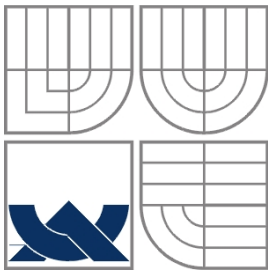
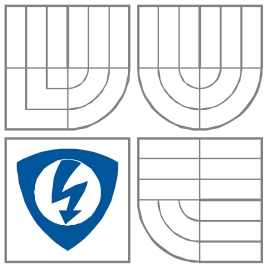


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MPOV
PROJEKT 2 – HUSTOTA PROVOZU

AUTOR PRÁCE — Bc. Milan Kořínek
Bc. Daniel Krykorka

VEDOUCÍ PRÁCE
SUPERVISOR — Ing. Čip Pavel

BRNO 2012

Obsah

1 ÚVOD

V současně době je snaha většinu činností automatizovat. Tomuto trendu se nevyhnula ani doprava. Zejména ve městech a na hlavních tratích je důležité mít přehled o dopravní hustotě během dne a dne v týdnu. Tato znalost umožní efektivněji nastavit semafory, dopravní značení, vybudovat efektivní obchvaty a neposledně tyto informace zpřístupnit samotným řidičům, kteří se pak mohou vyhnout dopravní zácpě.

Náš projekt má výrazně jednodušší zadání oproti obecným případům, díky čemuž jsme mohli některé algoritmy zjednodušit a zefektivnit pro zadané videosekvence oproti obecným případům. Kamera je v našem případě statická, video je natočené pouze během dne, výrazný pohyb je tvořen pouze projíždějícími vozidly apod. Při tvorbě algoritmů pro detekci pohybu v obraze a další zpracování jsme vycházeli z [1], [2] a přednášek MPOV.

2 PARAMETRY VOZOVKY

Než jsme mohli přistoupit k detekci vozidel v obrazu, tak jsme nejdříve potřebovali zjistit, kde se nachází jízdní pruhy. Znalost lokace jízdních pruhů nám, pak umožňuje eliminovat rušivý pohyb zeleně a počítat vozidla v jednotlivých jízdních pruzích.

2.1 Možnosti detekce jízdních pruhů

2.1.1 Manuálně

Nejdříve se vytvoří model pozadí scény bez vozidel. Model je poté zobrazen uživateli, jenž by pak měl např. barevně zakreslit jednotlivé jízdní pruhy. Takto barevně odlišené jízdní pruhy lze snadno programově zpracovat.

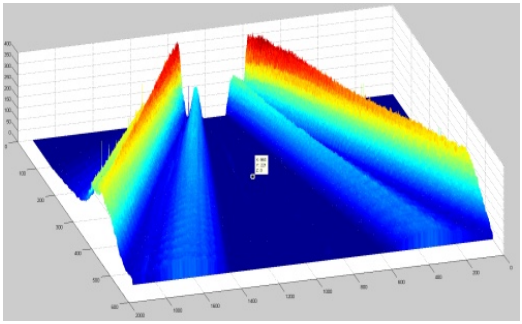
Manuální metoda rozeznání jízdních pruhů je nejjednodušší, ale zároveň nejméně praktická a efektivní. Může se vyplatit pro případy, kdy je kamera napevno umístěná a tedy jízdní pruhy se stále nacházejí na stejných místech. Další výhodou je možnost snadno nadefinovat odbočovací a odstavné pruhy.

2.1.2 Segmentace prahováním

Stejně jako u manuální metody se nejdříve vytvoří model pozadí. Poté se vy-segmentují jednotlivé jízdní pruhy. Tuto metodu jsme použili, podrobnější popis je v kapitole 2.2.

Použití segmentace je sofistikovanější než u manuální metody. Využití segmentace je omezeno na případy, kde lze jednoznačně odlišit jízdní pruhy od ostatních objektů ve scéně.

2.1.3 Pohyb v obraze



Princip metody spočívá v detekci významného pohybu v obraze v daném časovém úseku. Poté je možné identifikovat z obrazu jednotlivé jízdní pruhy.

Jedná se o nejobecnější metodu pro detekci jízdních pruhů. Výhodou je možnost detekce jízdních pruhů za mnoha různých podmínek (např. světelné podmínky: den – noc). Problém může nastat pokud v daném jízdním pruhu neprojede žádné nebo minimum vozidel. V opačném případě, pokud se ve scéně vyskytuje jiný pohyb než vozidel v hledaných jízdních pruzích např. chodci jdoucí podél vozovky na chodníku.

2.2 Detekce a parametry jízdních pruhů

Detekce hranic jízdních pruhů je realizováno funkcí `GetTrafficLane`. Funkce přebírá dva parametry, kde první `bcg` je barevný model pozadí a druhý parametr `show` slouží pouze pro diagnostické účely (zobrazení histogramu a vysegmentované scény). Návratová hodnota `roadLane` je struktura obsahující hranice a binární masky jednotlivých jízdních pruhů. Popis funkce rozebrán v následujících kapitolách.

```
function [roadLane] = GetTrafficLane(bcg,show)
% Function find traffic lanes in image
% bcg - color picture of background
% show - show histogram, original and segmented road
% roadLane - structure:
%         .left(x,y,line) - boundary of left two-lane road, two
%                           points for every line
%         .right(x,y,line) - boundary of right two-lane road, two
%                            points for every line
%         .surfLeft(:,:,1) - left two-lane road - mask of left lane
%         .surfLeft(:,:,2) - left two-lane road - mask of right lane
%         .surfRight(:,:,1) - right two-lane road - mask of left lane
%         .surfRight(:,:,2) - right two-lane road - mask of right lane
```

2.2.1 Model pozadí

Pro model pozadí používáme funkci `get_background`. Funkce provádí průměrování `n` snímků z videa (parametr `video`). Výsledný model pozadí je vrácen v

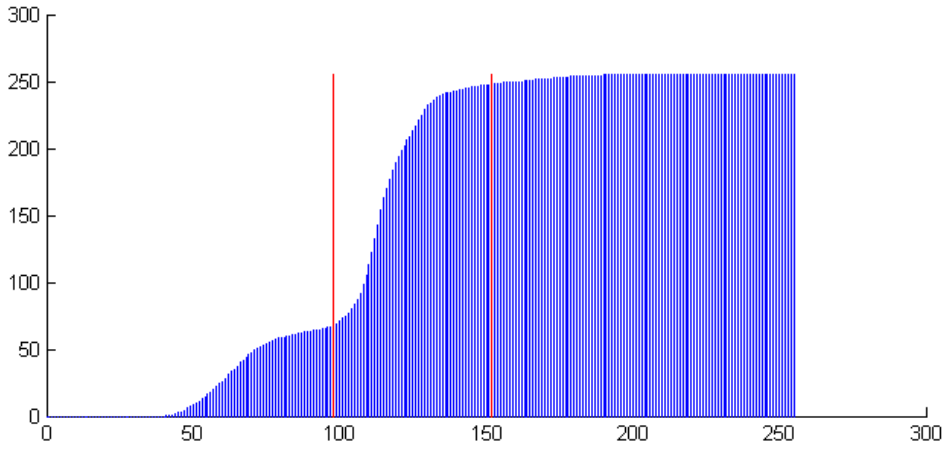
parametru `bcg`. Pro vytvoření modelu používáme 50 snímků. Podrobnější rozbor tvorby modelů lze nalézt v [1].

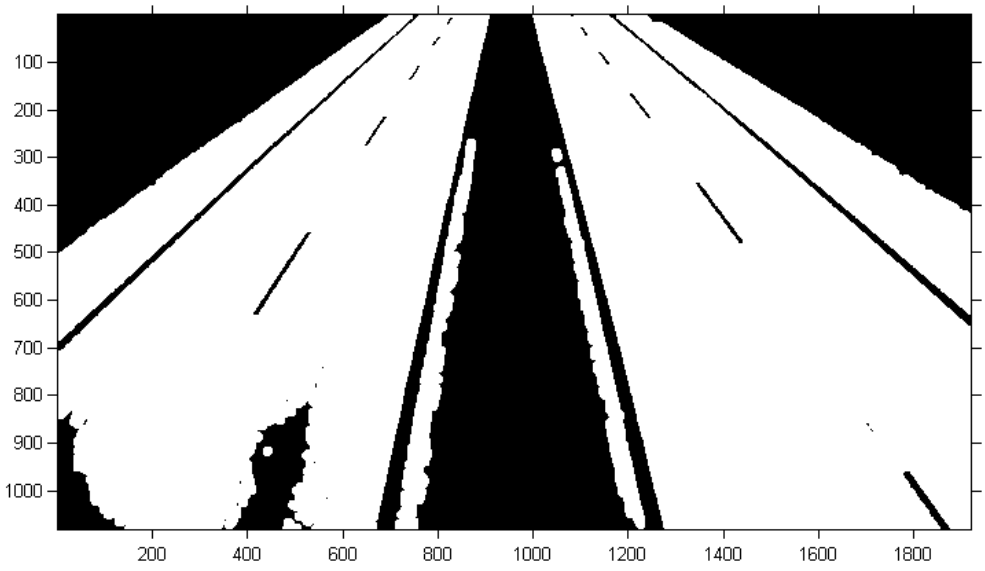
```
function [ bcg ] = get_background( video, n )
% Function is trying to get background of the video scene
%     by averaging
% video - video
% n - count of averaged frames
% bcg – model of background
```

2.2.2 Segmentace prahováním

Všechny modely pozadí pro zadané videosekvence mají stejnou kompozici. Nejvýraznějším objektem je vozovka. Abychom získali, co nejlepší výsledky segmentace, tak jsme použili pouze modrou složku obrazu, čímž se zvýšil kontrast mezi vozovkou a okolím.

Z analýzy kumulovaného histogramu jsme zjistili, že vozovce náleží určitý spojitý interval kumulovaného histogramu, pak tedy jsme mohli použít prahování s dvěma prahy. Nejdříve jsme ručně odhadli vhodné umístění prahů . Tuto znalost jsme poté algoritmizovali (za základě změny první derivace histogramu), čímž jsme potom mohli provést kvalitní segmentaci vozovky u všech zadaných videosekvencí. Z obrazu jsme nakonec vyfiltrovaly malé objekty.





Pro normalizovaný kumulovaný histogram používáme funkci `ImageHistogramSum`. Vstupní parametr je obraz `I` a funkce vrací histogram `H`.

```
function [H] = ImageHistogramSum(I)
% Cumulative histogram
% I - image
% H - vector - normalize cumulative histogram
```

2.2.3 Parametry vozovky

Aplikací funkcí `regionprops a sort` jsme získali jednotlivé objekty v obraze seřazené podle velikosti. Největší oblasti jsme identifikovali jako vozovky pro jeden a druhý směr jízdy. Funkce `GetRoadBoundary` slouží pro zjištění parametrů vozovky, respektive hranic jednotlivých jízdních pruhů. Funkci je

předán parametr `road`, jenž obsahuje vysegmentovanou vozovku pro jeden nebo druhý směr jízdy. Funkce vrátí hranice jízdních pruhů `bound`.

```
function [bound] = GetRoadBoundary(road)
% Function find and calculate boundary of two-lane road
% imBcg - color background of traffic situation
% bound - return 3 lines - boundary of lane
```

V dalších kroku se vygenerují masky pro jednotlivé jízdní pruhy. Případně, pokud požadováno, se zobrazí histogram s vysegmentovaná vozovka. Nakonec funkce `GetTrafficLane` vrátí parametry vozovky a je ukončena.

3 DETEKCE AUTOMOBILŮ

3.1 Model pozadí

Model pozadí může buď být statický nebo dynamický. U statického modelu se předpokládá, že se scéna příliš nemění až na žádaný pohyb vozidel. Tedy předpokládá konstantní osvětlení, statickou kameru apod. Takový model se vytvoří pouze jednou z určité části videa. Naopak dynamický model umožňuje eliminovat jmenované nedostatky.

V našem případě jsme nejdříve vytvořili statický model pozadí, který potom aktualizujeme podle následujícího kódu:

```
D = uint8(I ./ bcg); % rozdíl snímku od pozadí
M = uint8(D < 0.9);  % binární maska rozdílných pixelů
bcg = bcg + double((0.15 * (1 - M) + 0.03 * M) .* D); % model pozadí
```

Pomocí tohoto postupu se model pozadí přizpůsobí změnám osvětlení, dokáže potlačit stín od vozidel a jiné odchylky. Metoda je podrobněji popsána v [4].

3.2 Detekce pohybu v obraze

Jednotlivé vozy detekujeme pomocí odčítání aktuálního snímku od modelu pozadí a to tak, že jednotlivé barevné vrstvy od sebe odečteme hned provedeme prahování, podle předem stanovené minimální odchylky. Výsledný obraz dále filtrujeme s sledujeme pouze plochy, které mají plochu větší než 2500 pixlů.

Jako referenci bereme centrum této plochy. Popřípadě pokud se nám plochy od více voyů "slijí" dohromady, pak bere střed predikovaný pomocí Kalmanova filtru.

3.3 Predikce trajektorie automobilů – Kalmanův filtr

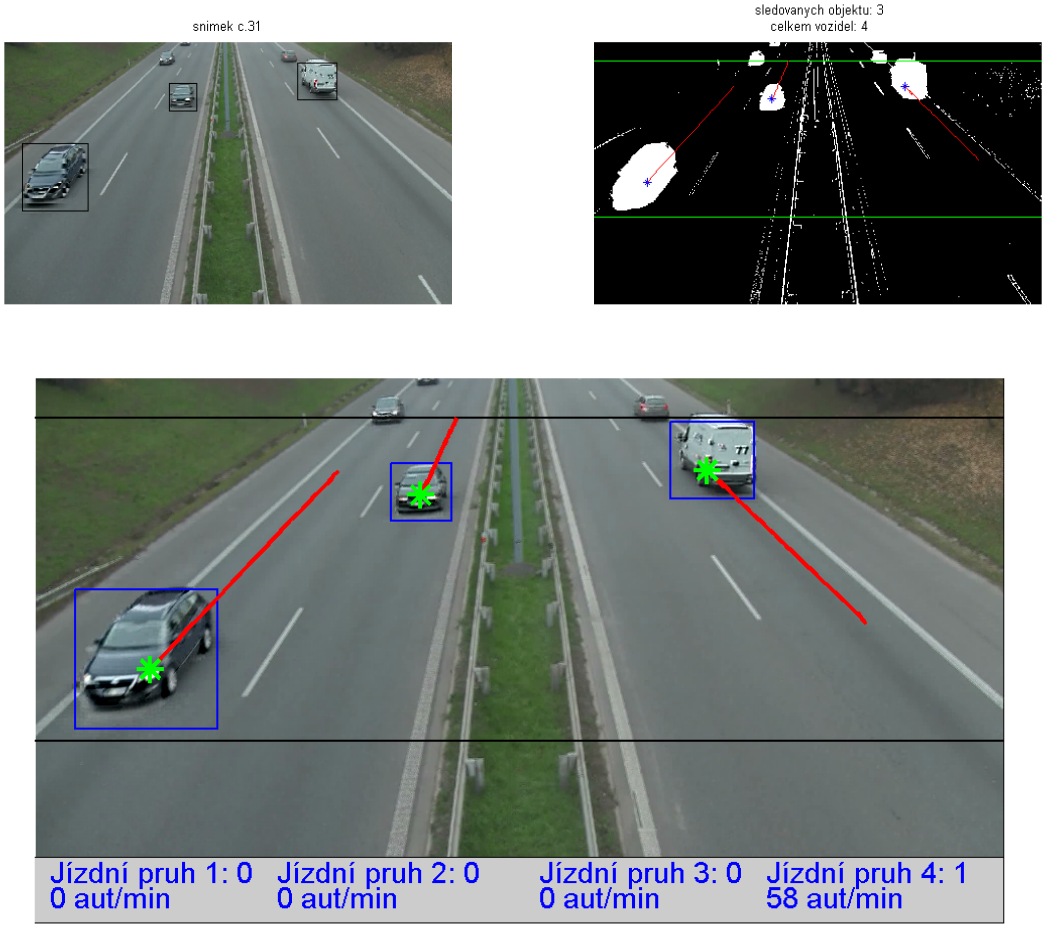
Jedná se o adaptivní filtr, jehož výpočet se dá rozdělit do dvou kroků.

V prvním kroce pomocí předem stanoveného modelu predikujeme polohu v následujícím snímku. V druhém kroku sledujeme jak se vypočtená hodnota odchylovala od reálné a poupravíme korelační matice.

Jednotlivé matice potřebné pro výpočet si držíme ve struktuře *cars* spolu se záznamem pozic pro následnou analýzu pohybu.

3.4 Hustota dopravy

Pokud vůz opustí sledovanou oblast, je analyzována jeho stopa a jednotlivým pruhům ve kterých se nacházel je přičtena 1. Výsledek poté dělíme konstantou získanou tak, že jsme hodnotu fsp podělili 1800. Výsledná hodnota poté odpovídá průměru za minutu.

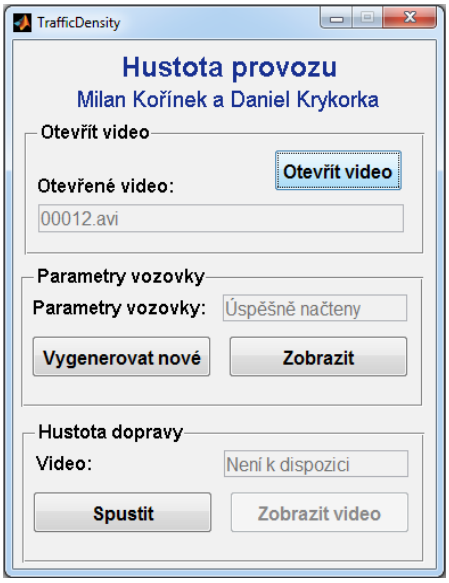


Obr. 1: Výstupní okna a snímek videa

Na obrázku nahoře je vyobrazeno jak postupuje při počítaní jednotlivých vozů a jak je sledujeme.

4 GUI APLIKACE

4.1 Popis GUI



Aplikace je napsaná v Matblab Guide [3]. Nejdříve uživatel musí otevřít video pro zpracování (tlačítko Otevřít video). Pokud existují, již vygenerované parametry vozovky, pak se automaticky načtou. V opačném případě lze vygenerovat parametry nové (tlačítko Vygenerovat nové). Vygenerované parametry lze zobrazit tlačítkem Zobrazit. Analýza hustoty dopravy se spustí tlačítkem Spustit. Průběh a zbývající doba analýzy je zobrazována v samostatném okně. Video s informacemi o dopravní hustotě se může spustit tlačítkem Zobrazit video.

4.2 Výstupní soubory

Všechny soubory jsou ukládány do adresáře s vybraným videem pro zpracování. Má-li video název souboru *nazev.avi*. Pak jsou vygenerované následující soubory:

- *nazev.png* – Barevný model pozadí.
- *nazev.mat* – Parametry vozovky (hranice a binární masky jednotlivých jízdních pruhů).
- *nazev_out.avi* – Vygenerované video.

5 ZÁVĚR

Podařilo se nám vcelku dobře detekovat jednotlivé vozy a naprogramovat metodu, která sleduje a zároveň předpovídá jejich pozici. Tento program dokáže sledovat i vozy, které jsou v zkrytu jiného vozu.

Problémy nastávají pokud se nám vůz rozpadne na více částí, což je způsobeno převážně odrazy světla. Problém by se dal řešit sledováním okrajů objektů a jejich následnou interpolací, ale na podrobnější analýzu, implementaci a odladění by bylo třeba se zabývat mnohem delší dobu, než bylo určeno na celý projekt.

Při zkompilování m-filu v poslední verzi MATLABu (2012b) jsme zjistili, že časově nejnáročnější operace při výpočtu bylo zjištění odchylek od modelu pozadí. Celý výpočet jednoho snímku zde trval cca 1s.

tab.1: statistika pro video 00012.avi. První sloupec odpovídá originálu a druhý výpočtu našho programu.

0-15s		15-30s		30-45s		45-60s	
8	9	6	5	4	4	3	4
4	3	6	6	2	2	5	5
2	3	2	2	0	0	1	1
8	7	5	3	3	3	5	6

tab.2: statistika pro video 00013.avi. První sloupec odpovídá originálu a druhý výpočtu našho programu.

0-15s		15-30s		30-45s		45-60s		60-75s		75-90s		90-105s		105-120s		120-138s	
6	5	2	2	4	8	5	5	6	7	6	7	4	5	7	5	7	7
1	1	0	0	7	8	3	4	3	3	2	2	2	2	3	3	2	1
1	1	0	0	2	2	1	1	1	1	3	3	3	3	5	5	3	2
6	6	4	4	6	5	10	9	5	5	4	4	4	4	6	6	7	9

Až na pár výjimek se výsledky našeho programu shodovai s reaitou. Výsledek by se dal výrazně zlepšit, pokud by jsme podrobněji analyzovaly přiřazování centroidů.

Výsledky výstupu se dají shlédnout na: http://youtu.be/cfWSu8nzPJY a http://youtu.be/zMgMjLFeGWQ . Stáhnout se pak dají na http://ulozto.cz/xbNibRE/projekt2-mpov-zip . Vývoj našeho projektu se dá shlédnout na https://github.com/Aqarel/MPOV_traffic_density/commits .

6 LITERATURA

[1] KOZINA, L. *Detekce a počítání automobilů v obraze (videodetekce)*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 60 s. Vedoucí diplomové práce Ing. Peter Honec, Ph.D.

[2] KOLLER, Dieter; WEBER, Joseph; MALIK Jitendra: Robust Multiple Car Tracking with Occlusion Reasoning. California PATH Program, Institute of Trasportion Stuides University of California, Berkley. January 1994. ISSN 1055-1417.

[3] Citation. In: Creating Graphical User Interfaces, MATLAB R2012b [online]. The MathWorks, Inc, 2012- , last modif. September 2012 [cit. 2011-12-11]. Dostupné na URL: http://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf

[4] Dieter Koller, Joseph Weber, Jitendra Malik: *Robust Multiple Car Tracking with Occlusion Reasoning.* Dostupné na URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.9651&rep=rep1&type=pdf