dlouho trvá zpracování 15000 snímků (při 50FPS je to pět minut). Aplikace pokaždé v reálu běžela přibližně 98% procent video-času - tedy necelých pět minut. Z toho vyplývá, že aplikace je vhodná pro realtime využití, neboť je průměrně schopná analyzovat jeden snímek ještě před příchodem snímku dalšího.

²Testováno na počítači s CPU Intel Core i5 2.2GHz

Kapitola 9

Závěr

Cílem této práce bylo vytvořit a implementovat komplexní aplikaci, která umožní vyhledání SPZ vozidel ve videu a jejich případné porovnání (reidentifikaci vozidel) s jiným videem. Implementace proběhla s využitím knihovny OpenCV v jazyce C++.

Tuto aplikaci jsem následně otestoval na mnou vytvořených videích na množině mnou nátrénované množině referenčních dat. Experimenty ukázaly, že zatímco detekce vozidel jako takových je velmi úspěšná (průměrně 95%), převod SPZ na text je již méně úspěšný (průměrně 65%). Při experimentování s délkou rozpoznávání se ukázalo, že aplikace je vhodná pro použití v reálném čase, neboť minutový úsek videa analyzuje za přibližně 59 vteřin. To je umožněno mezi jinými i optimalizací hledání SPZ pouze v dolní polovině obrazu.

Co se týče budoucího vývoje, šla by přidat podpora pro simultánní analýzu více videí, což by umožnilo řešit reidentifikaci v reálném čase. Tím nejjednodušším způsobem by bylo použití socketů místo souborů. Úspěšnost rozpoznání SPZ by šla dále zlepšit systémem podobným policejním radarům - přidáním fotoaparátu, který by byl ovládán aplikací. Ta by počkala, až auto dojede na nevhodnější místo, a vyfotila jej.

Literatura

- [1] Detekce jízdy na červenou. <http://www.camea.cz/cz/dopravni-aplikace/dopravni-prestupky/detekce-jizdy-na-cervenou/>, accessed: 2016-01-24.
- [2] Inteligentní dopravní systémy. <http://projekt150.ha-vel.cz/node/93>, accessed: 2016-01-27.
- [3] Kdo může měřit rychlost?
<http://www.dopravni-pravo.cz/prekroceni-rychlosti/kdo-muze-merit-rychlost/>, accessed: 2016-01-24.
- [4] Ochrana osobních údajů. <http://www.camea.cz/cz/dopravni-aplikace/dopravni-prestupky/ochrana-osobnich-udaju/>, accessed: 2016-01-24.
- [5] Policejní radary. http://www.army.cz/images/id_8001_9000/8753/radar/k33.htm, accessed: 2016-01-24.
- [6] Poznávací značky automobilů v Českých zemích.
<http://www.feudal.cz/spz/html/2002-dosud.htm>, accessed: 2016-01-24.
- [7] Překročení rychlosti. <http://www.dopravni-pravo.cz/prekroceni-rychlosti/>, accessed: 2016-01-24.
- [8] Registr SPZ: vyhledávání registračních značek. <http://spz.penize.cz/>, accessed: 2016-01-24.
- [9] REGISTR AČNÍ ZNAČKY NA PŘÁNÍ OD LEDNA 2016.
<http://www.automoc.eu/registracni-znacky-na-prani/>, accessed: 2016-01-24.
- [10] Systémy pro silniční dopravu [online].
www.azd.cz/admin/files/Dokumenty/pdf/Produkty/Silnicni/Dopravni-detektory.pdf, [cit. 2016-01-27].
- [11] Acharay, T.; Ray, A. J.: *Image Processing - Principles and Applications*. Hoboken: John Wiley & Sons, 2015, iSBN 978-0-471-74578-5.
- [12] et al., C.-N. E. A.: License Plate Recognition From Still Images and Video Sequences: A Survey.
<https://pdfs.semanticscholar.org/41b6/887c8165e00dbd6788580543956436bfa2eb.pdf>, accessed: 2016-05-17.

- [13] Šráamek et al., J.: Získávání a analýza obrazové informace [online].
www.med.muni.cz/biofyz/Image/prezentace_4.pdf, [cit. 2016-01-23].
- [14] et al., T. N. B.: A Survey on Offline-Methods of Character Segmentation.
<http://seekdl.org/nm.php?id=25>, accessed: 2016-05-17.
- [15] Dobeš, M.: *Zpracování obrazu a algoritmy v C#*. BEN - technická literatura, 2008, ISBN 978-80-7300-233-6.
- [16] Fiřt, J.; Holota, R.: Digitalizace a zpracování obrazu [online].
<http://home.zcu.cz/holota5/publ/DigZprO.pdf>, [cit. 2016-01-27].
- [17] Hlaváč, V.: MATEMATICKÁ MORFOLOGIE [online].
<http://cmp.felk.cvut.cz/~hlavac/Public/TeachingLectures/BinMatMorfolCesky.pdf>, [cit. 2016-01-23].
- [18] Mikšík, O.: Praktické využití metod zpracování obrazu [online].
<http://soc.nidv.cz/data/2007/01-2.pdf>, [cit. 2016-01-27].
- [19] Sojka, E.: *Digitální zpracování a analýza obrazů*. Ostrava: VŠB-Technická univerzita, 2000, ISBN 80-7078-746-5.
- [20] Tandon, G.: Intelligent Transportation System.
<http://www.slideshare.net/gauravhtandon1/intelligent-transportation-system-56391402>, accessed: 2016-01-27.
- [21] Wikipedia: Detekční smyčka. https://cs.wikipedia.org/wiki/Detekční_smyčka, 2016, [Online; accessed 27-January-2016].
- [22] Wikipedia: Ekvalizace histogramu.
https://cs.wikipedia.org/wiki/Ekvalizace_histogramu, 2016, [Online; accessed 28-January-2016].
- [23] Wikipedia: Google Street View.
https://cs.wikipedia.org/wiki/Google_Street_View, 2016, [Online; accessed 24-January-2016].
- [24] Wikipedia: Státní poznávací značky v Česku.
https://cs.wikipedia.org/wiki/Státní_poznávací_značky_v_Česku, 2016, [Online; accessed 24-January-2016].
- [25] Woesler, R.: *Fast Extraction of Traffic Parameters and Reidentification of Vehicles From Video Data*. Intelligent Transportation Systems, 2003, ISBN 0-7803-8125-4.
- [26] Šonka, M.; Hlaváč, V.: *Počítačové vidění*. Grada, Praha, 1992, ISBN 80-85424-67-3.
- [27] Španěl, M.; Beran, V.: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ [online].
<http://www.fit.vutbr.cz/~spanel/segmentace/>, [cit. 2016-01-07].

Přílohy

Seznam příloh

A Obsah DVD

37

Příloha A

Obsah DVD

Příložené DVD obsahuje zdrojové kódy programu, testovací videa, prezentační video, plakát, technickou zprávu a README.