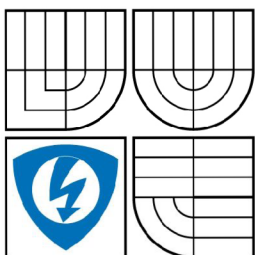


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

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

DETEKCE A ROZPOZNÁVÁNÍ DOPRAVNÍCH ZNAČEK

TRAFFIC SIGNS DETECTION AND RECOGNITION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

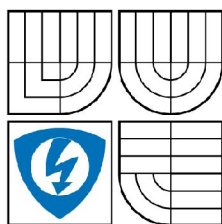
Bc. PAVEL ČÍP

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. KAREL HORÁK, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Pavel Číp

ID: 84084

Ročník: 2

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Detekce a rozpoznávání dopravních značek

POKYNY PRO VYPRACOVÁNÍ:

Studium metod detekce geometrických tvarů a jejich implementace pro rozpoznání dopravních značek v obrazových datech. Vstupem je video-sekvence jízdy vozidla v městském i mimo-městském prostředí, výstupem jsou obrazy detekovaných značek z databáze. Případné znázornění nalezených značek na displeji v zorném poli řidiče má asistenční povahu, musí být tedy prováděno v reálném čase.

DOPORUČENÁ LITERATURA:

- [1] HLAVÁČ, Václav, SEDLÁČEK, Miloš. Zpracování signálů a obrazů. Praha : ČVUT, 2005. 255 s. ISBN 80-01-03110-1.
- [2] SCHLESINGER, Michail I., HLAVÁČ, Václav. Deset přednášek z teorie statistického a strukturního rozpoznávání. Praha : ČVUT, 1999. 521 s. ISBN 80-01-01998-5.
- [3] SONKA, Milan, HLAVAC, Vaclav, BOYLE, Roger. Image Processing, Analysis and Machine Vision. 3rd edition. Toronto : Thomson, 2008. 829 s. ISBN 978-0-495-08252-1.
- [4] SVOBODA, Tomas, KYBIC, Jan, HLAVAC, Vaclav. Image Processing, Analysis and Machine Vision : A MATLAB Companion. Toronto : Thomson, 2008. 255 s. ISBN 978-0-495-29595-2.
- [5] VERNON, David. Machine Vision : Automated Visual Inspection and Robot Vision. Hemel Hempstead : Prentice Hall International (UK) Ltd., 1991. 260 s. ISBN 0-13-543398-3.

Termín zadání: 9.2.2009

Termín odevzdání: 25.5.2009

Vedoucí práce: Ing. Karel Horák, Ph.D.

prof. Ing. Pavel Jura, CSc.

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 § 152 trestního zákona č. 140/1961 Sb.

Vysoké učení technické v Brně
Fakulta elektrotechniky a komunikačních technologií
Ústav automatizace a měřicí techniky

Detekce a rozpoznávání dopravních značek

Diplomová práce

Studijní obor: Kybernetika, automatizace a měření
Student: Bc. Pavel Číp
Vedoucí: Ing. Karel Horák, Ph.D.

Abstrakt :

Práce se zabývá diskuzí nad metodami detekce a rozpoznávání dopravních značek v městském i mimoměstském prostředí. Předpokladem pro realizaci systému je zabudovaná kamera, obvykle ve zpětném zrcátku automobilu, snímající scénu před automobilem. Její obrazová data jsou posléze zpracována připojeným PC, kde dochází k převodu dat na informace a jejich vyhodnocení. O případné nalezené značce je řidič vizuálně či akusticky upozorněn.

Úloha vedoucí k úspěšnému cíli je rozdělena do čtyřech samostatných bloků. V první části je předzpracování obrazu jako takového. Pracujeme s barevným obrazem a s využitím znalosti o barevnosti dopravních značek v České republice, lze provést barevnou segmentaci žádaných intervalů. Druhým krokem je detekce geometrických tvarů odpovídajících dopravním značkám v segmentovaných datech. Krokem číslo tři je rozpoznání vnitřního piktogramu a jeho nalezení v databázi. Posledním krokem je vizuální výstup zobrazením nalezené dopravní značky.

Práce byla zpracována tak, aby byla zajištěna detekce všech důležitých dopravních značení ve třech základních barevných kombinacích platných dle Ministerstva dopravy České republiky. Výsledkem je zdrojový kód pro program MATLAB.

Klíčová slova:

detekce a rozpoznávání dopravních značek, dopravní značení, dopravní značka, neuronové sítě, zpracování obrazu, barevná segmentace, detekce geometrických tvarů, genetické algoritmy, MATLAB

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control, Measurement and Instrumentation

Traffic Signs Detection and Recognition

Diploma Thesis

Specialization of study: Cybernetics, control and Measurements
Student: Bc. Pavel Číp
Supervisor: Ing. Karel Horák, Ph.D.

Abstract:

The thesis deals with traffic sign detection and recognition in the urban environment and outside the town. A precondition for implementation of the system is built-in camera, usually in a car rear-view mirror. The camera scans the scene before the vehicle. The image data are transferred to the connected PC, where the data are transformation to information and evaluations. If the sign was detected the system is visually warned the driver.

For a successful goal is divided into four separate blocks. The first part is the preparing of the image data. There are color segmentation with knowledge of color combination traffic signs in Czech Republic. Second part is deals with shape detection in segmentation image. Part number three is deals with recognition of inner pictogram and its finding in the image database. The final part is the visual output of displaying founded traffic signs.

The thesis has been prepared so as to ensure detection of all relevant traffic signs in three basic color combinations according to existing by Decree of Ministry of Transport of Czech Republic. The result is the source code for the program MATLAB. .

Keywords:

traffic sign detection and recognition, traffic sign, neural network, computer vision, color segmentation, geometric shape detection, genetic algorithm, MATLAB

BIBLIOGRAFICKÁ CITACE DÍLA

ČÍP, P. *Detekce a rozpoznávání dopravních značek*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 91 s. Vedoucí diplomové práce Ing. Karel Horák, Ph.D.

PROHLÁŠENÍ

„Prohlašuji, že svou bakalářskou práci na téma "Systém automatického zaostřování" jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **25. května 2009**

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Karlu Horákovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **25. května 2009**

.....

podpis autora

OBSAH

1. ÚVOD	13
2. DOPRAVNÍ ZNAČENÍ	15
2.1 Detekce dopravních značek	16
3. SEGMENTACE OBLASTÍ	18
3.1 Barevné modely [3]	19
3.1.1 RGB	19
3.1.2 HSV	19
3.1.3 YCbCr	21
3.1.4 Převod RGB na YCbCr	21
3.2 Reprezentanti dopravních značek	22
3.3 Ukázka segmentace	23
3.3.1 Segmentace v RGB	23
3.3.2 Segmentace v HSV	25
3.3.3 Segmentace v YCbCr	26
3.4 Shrnutí metod, závěry	28
4. KLASIFIKACE OBLASTÍ	30
4.1 Popis oblastí pomocí příznaků	30
4.1.1 Extrakce, selekce příznaků	31
4.1.2 Identifikace oblastí	32
4.1.3 Radiometrické deskriptory založené na regionech	32
4.2 Klasifikace pomocí příznaků	34
4.3 Klasifikace kruhu pomocí genetického algoritmu	38
4.4 Vzdálenostní transformace - Distance transform (DT)	41
4.4.1 Popis metody D. M. Gavriila	43
4.5 Popis metody Hassan Shojanía [8]	46
4.5.1 Detekce trojúhelníku	48
4.5.2 Detekce kruhu, čtverce	49
4.5.3 Detekce osmiúhelníku	49

4.6 Detekce pomocí houghových transformací	51
4.6.1 Detekce přímek	51
4.7 Porovnání se šablonou.....	52
5. NORMOVÁNÍ OBLASTI	55
6. IDENTIFIKACE PIKTOGRAMU.....	58
6.1 Umělé neuronové sítě	59
6.1.1 Neuronová síť – M vstupů, 1 výstup, N hodnot.....	60
6.1.2 Neuronová síť – M vstupů, N výstupů, 1 hodnota	61
6.2 Momentové invarianty	62
6.2.1 Neuronová síť s momentovými invarianty	64
6.2.2 Statistické zpracování momentových invariantů.....	65
6.3 Porovnání se vzorem.....	68
6.4 Porovnání metod.....	69
7. SOFTWAREVÉ ŘEŠENÍ	71
8. NÁVRH SNÍMACÍ SOUSTAVY	73
9. VÝSLEDKY MĚŘENÍ A REALIZACE.....	75
9.1 Optimální nastavení snímacího zařízení	75
9.2 Ukázka detekce.....	78
9.3 Omezující podmínky.....	81
10. ZÁVĚR	82
11. POUŽITÉ ZDROJE	84

SEZNAM OBRÁZKŮ

Obrázek 2-1: Blokové schéma principu detekce značek – návrh řešení	16
Obrázek 3-1: RGB barevný model [3].....	19
Obrázek 3-2: HSV barevný model [3].....	20
Obrázek 3-3: Ukázka segmentace červené barvy v RGB prostoru s proměnlivými prahy: a) originální obrázek, b) práh větší, c) práh menší.....	24
Obrázek 3-4: Reprezentanti dopravních značek – HSV, histogram.....	25
Obrázek 3-5: Reprezentanti barevných složek – HSV prostor, jehlan	26
Obrázek 3-6: Reprezentanti dopravních značek – četnost v CbCr prostoru, vlevo červené značky, vpravo modré značky, dole žluté značky.	26
Obrázek 3-7: Segmentace YCbCr – průběh jasu reprezentanta.....	27
Obrázek 3-8: Segmentace barevných oblastí v RGB, YCbCr a HSV prostoru – a) originální snímek, b) RGB segmentace, c) YCbCr segmentace, d) HSV segmentace	28
Obrázek 4-1: Výpočet příznakového vektoru X [9]	31
Obrázek 4-2: Příklad příznakového prostoru [9].....	32
Obrázek 4-3: Ukázka identifikace oblastí.....	32
Obrázek 4-4: Ukázka trénování množiny - ČTVEREC	35
Obrázek 4-5: Klasifikace tvaru – vývojový diagram.....	37
Obrázek 4-6: Genetický algoritmus pro detekci kruhu: a) originální snímek, b) segmentace červené barvy, c) po aplikaci kruhového filtru, d) detekovaná elipsa	40
Obrázek 4-7: Vzdálenostní transformace – maticová ukázka [5]	41
Obrázek 4-8: Vzdálenostní transformace – ukázka různých metrik [8]	43
Obrázek 4-9: Metoda podle D. M. Gavriila [4].....	44
Obrázek 4-10: D. M. Gavriilla - a) originální obrázek, b) šablona, c) binární snímek obrázku, d) DT transformace [4]	45
Obrázek 4-11: D.M.Gavriila – výsledky detekce [4]	46

Obrázek 4-12: Hassan Shojanian – metoda barevné segmentace a vyhledávání rohů [8]	47
.....	
Obrázek 4-13: Hassan Shojanian – a) maska C1 pro detekci rohu, b) maska Y1 [8] ..	48
Obrázek 4-14: Hassan Shojanian – a) maska pro levý horní roh trojúhelníka, b) její aproximace [8]	48
Obrázek 4-15: Hassan Shojanian – a) detekce kruhu, b) detekce čtverce [8].....	49
Obrázek 4-16: Hassan – a) detekce trojúhelníka, b) detekce osmiúhelníku [8].....	50
Obrázek 4-17: Hassan Shojanian – ukázka výsledku detekce dopravních značek a) originál s vyznačenými výsledky, b) barevná segmentace [8]	50
Obrázek 4-18: Houghova transformace: mapování přímky na křivky	51
Obrázek 4-19: Houghova transformace: ukázka detekce přímek	52
Obrázek 4-20: Ukázka trénování množiny šablon, v horní řadě uvedeny patvary	54
Obrázek 5-1: Ukázka normování nalezené elipsy	56
Obrázek 5-2: Normované oblasti 100x100 px	57
Obrázek 6-1: Ukázka trénovací množiny jednoho vzoru	58
Obrázek 6-2: Umělá neuronová síť	59
Obrázek 6-3: Umělá neuronová síť – učení s učitelem	60
Obrázek 6-4: Neuronové síť: regresní funkce na výstupu sítě M-1-N	61
Obrázek 6-5: RapidMiner – struktura programu pro zpracování dat	66
Obrázek 6-6: RapidMiner – vlevo je výstup W-RandomTree z programu RapidMiner, vpravo transformovaný kód pro MATLAB pomocí JavaScriptu .	67
Obrázek 8-1: Schéma DAS	73
Obrázek 8-2: Koncept se třemi kamerami, Gavrilla [4]	74
Obrázek 9-1: Měřicí zařízení	75
Obrázek 9-2: GRAF: průběh šířka značky při změně vzdálenosti automobilu	76
Obrázek 9-3: GRAF: průběh šířka značky vlivem změny ohniskové vzdálenosti objektivu.....	76
Obrázek 9-4: Ukázka měřicích dat pro změnu ohniskové vzdálenost.....	77
Obrázek 9-5: Ukázka výstupu při detekci na testovacích snímcích	79
Obrázek 9-6: Ukázka výstupních snímků jak je algoritmus reprezentuje řidiči	80
Obrázek 13-1: Dopravní značení – výstražné značky [10]	88

Obrázek 13-2: Dopravní značení – Zákazové značky [10].....	88
Obrázek 13-3: Dopravní značení – Příkazové značky [10]	89
Obrázek 13-4: Dopravní značení – Informativní značky [10]	89
Obrázek 13-5: Dopravní značení – značky upravující přednost vozidel [10].....	89
Obrázek 13-6: Dopravní značení – Informační značky provozní [10]	89
Obrázek 13-7: Dopravní značení – Informativní značky [10]	90
Obrázek 13-8: Tvary, které systém rozlišuje – číslo odpovídá proměnné ID_tvar....	90

SEZNAM TABULEK

tabulka 1: Přehled barevných reprezentantů dopravního značení	22
tabulka 2: Porovnání segmentačních metod.....	29
tabulka 3: Elipsa – využití symetrie podle středu [16]	39
tabulka 4: Vzdálenostní transformace – přehled metrických vzdáleností [5].....	42
tabulka 5: Porovnání metod pro rozpoznání piktogramu.....	70
tabulka 6: Přehled predikce pomocí rozhodovacího stromu v programu RapidMiner, přesnost 84.48%.....	91

1. ÚVOD

Počítačové vidění dnes zasahuje do všech odvětví vědy a techniky, a tak není divu, že si své místo našlo i v automobilovém průmyslu. Dnešní počítačové systémy hlídají v automobilu všechno: zda řidič nepožil před jízdou alkohol, zda je teplota uvnitř dostatečná, zda nejsou podhuštěny pneumatiky, případně zaznamenávají trasu řidiče a odesílají ji na centrálu.

Systém se ve zkratce nazývají DAS – driving assistance systém – podpůrné systémy pro řidiče. Pokud palubní počítač rozšíříme o možnost „vidět“ co je před či za automobilem, ještě více se zvýší bezpečnost samotného řidiče a o to právě těmto DAS jde především.

S vývojem kvalitnějších snímacích prvků, rychlejších procesorů a vývoje nových postupů, lze automobil udělat vidoucím a tak jsou známy v dnešní době i systémy, které podporují noční vidění, detekci chodce na přechodu, sledování středové čáry na vozovce či detekci dopravních značení.

A právě o posledním zmiňovaném bloku bude pojednávat tato práce. Systém, který „sleduje“ trasu před automobilem a „čte“ dopravní značení dnes aplikuje do svých automobilů například automobilka Opel s názvem OpelEYE, podobným tématem se zabýval i ing. Pavel Paclík z ČVUT. Algoritmy pro zpracování dat musejí být dostatečně robustní, aby byly spolehlivé a eliminovaly neplatné značení zobrazené například na billboardu nebo značení z vedlejší silnice. Dostí tvrdým kritériem je i rychlost zpracování dat – při pomalém procesu je možné, že systém nezvládne číst kontinuálně síť dopravního značení.

DAS, které zpracovávají obrazové informace jsou obvykle ještě rozšířeny o informaci z tachometru a při detekci značení s rychlostním limitem a jeho porušení jsou schopny dát řidiči patřičně najevo – zvukovým signálem, obrazovým výstupem apod.

Předpokládá se, že by se v budoucnu mohly automobily stát více autonomními, tak jak známe z vědecko-fantastické literatury.

Upozornění

V práci je na několika místech uvedena následující tabulka, která symbolizuje ikonu programu MATLAB a čtenáře tak upozorňuje, které funkce přiložené na datovém nosiči se týkají právě projednávaného tématu.



- `test ()`
 - `help ()`
 - `example ()`
-

2. DOPRAVNÍ ZNAČENÍ

Z historického hlediska dříve plnily funkci dopravního značení pouze vyšlapané stezky a vyjeté kolejí od vozů či koní. V dobách antiky se objevily první náznaky značení – patníky. Ty vymezovaly prostor pro jízdu a pro chůzi. Objevil se též první přechod pro chodce v podobě zvýšeného chodníku, tak jak jej známe dnes. Později Římané začali své cesty značit návěštími se vzdálenostmi. První moderní dopravní signál, určující přednost na křižovatce, byl zaveden již v 18. století. Jednalo se o mechanický semafor se dvěma signály – volno a stůj. Hlavní linie vývoje dopravního značení byla až v rozmachu automobilismu na počátku 20. století.

První dopravní značení v ČR je dokumentováno z roku 1935 a obsahovalo pět druhů výstražných značek. O pár let později bylo rozšířeno o celou řadu dalších. Jelikož v té době, ještě nebyla známa reflexní folie (využívaná ve značení dnes) umísťovaly se na značky odrazka, aby byla zajištěna dostatečná viditelnost i v noci.

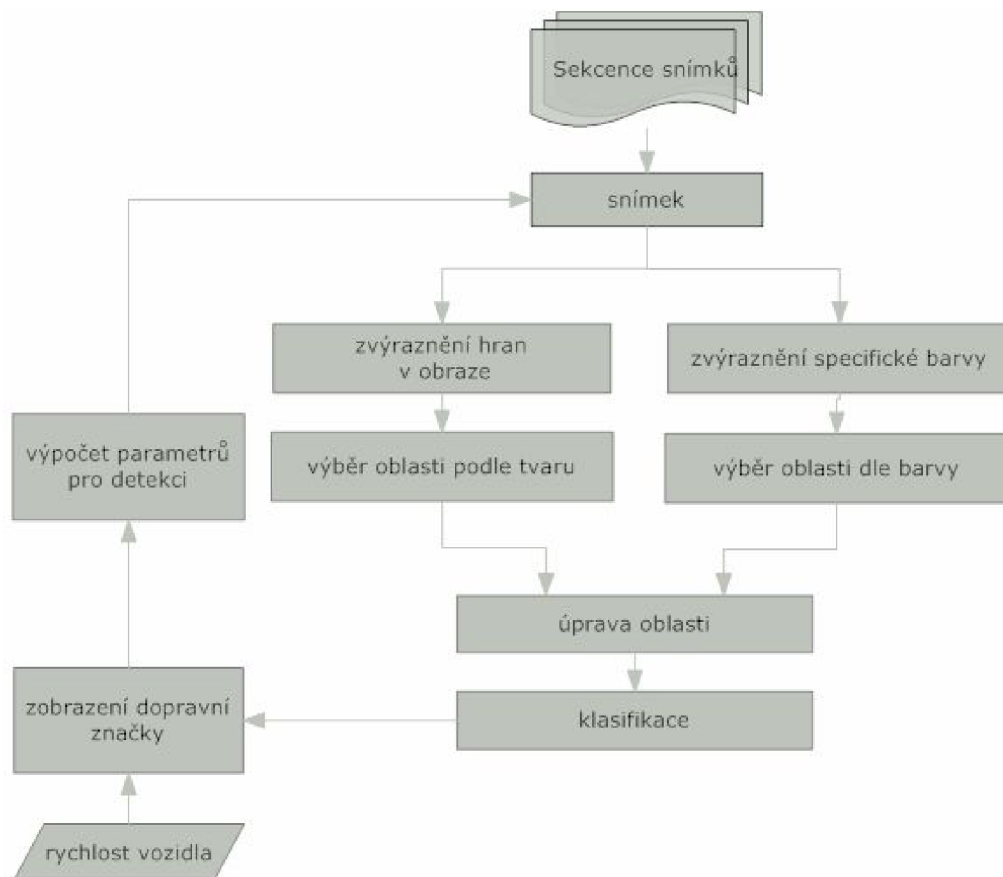
Dopravní značky jsou jednoduché piktogramy určené pro řízení a regulaci silničního provozu na pozemních komunikacích. Jedná se o zařízení upozorňující účastníky silničního provozu na nebezpečná místa, ukládají jim zákazy, příkazy nebo omezení, poskytují jim informace nebo zpřesňují, doplňují nebo omezují význam jiné dopravní značky. Význam dopravních značek zpravidla stanoví Pravidla silničního provozu [11]. Kromě dopravních značek je provoz na pozemních komunikacích také řízen, zabezpečován a usměrňován ještě dalšími dopravními zařízeními a dále světelnými a akustickými signály. [10]

Celkem je ve vyhlášce uvedeno více jak 200 druhů svislého dopravního značení rozdělených 11 skupin. Mezi nejdůležitější patří: výstražné značení, značky upravující přednosti, příkazové, zákazové a informativní dopravní značky. Kompletní grafický přehled je umístěn v příloze této práce. Více informací o dopravním značení lze nalézt přímo ve vyhlášce, která je umístěna na webu Ministerstva dopravy [11],[22].

2.1 DETEKCE DOPRAVNÍCH ZNAČEK

Detekce dopravních značek se skládá ze čtyř základních bodů: 1) segmentace, 2) klasifikace, 3) úprava oblasti, 4) identifikace. Segmentace obvykle zahrnuje použití určitého přístupu k obrazové informaci a vyhledávání podezřelých oblastí, které systém vyhodnocuje jako oblasti potenciálního výskytu dopravní značky. Tento přístup může být na bázi barevného prostoru s vymezením určitých barevných oblastí nebo použitím vyhledávání hran v obraze. Pixely v podezřelých oblastech jsou analyzovány v klasifikační a identifikační části, kde dochází k rozhodnutí, zda se jedná o dopravní značku či omyl.

Obrázek 2-1 je typickým schématem postupu při zpracování obrazových dat, který byl popsán výše.



Obrázek 2-1: Blokové schéma principu detekce značek – návrh řešení

Segmentace obrazových dat může být na bázi:

1. **segmentace obrazových dat na základě barevných informací** - značky mají zákonem definovaná barevná schémata, tak k dochází k výběru pouze oblastí s barvou odpovídající určité skupině značek – obvykle červená a modrá a žlutá.
2. **metoda založená na detekci gradientu obrýsu** – Metody založené na gradientu obrýsu (změně hrany) jsou více robustní s ohledem na světelné podmínky. Musíme však rozlišovat mezi metodami, které mají silné požadavky na členitost, tj. nalezení oblastí a spojení v jednu a na metody, které jsou řízeny modelem, tj. porovnávání se vzorem. Ty jsou bohužel více náročné na výpočetní výkon.
3. **kombinace obou předchozích** – dojde k barevné segmentaci a následnému zpracování těchto oblastí a například Houghovou transformací [14] se vyloučí nepravidelné objekty případně pomocí rohového operátoru [8].

Obvykle se využívá předpokladu, že dopravní značky jsou souměrné geometrické tvary, jejichž popis by měl být jednoduchý. Podezřelé oblasti ze segmentace jsou popsány určitými parametry – délka majoritní a minoritní osy, podobnost s kruhem, pravoúhlost, velikost natočení apod.

Následuje normalizování nalezené oblasti, která byla označena za potenciaální značku, vzhledem k měřítku a natočení a perspektivě. Obvykle také dochází k úpravě jasové složky v histogramu.

V části identifikace jsou tyto oblasti porovnávány s databází dopravních značek, které systém umí vyhodnotit a výsledná shoda je poté vhodně reprezentována řidiči - např. ve formě grafického zobrazení na displeji, případně doprovázené zvukovým signálem apod.

Jednoduché algoritmy zahrnují velmi málo informací o skutečném světě a spoléhají se výhradně na obrazové informace, které jsou bohužel ve dvourozměrném měřítku, odhadují pohyb jednotlivých objektů a stanovují korespondenci mezi snímky. Důmyslnější techniky rozpoznávání využívají navíc informaci o rychlosti vozidla k tomu, aby se lépe odhadlo, kde se daný objekt bude v budoucnu vyskytovat. Například vozidlo jedoucí konstantní rychlostí po rovné vozovce by mělo využívat k predikci Kalmanův filtr [12], [13] aplikovaný na jednotlivé snímky k tomu, aby sledoval středy nalezených dopravních značek.

3. SEGMENTACE OBLASTÍ

Metody založené na detekci pomocí barevné informace mají obvykle za účelem nalézt pouze barvy, které jsou typické pro dopravní značky - tedy červenou, modrou a žlutou. V některých systémech je ohraničení barevného prostoru přesně vybráno programátorem. Jiné metody využívají přibližného porovnávání barevné informace. V literatuře je zmíněna metoda J. Ansena [4], který používá jako základ segmentace vyhledávací tabulku, která byla vygenerována pomocí trénovací množiny několika set snímků dopravních značek. Snímky byly pořízeny za různých světelných podmínek s přihlédnutím k počasí (sníh, déšť, den, noc ..) a ručně na nich označeny oblasti odpovídající barevné dopravní značce. Tím dosáhl velké škály reálných barev a lepších výsledků při segmentaci. Po klasifikaci jsou jednotlivé pixely v mnoha barevných třídách a následuje fáze, kdy jsou vybrány pouze oblasti jejichž barva a tvar odpovídá dopravní značce v daném vztahu.

Barevná informace bohužel samostatně neposkytuje dostatečné ohraničení oblasti. Praktickým měřením barevných hodnot je zřejmé, že jejich barevnost velmi závisí na světelných podmínkách. Často tedy bývá ohraničení barevného prostoru velmi problematické i bez ohledu na jeho vyjádření - RGB, HSV či YCbCr.

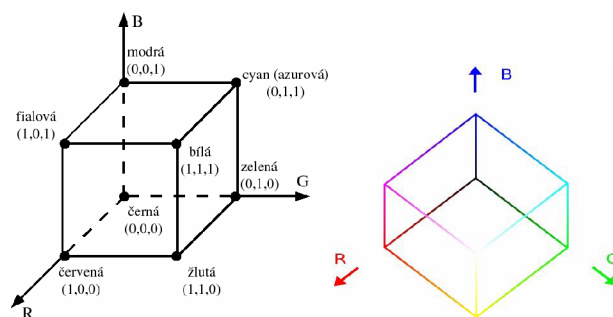
Navrhovaný systém by měl být schopný rozlišit tři základní barevné druhy značek – červené, žluté a modré. Pokud v obraze vybereme pouze oblasti, které svojí barevností odpovídají požadované skupině barev vzniknou nám celkem tři binární mapy (masky), které je potřeba dále předzpracovat binárními operacemi – eroze a dilatace, případně vyplnění děr a odstranění malých objektů jejichž plocha je menší než určité procento celkové plochy. Po těchto úpravách jsou již plochy připraveny k dalšímu kroku rozpoznání tvaru.

3.1 BAREVNÉ MODELY [3]

3.1.1 RGB

RGB (red, green, blue) - je aditivní barevný model, ve kterém je smícháno společně červené, zelené a modré světlo v různém poměru a vytváří tak celou škálu barevných odstínů (Obrázek 3-1), Každá barva je definována množstvím tří základních barev. Základní barvy mají vlnové délky 630, 530 a 450 nm. Mohutnost se udává buď v procentech (dekadický způsob) nebo podle použité barevné hloubky jako určitý počet bitů vyhrazených pro barevnou komponentu (pro 8 bitů na komponentu je rozsah hodnot 0 – 255, pro 16 bitů na komponentu je rozsah hodnot 0 – 65535), přičemž čím větší je mohutnost, tím s vyšší intenzitou se barva komponenty zobrazuje.

Model RGB je možné zobrazit jako krychli (Obrázek 3-1), ve které každá z kolmých hran udává škálu mohutností barevných složek. Potom libovolný bod se souřadnicemi (R-G-B) v této krychli udává hodnotu výsledné barvy [1],[2].



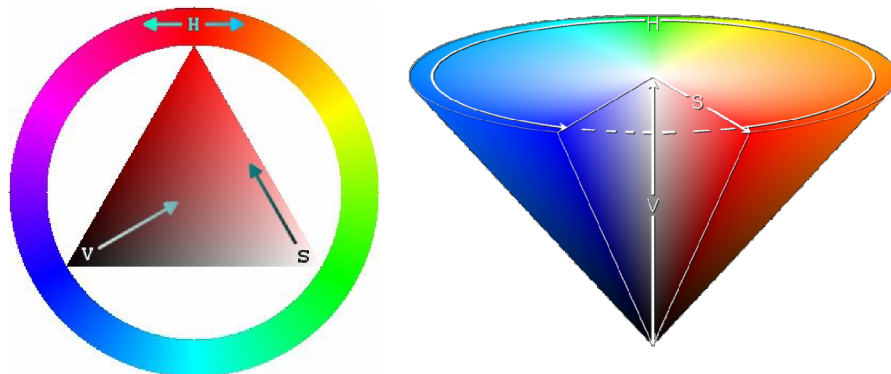
Obrázek 3-1: RGB barevný model [3]

3.1.2 HSV

HSV je známý také jako HSB (Hue, Saturation, Brightness), Tento barevný model nejvíce odpovídá vnímání barev člověkem. Skládá se ze tří složek, které jsou od RGB rozdílné. Je nutné hlídat jejich hodnoty, aby nedošlo k nesmyslným kombinacím.

- **Hue** – převládající barevný tón - odstín - barva odražená nebo procházející objektem. Měří se jako poloha na standardním barevném kole (0° až 360°). Obecně se odstín označuje názvem barvy.
- **Saturation** - sytost barvy, příměs jiné barvy. Představuje množství šedi v poměru k odstínu, měří se v procentech od 0% (šedá) do 100% (plně sytá barva). Na barevném kole vzrůstá sytost od středu k okrajům.
- **Value** - hodnota jasů, množství bílého světla. Jas vyjadřuje kolik světla barva odráží.

Tento barevný model bývá nejčastěji zobrazen v podobě jehlanu nebo kruhu s rotujícím trojúhelníkem (Obrázek 3-2).



Obrázek 3-2: HSV barevný model [3]

3.1.2.1 Přepočítání z RGB do HSV

Složky RGB odpovídají hodnotám červené, zelené a modré v hodnotách 0 až 1. Jsou přepočítány na hodnoty v HSV prostoru, kde $h \in \langle 0^\circ, 360^\circ \rangle$ je úhel odstínu v mírách, $l \in \langle 0, 1 \rangle$ je světlost a s saturace dle rovnice (1).

$$h = \begin{cases} \text{nedefinováno} & \text{jestliže } \max = \min \\ 60^\circ \cdot \frac{g-b}{\max-\min} + 0^\circ & \text{jestliže } \max = r, g \geq b \\ 60^\circ \cdot \frac{g-b}{\max-\min} + 360^\circ & \text{jestliže } \max = r, g < b \\ 60^\circ \cdot \frac{b-r}{\max-\min} + 120^\circ & \text{jestliže } \max = g \\ 60^\circ \cdot \frac{r-g}{\max-\min} + 240^\circ & \text{jestliže } \max = b \end{cases} \quad (1)$$

$$s = \begin{cases} 0, & \text{jestliže } \max = 0 \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max}, & \text{jinak} \end{cases}$$

$$v = \max$$

3.1.3 YCbCr

YCbCr je barevný model vyžívaný především u digitálních fotografií nebo videa. Složka Y vyjadřuje hodnotu světlosti (jasu) a složky Cb a Cr hodnoty červené a modré barvy. YCbCr není absolutní barevný model, ale je to způsob kódování RGB informací. Přímé zobrazení barev závisí na aktuálním užití RGB barev v signálu. Důvodem takového kódování je zejména použití ve video technice, kde složka Y obsahuje nejvíce informací a zaručí tak vysoké rozlišení signálu, a dvě chromatické komponenty – ty mohou být redukovány (komprimované).

Protože rovnice uvedené v [3], které definují YCbCr jsou formulovány ve tvaru, který natáčí celou nominální RGB barevnou kostku a mění její měřítko tak, aby se vešla do větší YCbCr barevné kostky, existují uvnitř krychle YCbCr barevné body, které nemohou být v RGB zobrazeny. To způsobuje určité potíže ve stanovené jak správně interpretovat některé YCbCr signály.

3.1.4 Převod RGB na YCbCr

Jak je výše uvedeno je zpětný převod z YCbCr velmi nepřesný, nám však pro práci při detekci dopravního značení postačí jednosměrný převod, který se provádí dle vzorců uvedených v rovnici (2).

$$\begin{aligned}
 Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\
 Cb &= 128 - 0.1687 \cdot R - 0.3312 \cdot G + 0.5000 \cdot B \\
 Cr &= 128 + 0.5000 \cdot R - 0.4186 \cdot G - 0.0813 \cdot B
 \end{aligned}
 \tag{2}$$

3.2 REPREZENTANTI DOPRAVNÍCH ZNAČEK

V průběhu práce na Semestrálním projektu 1 a 2 bylo pořízeno digitálním fotoaparátlem CANON S850 (Obrázek 9-1) několik stovek fotografií dopravního značení v ulicích města Brna. Cílem je, aby systém spolehlivě pracoval v městském i mimoměstském prostředí a detekoval veškeré dopravní značení platné na území České republiky. Nasnímané fotografie byly posléze roztrženy do několika kategorií dle tvaru a barvy dopravní značky, kterou zachycují. V sadě je i několik desítek snímků s více značkami. Celková množina čítá na 350 snímků. V každém z nich bylo ručně vybráno cca 15 bodů, které reprezentují barevnost dopravní značky. Každému takovému bodu bylo vzato i jeho nejbližší okolí o velikosti 5x5 px a tím se vytvořila velmi početná počáteční populace reprezentant vhodná ke stanovení patřičných závěrů. Inspirace dle [4]. Podrobný přehled uveden v tabulka 1.

Druh	Počet snímků	Označeno bodů	Celkem bodů
Kruhové červené značky	180	2600	65000
Trojúhelníkové červené značky	47	550	14000
Modré kulaté značky	16	420	10500
Modré čtvercové značky	82	1600	40250
Žluté hlavní značky	18	340	8500
Celkem	343	5510	138250

tabulka 1: Přehled barevných reprezentantů dopravního značení

Jak je z tabulky patrné, bylo označeno přes 5500 bodů, které včetně svého okolí vytvořily téměř 140000 barevných reprezentantů červených, modrých a žlutých značek. Pro další využití byly polohy označených bodů uloženy, stejně tak i hodnoty jednotlivých barev v těchto bodech. Posléze byla stanovena analýza převodu těchto RGB bodů na další barevné prostory, aby bylo možné stanovit závěry.

3.3 UKÁZKA SEGMENTACE



- `segmentace_RGB ()`
- `segmentace_YCBCR ()`
- `segmentace_HSV ()`
- `uprava_segmentace`

3.3.1 Segmentace v RGB

Jak již bylo řečeno výše je detekce v RGB prostoru méně používaná pro svoji netolerantnost ke světelným podmínkám. V [4], [5], [19] se vyskytuje několik možností jak této detekce dosáhnout. Často se tak používá rovnice (3), která detekuje čistě červené, modré nebo žluté oblasti v rámci stanovených poměrů mezi jednotlivými barevnými složkami.

$$\begin{aligned}
 seg_r(x, y) &: \begin{cases} 1 & RG_R \leq \frac{R(x, y)}{G(x, y)} \ \& \ RB_R \leq \frac{R(x, y)}{B(x, y)} \ \& \ BG_R \leq \frac{B(x, y)}{G(x, y)} \\ 0 & \text{jinak} \end{cases} \\
 seg_g(x, y) &: \begin{cases} 1 & GR_Y \leq \frac{G(x, y)}{R(x, y)} \ \& \ GB_Y \leq \frac{G(x, y)}{B(x, y)} \ \& \ RB_Y \leq \frac{R(x, y)}{B(x, y)} \\ 0 & \text{jinak} \end{cases} \\
 seg_b(x, y) &: \begin{cases} 1 & BG_B \leq \frac{R(x, y)}{G(x, y)} \ \& \ BR_B \leq \frac{B(x, y)}{R(x, y)} \ \& \ GR_B \leq \frac{G(x, y)}{R(x, y)} \\ 0 & \text{jinak} \end{cases}
 \end{aligned} \quad (3)$$

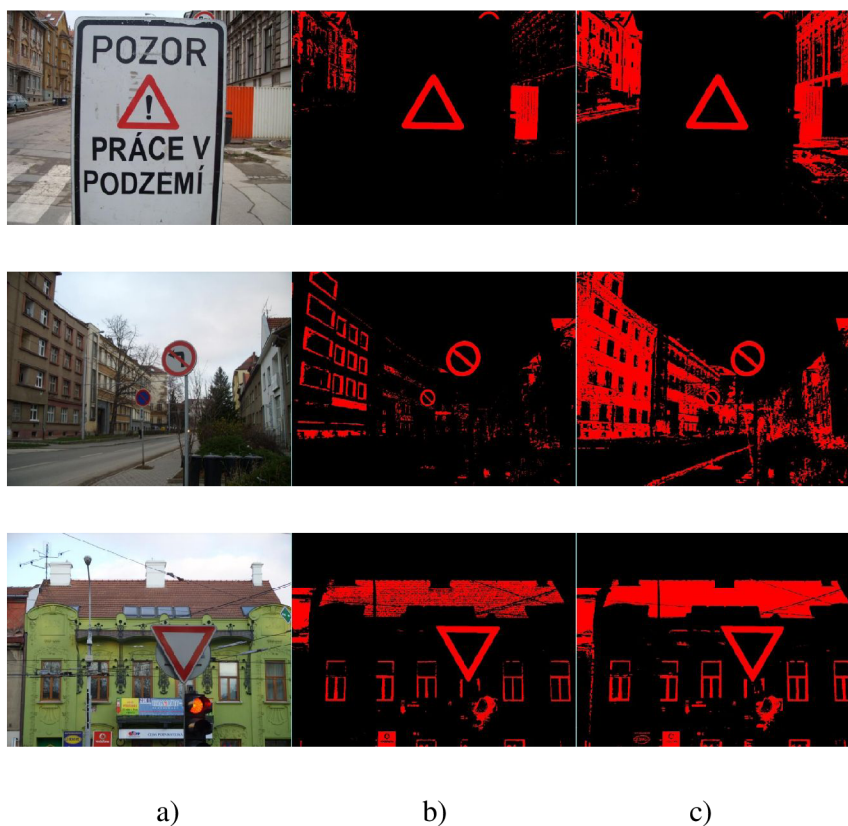
Přesto se v průběhu vývoje práce rovnice (3) ukázala jako nejlepší možná. Rovnice vychází z [19], kde bohužel nejsou uvedeny přesné hodnoty pro detekci barevných ploch a také vzorce pro detekci žluté či modré barvy. Díky využití vlastních barevných reprezentantů byly meze stanoveny tak, aby 99% reprezentantů bylo poměrem vyhovující vlastní barevné skupině.

Možná volba nastavení prahů je následující:

- Červená: $RG = 1.1, RB = 1.3, BG = 0.2$
- Žlutá: $GR = 0.9, GB = 1.8, RB = 2.8$
- Modrá: $BG = 1.2, BR = 1.35, GR = 1.34$

Důležité je upozornit, že uvedené hodnoty jsou v tuto chvíli platné pouze s přihlednutím ke snímacímu prvku, kterým byl digitální fotoaparát CANON S850. V případě použití jiného snímacího zařízení je nutné hodnoty vhodně pozměnit.

Jak je uvedeno na Obrázek 3-3, je segmentace červené barvy poměrně přesná, pokud je práh zvolen vhodně. Na snímku je ve sloupci c) uvedena záměrně hodnota prahu nižší než doporučená. To mělo za následek, že se do výstupu dostala i nežádoucí nevýrazná červená barva zdiva apod. Pakliže bude práh ještě vyšší než doporučený, je možné, že systém bude eliminovat nepříliš výrazné barevné značky.

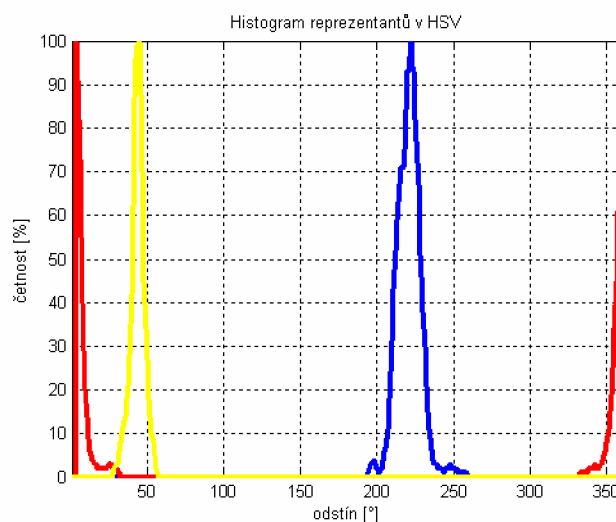


Obrázek 3-3: Ukázka segmentace červené barvy v RGB prostoru s proměnlivými prahy: a) originální obrázek, b) práh větší, c) práh menší

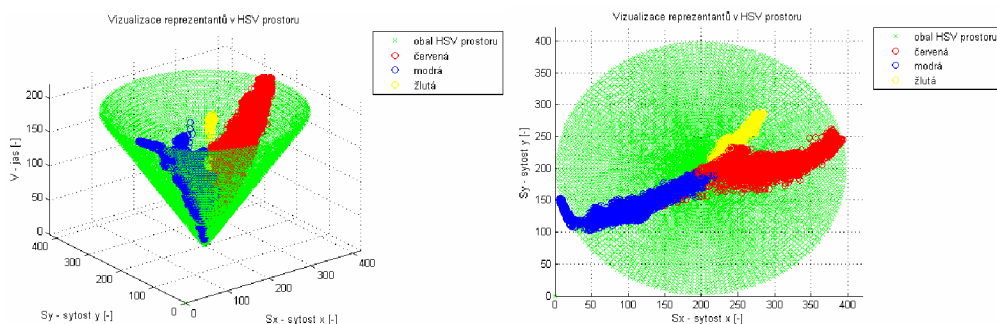
3.3.2 Segmentace v HSV

Reprezentanti barevnosti dopravních značek byly převedeni z RGB prostoru do HSV a vykresleny v podobě jehlanu, jak je uvedeno v kap. 3.1.2 této práce. Výsledek převod je zobrazen na snímku Obrázek 3-5 na němž je vidět rozložení jednotlivých odstínů. Popis os v obrázku neodpovídá skutečnému HSV prostoru jak na Obrázek 3-2. Je to dáno zobrazovacím systémem, který reprezentanty do jehlanu namodeloval (program MATLAB). Ke stanovení vhodných úhlů k separaci byl vykreslen HSV histogram pro složku H – odstín. Histogram vyjadřuje četnost výskytu daného odstínu v množině reprezentantů. Na obrázku je patrná separabilita těchto skupin a tak se úloha stává snáze řešitelnou. Pro segmentaci tedy použijeme podmínku, zda odstín leží v daných intervalech a jeho sytost je větší než minimální práh. Literatura [8] uvádí jako alternativu k HSV segmentaci využít Fuzzy logiky a na základě její odpovědi vyhodnotit, zda daný pixel odpovídá či neodpovídá barvě značky.

Přepočítání z RGB do HSV prostoru je bohužel časově náročnější jelikož využívá složitější operace, ale vzhledem k jeho výsledkům a viditelné separabilitě skupin, je toto prodloužení zanedbatelné.



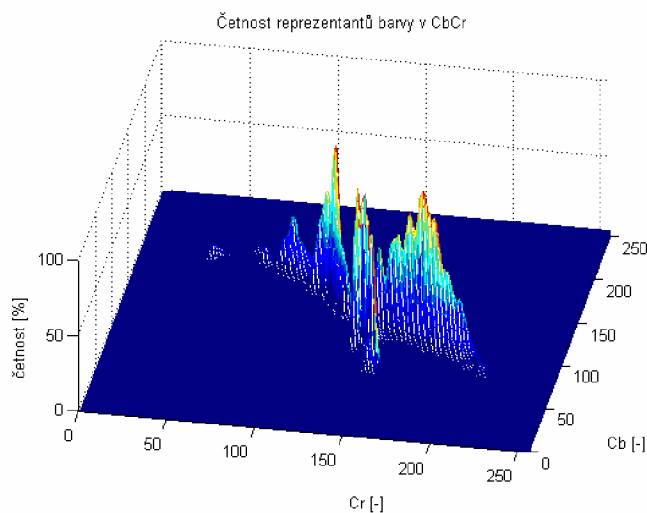
Obrázek 3-4: Reprezentanti dopravních značek – HSV, histogram



Obrázek 3-5: Reprezentanti barevných složek – HSV prostor, jehlan

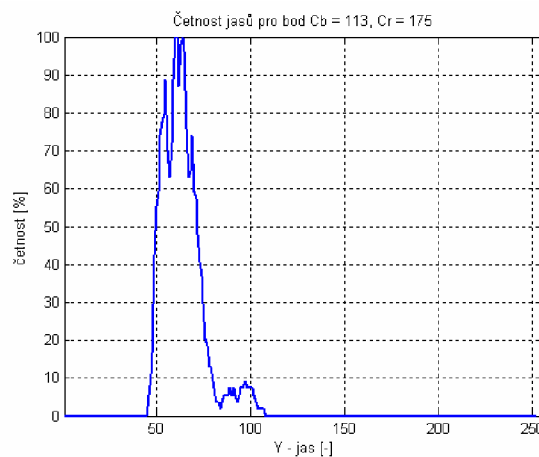
3.3.3 Segmentace v YCbCr

Převod do YCbCr prostoru již není tak výpočetně náročný jako převod z RGB do HSV. Veškeré reprezentanty červené, modré a žluté barvy jsou zobrazeni v CbCr ploše uvedené na Obrázek 3-6. Na ose z je vynesena jejich procentuální četnost. Pro eliminaci šumu byla provedena filtrace pomocí Gaussova filtru – tím se plochy reprezentantů více rozšířily do okolí a odstranily tak chybějící či přebývající informační šum.



Obrázek 3-6: Reprezentanti dopravních značek – četnost v CbCr prostoru, vlevo červené značky, vpravo modré značky, dole žluté značky.

Segmentace v tomto prostoru probíhá následovně. Celkem jsou vytvořeny pro každou barvu vlastní 3D mapy, kde výška odpovídá četnosti. Pakliže v obraze filtrujeme např. pouze žluté značky, převedeme celý snímek do YCbCr prostoru a pixel po pixelu prozkoumáváme, zda se nachází v této žluté mapě. Pokud ano a je jeho četnost větší než určitý práh (experimentálně stanovena hodnota 2%), považujeme jej za platný. Segmentace tímto způsobem je velmi přesná a systém tak pozná pouze barvy, které již zná díky reprezentantům. Přesto se však některé pixely do výstupu dostanou, aniž by byly platné. Může za to jasová složka, kterou jsme opomněli. Rozšířená segmentace, kterou používá navržený systém však pracuje i s touto složkou. Pro každého reprezentanta v prostoru CbCr byl vypočten maximální a minimální známý jas. Tímto rozšířením se segmentace stává ještě více robustnější. Na Obrázek 3-7 je uveden pro bod **Cb = 113 a Cr = 175** v červené mapě jeho průběh četnosti jasů. Z něj si lze snadno odečíst maximální a minimální známé jasy: **min = 45, max = 112**.

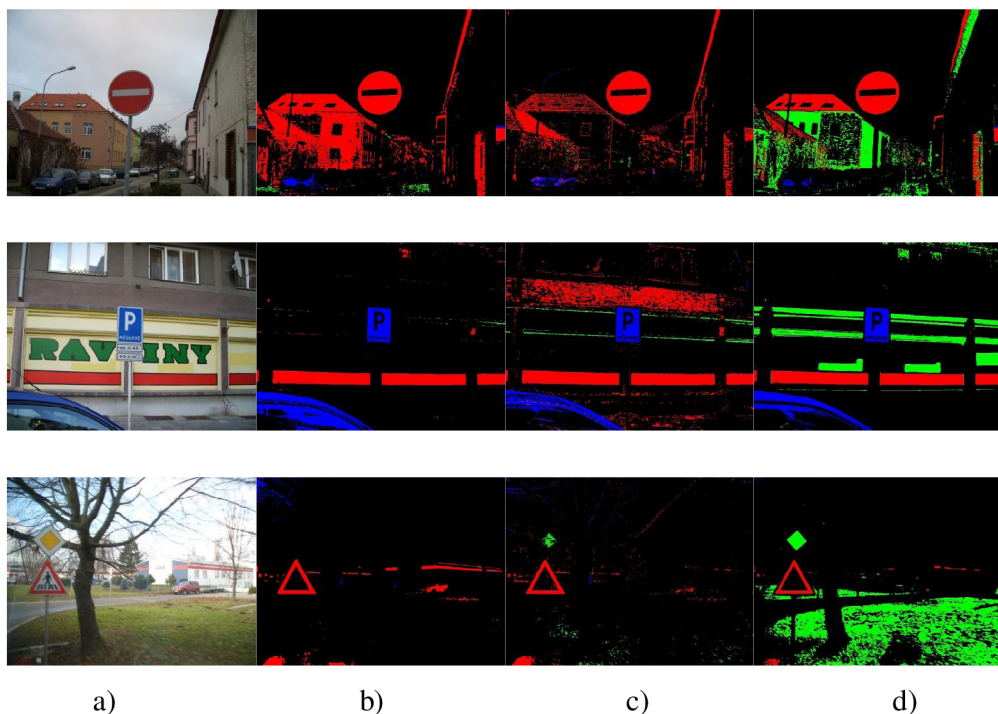


Obrázek 3-7: Segmentace YCbCr – průběh jasů reprezentanta

Pro segmentaci v YCbCr prostoru tak máme uloženo celkem 9 prostorových map. Pro každou barvu četnost v CbCr, maximum jasů v CbCr a minimum jasů v CbCr. Segmentace tedy probíhá dle výše uvedeného postupu, přičemž se nejprve pixel prozkoumává, zda leží v mapě četností, jestli je četnost této barvy větší než stanovený práh a zda je jeho jasová složka ve známých mezích.

3.4 SHRNU TÍ METOD, ZÁVĚRY

Výše uvedené metody segmentace mají své výhody i nevýhody. Například RGB segmentace ačkoliv je vlivem své jednoduchosti velmi rychlá je také bohužel nepřesná co se týče segmentace žluté barvy. Červená a modrá barva touto metodou byla však detekována velmi přesně. Při segmentaci metodou pomocí YCbCr map, bylo dosaženo uspokojivých výsledků. Bohužel metoda je velmi strohá co se týče snímacího prvku. Pakliže by bylo použito jiné snímací zařízení, musela by se celá barevná mapa modifikovat případně vytvořit zcela znova, což by bylo velmi časově a vývojově náročné. Z měření vyšla jako druhá nejrychlejší, avšak její přesnost nebyla uspokojivá. Ačkoliv segmentace v HSV prostoru je časově nejnáročnější dosáhlo se s ní během pokusů nejlepších výsledků. Velmi přesně totiž segmentovala žluté dopravní značení, se kterým předchozí metody měly problémy. Na Obrázek 3-8 je uveden přehled výsledků při detekci výše uvedenými metodami.



Obrázek 3-8: Segmentace barevných oblastí v RGB, YCbCr a HSV prostoru – a) originální snímek, b) RGB segmentace, c) YCbCr segmentace, d) HSV segmentace

Možnost zrychlení segmentace je použitím menších vstupních snímků či kombinace několika metod v závislosti na potřebě detekce určitých značek. Je tedy možné, pokud by měl uživatel nastaven systém pouze pro detekci červených značek tak využívat pouze RGB segmentaci.

Měření byla prováděna v programu MATLAB nad snímky o velikosti 640x480px. Hodnotila se míra detekce všech dopravních značek. Výsledky jsou uvedeny v tabulka 2. Jednotlivé algoritmy mají bohužel problémy s vybledlým značením, které je nedostatečně barevné. Dále pak se značením, které je nevýrazně kontrastní vzhledem ke svému pozadí (červená značka oproti červenému domu apod.). Tato nevýhoda se dá eliminovat pokud dopravní značky budou opatřeny bílým lemem jak je tomu již dnes u nově instalovaného značení.

	Segmentační čas [s]	Úspěšnost
RGB	0.200	93%
YCBCR	0.351	82%
HSV	0.420	97%

tabulka 2: Porovnání segmentačních metod

4. KLASIFIKACE OBLASTÍ

Po předchozím kroku segmentace jsou získány tři binární mapy (masky), které svoji úrovní $\log.1$ určují zda pixel odpovídá místu s pravděpodobným výskytem dopravní značky. Byly vytvořeny tři mapy pro červenou, žlutou a modrou.

Klasifikace, tedy rozpoznání tvaru, je velmi důležitým krokem. V této fázi totiž dojde k odstranění nežádoucích oblastí, které svým tvarem neodpovídají požadavkům a zůstanou tak pouze platné tvary.

Existuje celá řada postupů jak toho dosáhnout – maximální korelace se šablonou, genetické algoritmy aproximující kružnice a elipsy, rozhodování ze znalosti příznaků, detekce pomocí Houghovy transformace, porovnání oblasti se vzorem apod.

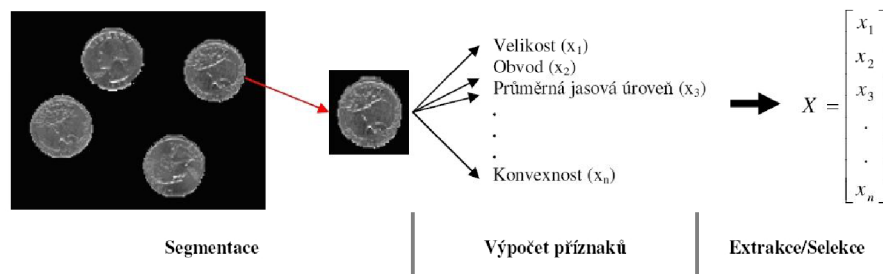
4.1 POPIS OBLASTÍ POMOCÍ PŘÍZNAKŮ

Obvykle je potřeba jednotlivé objekty popsat pomocí příznaků, které slouží pro klasifikaci objektů a proto musí přesně vystihnout charakteristické rysy. Na ně jsou současně kladeny často protichůdné požadavky mezi něž patří:

- **Invariantnost** – na změně jasu, kontrastu, na translaci, rotaci a změně měřítka
- **Spolehlivost** – podobné hodnoty příznaku pro týž objekt
- **Diskriminabilita** – různé hodnoty příznaků pro různé objekty
- **Efektivita výpočtu** – dobrá detekovatelnost
- **Časová invariance** – v případě zpracování dynamických obrazů

Hlavním problémem popisu segmentovaných dat je volba příznaků. Proti počítačovému vidění není rozpoznávání objektů lidským organismem postaveno na výpočtu a klasifikaci konkrétních hodnot, ale na využití naučených analogií rozsáhlou paralelní neuronovou sítí. Přes 80% všech přichozích informací je v mozku zpracováno pomocí vidění. Mnoho předmětů reálného světa je snadno rozpoznatelných lidským mozkem a přitom nejsou matematicky formálně definovatelné: lampa, ruka, strom apod.

Cílem popisu je reprezentovat obrazové objekty matematicky souborem číselných charakteristik nebo symbolů zvolené abecedy. Z obrazu segmentovaných objektů jsou vypočítány příznaky x_i , ze kterých je sestaven příznakový vektor X . Příznakový vektor může v obecném případě obsahovat vysoký počet zcela odlišných charakteristických rysů objektu (příznaku) bez ohledu na jejich konkrétní diskriminabilitu – rozlišovací schopnost.



Obrázek 4-1: Výpočet příznakového vektoru X [9]

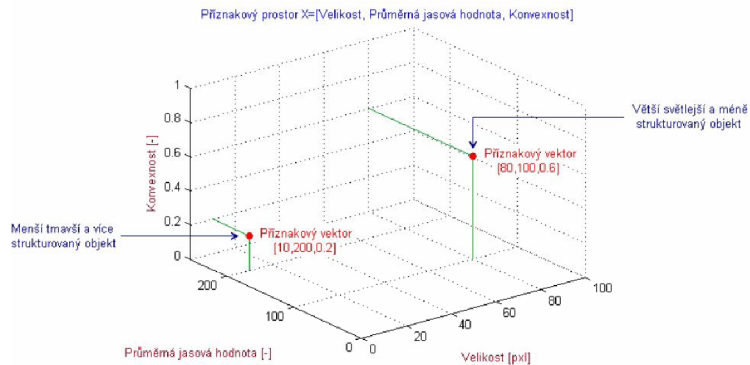
4.1.1 Extrakce, selekce příznaků

Vzhledem k tomu, že ze segmentovaného obrazu lze velmi často separovat nepřehledné množství příznaků, je nutné provést redukci dimenze příznakového prostoru. K tomu účelu lze využít dva základní přístupy.

1. **Extrakce příznaků** – transformace původního příznakového vektoru na vektor s nižším počtem členů. Nové příznaky mají odlišný význam od původních.
2. **Selekce příznaků** – výběr podmnožiny příznaků z původního příznakového vektoru, které vykazují nejvyšší diskriminabilitu. Výhodou je, že vybraným příznakům zůstává jejich původní význam.

Podle oblastí výpočtu lze rozeznávat deskriptory:

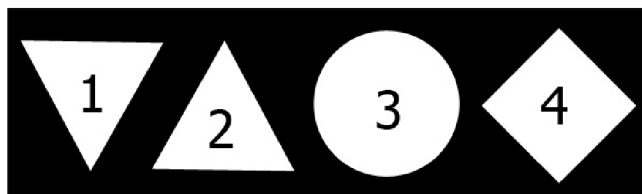
1. **založené na regionech** – nutná znalost jasových hodnot pixelů objektu
2. **založené na hranicích** – nutná znalost hranice objektu určená buď implicitně indexovaným seznamem pixelů objektu nebo explicitně funkčním předpisem.



Obrázek 4-2: Příklad příznakového prostoru [9]

4.1.2 Identifikace oblastí

Segmentací vzniká v nejjednodušším případě binární obraz. Tento obraz obsahuje oblasti vyznačující se spojitostí ve čtyř- nebo osmi- okolí, které je třeba indexovat – to znamená označit unikátním číslem. V programu MATLAB nám toto zajistí příkaz **bwlabel**. Počet značených oblastí je pak dán celkovým počtem použitých indexovacích symbolů. Výsledky z měření testovacích tvarů jsou uvedeny v elektronické příloze této práce.



Obrázek 4-3: Ukázka identifikace oblastí

4.1.3 Radiometrické deskriptory založené na regionech

Radiometrické deskriptory založené na regionech popisují vlastnosti objektu, které závisí pouze na rozměrech a prostorovém rozložení pixelů nikoliv na jejich jasu. Program MATLAB k tomuto určení nabízí funkci **regionprops**. V následujících statích nabízím přehled vlastností, které lze u objektu měřit a pro zpracování dopravních značek budou jistě nejzajímavější. Úplný přehled vlastností lze získat v [9].

4.1.3.1 Velikost (area)

Vyjadřuje počet pixelů spojitě oblasti a určuje tak plochu daného objektu. S rostoucím rozlišením samozřejmě počet pixelů roste.

4.1.3.2 Obvod (perimeter)

Počet ohraničených pixelů. Ten lze vypočítat jako počet pixelů ohraničujících objekt ve smyslu čtyř nebo osmiokolí. V případě osmiokolí se počítá s délkou $\sqrt{2}$ (což je délka diagonálního přechodu mezi pixely).

4.1.3.3 Nekompaktnost – obvod² / velikost

Vyjadřuje míru podobnosti oblasti s ideálním kruhem, který má obsah 4π . Čím je hodnota deskriptoru nižší, tím více se oblast podobá kruhu.

4.1.3.4 Konvexnost (solidity) = velikost / plocha konvexního obalu

Udává míru podobnosti objektu ke své konvexní schránce. Příznak může nabývat hodnot v intervalu $\langle 0;1 \rangle$, přičemž maximální hodnota je 1 platná pro konvexní (vypouklé objekty) a klesá k hodnotě 0 s rostoucí konkávností objektu.

4.1.3.5 Délka hlavní osy (MajorAxisLenght)

Délka hlavní osy elipsy v pixelech, jejíž centrální moment druhého řádu má stejnou hodnotu jako centrální moment druhého řádu segmentovaného objektu.

4.1.3.6 Délka vedlejší osy (MinorAxisLenght)

Délka vedlejší osy elipsy v pixelech, jejíž centrální moment druhého řádu má stejnou hodnotu jako centrální moment druhého řádu segmentovaného objektu.

4.1.3.7 Orientace (orientation)

Úhel hlavní osy objektu s některou souřadnicovou osou, zpravidla osou x. Vzhledem k tomu, že většina rozpoznávacích systémů směřuje k invariantnosti příznaků na rotaci, translaci a změně měřítka používá se jen ve specifických případech, kdy lze k natočení objektu usuzovat na nějaký charakteristický rys. Pixelech

4.1.3.8 Pravoúhlost (*extent*)

Maximální poměr velikost / plocha opsaného obdélníku. Postupným natáčením objektu se hledá minimální plocha jemu opsaného obdélníka. Čím je poměr plochy oblasti ke ploše obdélníka bližší hodnotě jedna, tím je vyšší jeho pravoúhlost.

4.2 KLASIFIKACE POMOCÍ PŘÍZNAKŮ

Jak již bylo zmíněno v úvodu této práce, dopravní značky jsou jednoduché geometrické tvary, které lze analyticky popsat jednak jejich rovnicí nebo právě pomocí příznaků. S využitím výše uvedených radiometrických vlastností, které lze u oblastí zjistit byl navržen vlastní rozhodovací strom, který dokáže u jejich hodnoty geometrický tvar určit. Před samostatným sestavením stromu, v kterém jsou aplikována rozhodovací pravidla závislá na radiometrických deskriptorech, byla potřeba vygenerovat počáteční množinu dat nutnou k měření a to následovně:

1. **obdélník** – 52 prvků
2. **čtverec** – 52 prvků
3. **elipsa** – 52 prvků
4. **kruh** – 52 prvků
5. **trojúhelník** – 205 prvků

Tyto tvary byly podrobeny základním afinním operacím, jako je změna měřítka, rotace a translace v obraze a poté vypočteny jejich radiometrické vlastnosti. Ukázka takovéto množiny (např. čtverců) je uvedena na Obrázek 4-4.

Z provedené analýzy byly stanoveny následující závěry.

1. Čtverec a kruh vykazovaly přibližně stejnou velikost majoritní a minoritní osy, naopak elipsa, obdélník velikosti rozdílné. Trojúhelník vykazoval poměr vždy různý – dle natočení.
2. Kruh vykazoval nekompaktnost blízkou hodnotě 15. Naopak čtverce tuto hodnotu měly okolo 23. Kruh navíc vykazoval hodnotu pravoúhlosti v rozmezí 0.7 – 0.8.

3. Všechny testované tvary měli vysokou konvexnost vyšší než 0.9.
4. Parametr orientace navrácí úhel natočení majoritní osy. Obdélník a elipsu tedy budeme rotovat o daný úhel a zajistíme tak, že pro následující kroky budou vždy ve stejné poloze.
5. Pokud byl obdélník pootočen zpět o záporně vzatý zjištěný úhel, lze předpokládat, že splňuje podmínku pravoúhlosti. Bylo změřeno, že obdélníky mají hodnotu pravoúhlosti větší než 0.9, elipsy okolo hodnoty 0.7 a trojúhelníky v rozmezí 0.4 až 0.6.
6. Měřením bylo zjištěno, že nejdůležitějšími parametry pro další klasifikaci jsou: poměr majoritní a minoritní osy, orientace, pravoúhlost, konvexnost.
- 7.



Obrázek 4-4: Ukázka trénovací množiny - ČTVEREC

Při této znalosti byl tedy realizován rozhodovací strom, který lze vidět na Obrázek 4-5. V něm uvedené číselné hodnoty jednotlivých stavů jsou pro lepší orientaci při programování a ladění programu, který zajišťuje rozhodování – klasifikaci tvaru.

Původně byl tento algoritmus realizován jako univerzální. Dovedl rozeznat o jaký tvar se jedná, ale vzhledem k optimalizaci zdrojového kódu pro rychlost byl jemně upraven. Je nutné si uvědomit hlavní myšlenku:

1. červené dopravní značky mohou být kruhové (elipsy) nebo trojúhelníky
2. modré dopravní značky mohou být kruhové (elipsy) nebo čtverce (obdélníky)
3. žluté dopravní značky mohou být pouze čtverce či obdélníky.

Pakliže je kruhová značka snímána z jiné roviny, než je rovina kolmá na rovinu značky, jeví se jako elipsa. Podobné je to i u čtvercových značek, které degradují k lichoběžníkovému tvaru

Do vývojového diagramu rozhodovacího stromu tak byly dodány informace, která větev na jaký finální tvar vede. Proto není tedy nutné, pro každý podezřelý objekt procházet celý vývojový strom a dojít k přesnému názvu tvaru, ale lze jej opustit v jakémkoliv místě, pokud se detekce odchýlí od hlavní linie.

Příklad: máme segmentovanou plochu z červené barvy a objekt tedy může být pouze kruh nebo trojúhelník. Jestliže se ve stavu č.7 zjistí, že pravoúhlost ukazuje na čtverec, prohlásíme objekt za neplatný, dále stromem nepokračujeme a tudíž není ztrácen zbytečný výpočetní čas k detekci Houghových přímek apod.

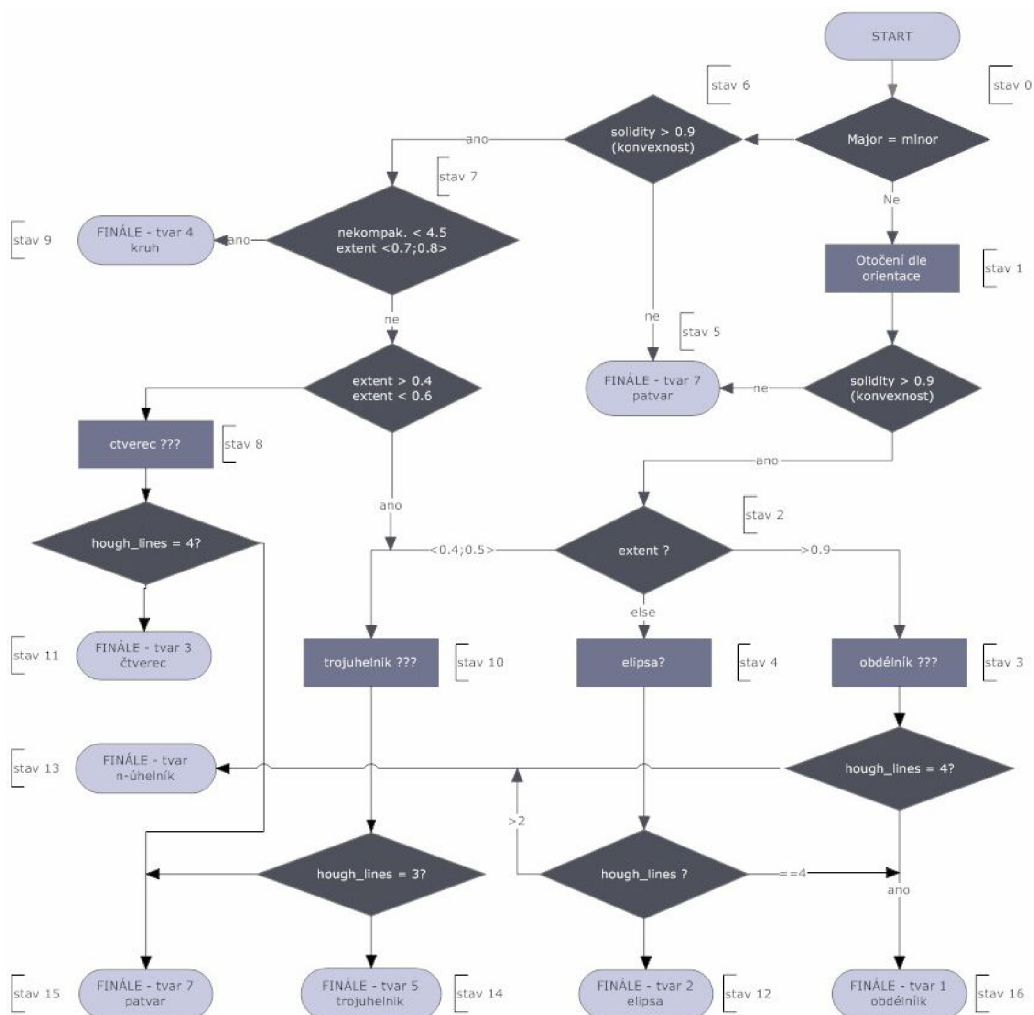
Pro detekci tvarů, které vykazují lineární části, tedy například čtverce, obdélníky či trojúhelníky, byla využita Houghova transformace [14], [15] pro přímky. Tyto přímky jsou dále prověřeny, zda jejich kombinace odpovídá nějakému geometrickému tvaru. Bylo využito znalosti úhlu natočení (φ) a vzdálenosti od počátku souřadného systému (r).

Čtverce a obdélníky mají tu vlastnost, že mají pouze dva různé úhly a ještě k tomu na sebe kolmé (případně úhel svírá více jak 80° budeme-li uvažovat optické zkreslení vzhledem k rovině snímače). Trojúhelníky mají vnitřní úhly 3, které v svírají v rovnostranném trojúhelníku úhel 60° (samozřejmě jen přibližně vzhledem ke zkreslení). S tímto poznatkem lze tedy tyto geometrické objekty odlišit.

Pro detekci elipsy a kruhu lze využít stejně tak i Houghovu transformaci pro kruhy či Modifikovanou Houghovu transformaci upravit pro detekci elips s tím, že se její stavový prostor rozšíří o několik dimenzí navíc [14], [15].



- `urci_tvary_strom()`
- `kontrola_cesty()`
- `houghtest()`
- `houghtvar()`



Obrázek 4-5: Klasifikace tvaru – vývojový diagram

Celý algoritmus je patrný z uvedeného obrázku 4-5. V první fázi dochází k porovnání velikostí majoritní a minoritní osy. Pakliže jsou si jejich hodnoty přibližně podobné algoritmus usuzuje, že se může jednat o symetrické tvary – čtverce, kruhy. Jestliže je poměr rozdílný lze usuzovat, že se jedná o elipsu či obdélník. Pokud je poměr os větší než stanovená mez je okamžitě zkoumaný objekt prohlášen za patvar. U trojúhelníku nelze vlivem nestálého poměru os přesně říci, která ze dvou hlavních větví je ta pravá. Proto se v dalších krocích rozhoduje o podobnosti trojúhelníku v obou větvích stromu.

Důležitým faktorem je příznak konvexnosti – ten odliší pravidelné tvary od nepřesně segmentovaných patvarů.

Vlastní navržený rozhodovací strom je velmi přesný, ale vzhledem k nutnosti výpočtů radiometrických deskriptorů není příliš rychlý a tak se stává pro real-time aplikace bohužel nevhodným.

4.3 KLASIFIKACE KRUHU POMOCÍ GENETICKÉHO ALGORITMU

Genetický algoritmus (GA) [17],[18] je heuristická postup, který se snaží aplikací principů evoluční teorie nalézt řešení složitých problémů, pro které neexistuje použitelný exaktní algoritmus. Genetické algoritmy, resp. všechny postupy patřící mezi tzv. evoluční algoritmy používají techniky napodobující evoluční procesy známé z biologie – dědičnost, mutace, přirozený výběr a křížení.

Princip práce genetického algoritmu je postupná tvorba generací různých řešení daného problému. Při řešení se uchovává populace jejíž jedinec představuje jedno řešení. Jak populace probíhá evolucí, řešení se zlepšuje. Tradičně je řešení reprezentováno binárními čísly, řetězci nebo i jinak. Typicky je na začátku simulace (v první generaci) vytvořena zcela náhodná populace. Při přechodu do další generace je pro každého jedince vypočtena fitness funkce, která určuje míru kvality daného jedince – shoda s výsledkem řešení. Podle této kvality jsou poté stochasticky vybráni jedinci a je provedena mutace a křížení. Kvalitnější jedinec je častěji křížen s ostatními, slabší jedinci se pokud možno eliminují. Tento postup se opakuje tak

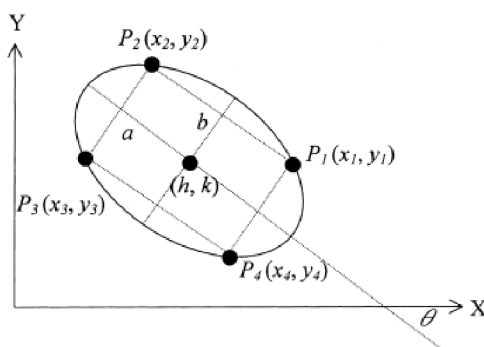
dlouho, dokud není nalezeno vhodné řešení – vynikající jedinec či po určitém počtu generací.

Pokud se tedy omezíme, pouze na detekci kruhových značek, můžeme použít následující genetický algoritmus tak jak jej zmínil Soedtedja [16] ve své práci. Na barevném obraze je provedena segmentace červené barvy RGB metodou. Ve vzniklém binárním snímku dojde k detekci kruhových objektů podle masky [8]. Masku je rotována o 90° a tak jsou nalezeny všechny strany kruhu. Masku filtru musí být dostatečně robustní, tak aby eliminovala přímé linie a jiné nevhodné tvary. Další metodou je využití Cannyho detektoru k nalezení kontury objektu [1],[2] či nalezení hranice objektu s využitím vhodné kombinace binárních operací.

Každá elipsa je určena následujícími parametry: h, k (poloha středu), a, b (délky majoritní a minoritní osy) a φ (natočení), kde $0^\circ \leq \varphi \leq 360^\circ$. Jednotlivé body na elipse mohou být vyčísleny podle (5). [16].

$$(x, y) = (a \cdot \cos \gamma \cdot \cos \varphi + b \cdot \sin \gamma \cdot \sin \varphi + h, -a \cdot \cos \gamma \cdot \sin \varphi + b \cdot \sin \gamma \cdot \cos \varphi + k) \quad (4)$$

Výpočet bodů elipsy přes celou škálu úhlů $0^\circ \leq \gamma \leq 360^\circ$ může být výpočetně náročný, proto se v mnoha aplikacích využívá její symetrie podle středu jak je ukázáno v tabulka 3.



V první části je vyčíslena hodnota (x_γ, y_γ) v úhlu $0^\circ \leq \gamma \leq 90^\circ$ a 4 souměrné body se vypočtou následovně:

$$\begin{aligned} (x_\gamma, y_\gamma) &= (a \cdot \cos \gamma, b \cdot \sin \gamma) \\ (x_1, y_1) &= (x_\gamma \cdot \cos \varphi + y_\gamma \cdot \sin \varphi + h, -x_\gamma \cdot \sin \varphi + y_\gamma \cdot \cos \varphi + k) \\ (x_2, y_2) &= (-x_\gamma \cdot \cos \varphi + y_\gamma \cdot \sin \varphi + h, x_\gamma \cdot \sin \varphi + y_\gamma \cdot \cos \varphi + k) \\ (x_3, y_3) &= (-x_\gamma \cdot \cos \varphi - y_\gamma \cdot \sin \varphi + h, -x_\gamma \cdot \sin \varphi - y_\gamma \cdot \cos \varphi + k) \\ (x_4, y_4) &= (x_\gamma \cdot \cos \varphi - y_\gamma \cdot \sin \varphi + h, x_\gamma \cdot \sin \varphi - y_\gamma \cdot \cos \varphi + k) \end{aligned}$$

(5)

tabulka 3: Elipsa – využití symetrie podle středu [16]

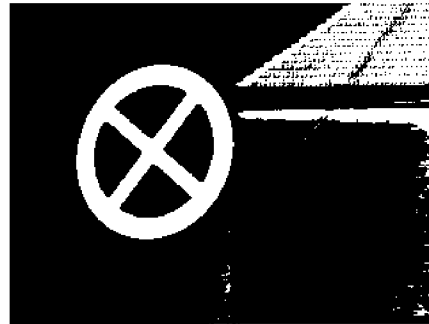


- `detekce_elipsy()`

Tento proces urychlí výpočet přinejmenším 4x. Další zrychlení přinese vygenerování vyhledávací tabulky pro funkce sin/cos pro různé úhly v rozmezí $0^\circ - 90^\circ$. Počáteční generace byla zvolena o N jedincích, přičemž každý jedinec měl uloženo celkem 5 parametrů elipsy. Jako fitness funkci byla zvolena metoda průniku nalezené elipsy vypočtené dle rovnic (5) se vstupní elipsou v obraze. Po M generacích, kdy je průběh obou elips téměř totožný je získáno optimální nastavení parametrů elipsy.



a)



b)



c)



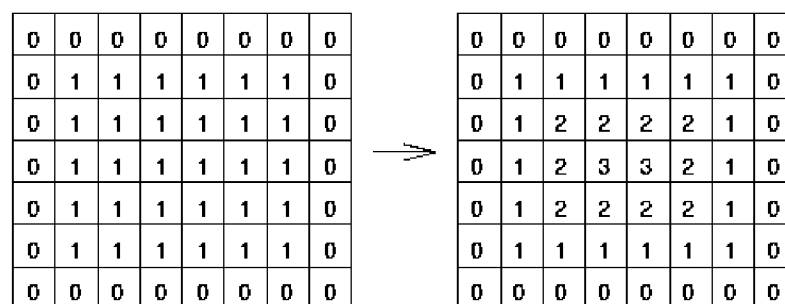
d)

Obrázek 4-6: Genetický algoritmus pro detekci kruhu: a) originální snímek, b) segmentace červené barvy, c) po aplikaci kruhového filtru, d) detekovaná elipsa

Jako nevýhodu GA je nutné uvést zejména jejich náročnost na výpočetní výkon úměrný velikosti populace a počtu generací. Navíc, jak je patrné z Obrázek 4-6 výsledek není zrovna nejpřesnější. Další modifikace algoritmu je možná pokud bude stanovena vhodnější počáteční populace vzhledem k parametrům vstupního obrazu apod.

4.4 VZDÁLENOSTNÍ TRANSFORMACE - DISTANCE TRANSFORM (DT)

Vzdálenostní transformace se běžně používá pro binární obrazy. Výsledkem této transformace je šedotónový obrázek, který vypadá podobně jako vstupní snímek, s výjimkou toho, že intenzity uvnitř výstupu odpovídají vzdálenosti k nejbližší hranici z daného bodu. Jedním ze způsobů, jak si DT představit je ten, že se podobá hoření papíru. Na okrajích oblasti necháme jakoby vzplanout oheň a sledujeme jeho šíření směrem do středu oblasti a každému pixelu vně určíme jeho čas vzplanutí, což ve skutečnosti znamená počet pixelů od okraje oblasti. V každý časový úsek totiž vytvoříme nové hranice objektu a plochu co je uvnitř označíme vždy indexem větším než byl předchozí okraj a tak stále pokračujeme až do středu oblasti. Na snímku Obrázek 4-10 je pak zobrazena ukázka jak vypadá výstupní snímek DT.



Obrázek 4-7:: Vzdálenostní transformace – maticová ukázka [5]

Existuje i druhý přístup k DT, který produkuje DT pro pozadí mimo oblasti. Je tedy zřejmé, že jde o invertování vstupního snímku a poté uplatňování výše uvedeného algoritmu. Dále je zde několik různých druhů vzdálenostní transformace

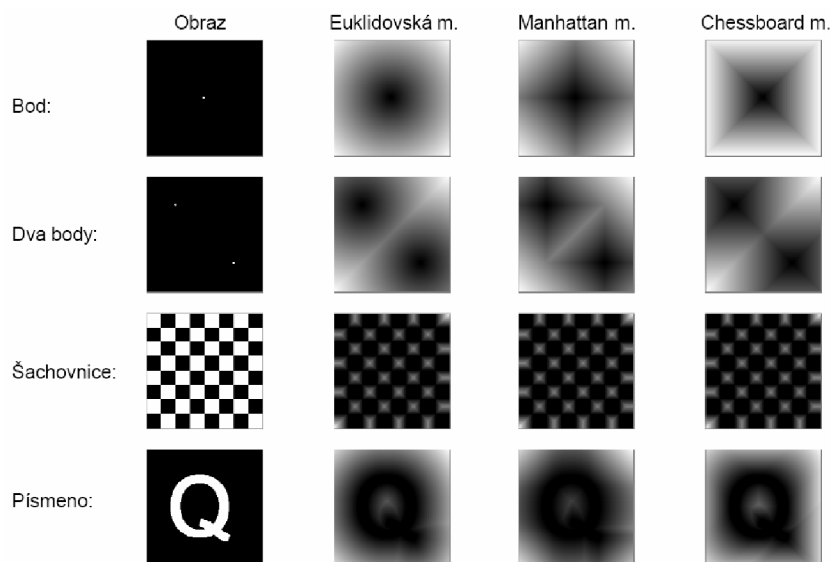
v závislosti na tom, jaké metrické měřítko bude zvoleno pro určení vzdálenosti mezi pixely (tabulka 4).

<p><u>Euklidovská vzdálenost</u></p> $d(P, Q) = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2}$	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table> <p>Image</p> </div> <div style="text-align: center;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td>1.41</td><td>1.0</td><td>1.41</td></tr> <tr><td>1.0</td><td>0.0</td><td>1.0</td></tr> <tr><td>1.41</td><td>1.0</td><td>1.41</td></tr> </table> <p>Distance transform</p> </div> </div>	0	0	0	0	1	0	0	0	0	1.41	1.0	1.41	1.0	0.0	1.0	1.41	1.0	1.41
0	0	0																	
0	1	0																	
0	0	0																	
1.41	1.0	1.41																	
1.0	0.0	1.0																	
1.41	1.0	1.41																	
<p><u>Městské bloky</u></p> $d(P, Q) = p_1 - q_1 + \dots + p_n - q_n $	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table> <p>Image</p> </div> <div style="text-align: center;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td>2</td><td>1</td><td>2</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>2</td></tr> </table> <p>Distance transform</p> </div> </div>	0	0	0	0	1	0	0	0	0	2	1	2	1	0	1	2	1	2
0	0	0																	
0	1	0																	
0	0	0																	
2	1	2																	
1	0	1																	
2	1	2																	
<p><u>Šachovnicová vzdálenost</u></p> $d(P, Q) = \max(p_1 - q_1 + \dots + p_n - q_n)$	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table> <p>Image</p> </div> <div style="text-align: center;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p>Distance transform</p> </div> </div>	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0																	
0	1	0																	
0	0	0																	
1	1	1																	
1	1	1																	
1	1	1																	

tabulka 4: Vzdálenostní transformace – přehled metrických vzdáleností [5]

Poté, co byla zvolena metrika, existuje řada způsobů výpočtu jak DT aplikovat na binární obraz. Jeden jednoduchý, ale neefektivní způsob výpočtu vzdálenosti je provedení několika po sobě jdoucích erozí s vhodným tvarem elementu do té doby, dokud se oblast zcela nevymaže. Je-li každý pixel označen počtem erozí, které musely být provedeny předtím, než zmizel, pak je to právě vzdálenost od okraje. Erozní prvek závisí na zvolené, která byla zvolena. Prvek ve tvaru 3x3 dává šachovnicovou vzdálenost, kříž udává Manhatanskou vzdálenost (městských bloků) a kolečko odpovídá euklidovské vzdálenosti. Samozřejmě, že není skutečně možné vytvořit dobrý tvar kolečka v diskretní mřížce v malém měřítku, ale existují algoritmy, které mění tvar kolečka během eroze a tak aproximují přibližně

kruhový tvar. DT je možné vypočítat i mnohem efektivněji dle algoritmů založených na metodě rekurzivní morfologie [4].

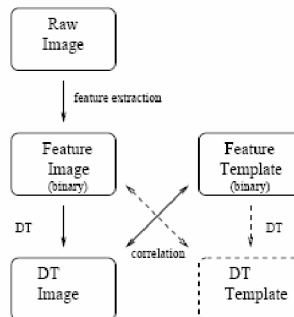


Obrázek 4-8: Vzdálenostní transformace – ukázka různých metrik [8]

4.4.1 Popis metody D. M. Gavřila

Metoda, kterou popisuje D. M. Gavřila ve svém dílu [4], na téma rozpoznávání dopravních značek, se právě zabývá vzdálenostní transformací, díky které rozpoznává základní tvary v reálném čase. Důkazem jeho metody je 90% úspěšnost při detekci značek zachycených na fotografiích, které byly pořízeny z paluby automobilu.

Detekční část využívá šablonový základ, který je založený na korelačních metodách a identifikuje potenciální dopravní značky v obraze přičemž využívá právě vzdálenostní transformaci (DT).



Obrázek 4-9: Metoda podle D. M. Gavrila [4]

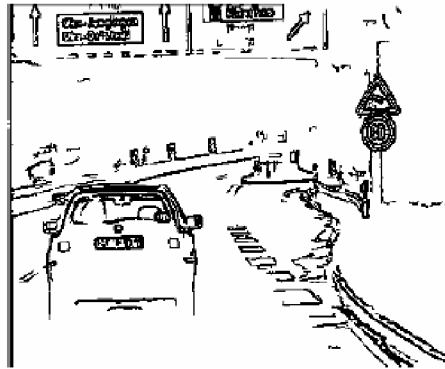
Pracuje na principu, kdy snímaná scéna je pomocí hranového filtru převedena na binární snímek a ten je segmentován. Dále obsahuje šablony známého tvaru dopravní značky. Oba dva snímky jsou podrobeny DT transformaci. Poté je šablona značky transformována (rotace, translace, měřítko) a umístěna přímo do výsledky DT vstupního obrazu I. Vyhledává se maximální korelace obou snímků.

Na Obrázek 4-10 je ukázka toho, jak DT funguje v praxi. Ve snímku DT jsou patrné velké vzdálenosti (zobrazeny světle) a malé (zobrazeny tmavě), které signalizují i možnou přítomnost dopravní značky, jak je z obrázku patrné.



a)

b)



c)



d)

Obrázek 4-10: D. M. Gavrilla - a) originální obrázek, b) šablona, c) binární snímek obrázku, d) DT transformace [4]

Navržený systém rozšiřuje základní DT systém ve dvou krocích. V prvním jsou hranové obrysy diferencovány dle jejich orientace. Odděleně je poté DT vypočtena pro každou orientaci v intervalu hranového filtru samostatně. Hranový filtr je rozdělen do několika částí s podobnou hranovou orientací. Spojování probíhá tak, že je souhlas spojení měření mezi obrazem a šablonou součtem měření mezi šablonou a DT obrazem korespondence stejně orientovaných hran. Přidáním informace o směru hrany přináší do detekce výrazné urychlení [4].

Po detekci a nalezení korespondujících oblastí jsou z originálního obrazu vybrány pouze vnitřní části dopravních značek. Následuje krok normování (pokud nebyl proveden již při kroku detekce), kdy jsou jednotlivé oblasti upraveny na stejné měřítko, natočení a velikost $N \times N$ pixelů, přičemž okolní oblasti jsou při dalším zpracování ignorovány

Experiment, který D. M. Gavrila provedl dokázal poznat základní dopravní značky – trojúhelník, kruh atd. Pro detekci použil šablony o velikostech 7-18px a snímané obrazy o velikosti 360x288px. V databázi měl přes 1000 snímků. Ty byly pořízeny z kapoty automobilu za různých světelných podmínek a počasí. Při pokusech dosáhl 95% úspěšnosti. V průměru byly detekovány 1-2 falešné oblasti na snímku. Přesto je nutné pamatovat, že existují náročnější podmínky počasí, které mohou detekci podstatně zhoršit – sněhová vločka před objektivem, dešťové kapky, nečistoty, přímé sluneční světlo apod. přičemž úspěšnost samozřejmě klesá.

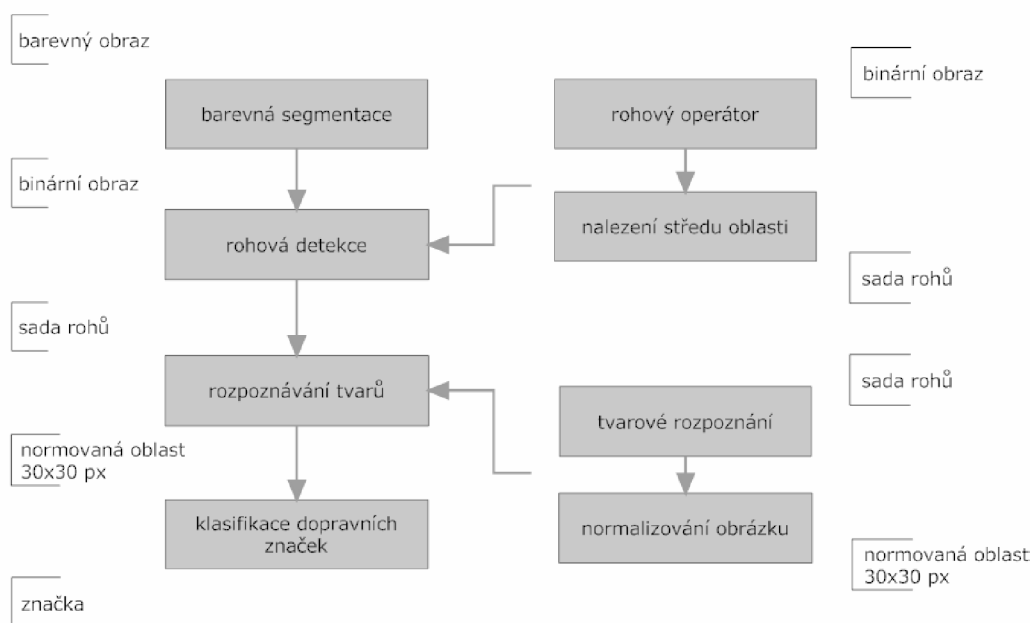


Obrázek 4-11: D.M.Gavrila – výsledky detekce [4]

4.5 POPIS METODY HASSAN SHOJANIA [8]

Hassan Shojania [8] použil ve své práci metodu, která je založena na barevné segmentaci a na následném zpracování rohů a významných bodů segmentovaného obrazu. Sestrojil řadu konvolučních jader (matic) k nalezení rohů v binárním obraze. Jeho metoda byla úspěšná pro detekci trojúhelníků, osmiúhelníků, kruhových červených značek a čtverců. Jakékoliv deformace a natočení již nebyla akceptována. Metoda spolehlivě určovala značky, které byly přímo namířeny na kamerový systém, případně s mírným sklonem k rovině snímání. Blokové schéma metody je vyobrazeno na Obrázek 4-12.

Dopravní značky jsou velmi diskriminabilní barvou a tvarem. Shojania metodu založil na detekci v barevném snímku, ve kterém segmentoval pouze oblasti se stejnou barvou – převážně červenou s využitím segmentace v RGB.



Obrázek 4-12: Hassan Shojania – metoda barevné segmentace a vyhledávání rohů [8]

Pro detekci rohů existuje nekonečné množství konvolučních jader (masek). Rohy mohou být pod libovolným úhlem a libovolné velikosti. Z celého množství lze stanovit celkem 12 masek, které aproximují celou škálu rohů a jsou nazývány jako rohové detektory. Tyto masky nejsou citlivé na šum, protože svým rozměrem zpracovávají velké okolí. Rohové detektory slouží též k detekci hran. Nevýhoda, které přináší tato metoda je ta, že pokud potřebujeme vyčíslit všechny rohy v obraze, musíme spočítat hodnotu pro všechny masky a pro každý pixel v obraze. Řešením tohoto nešvaru by mohla být aproximace několik podobných masek v jednu.

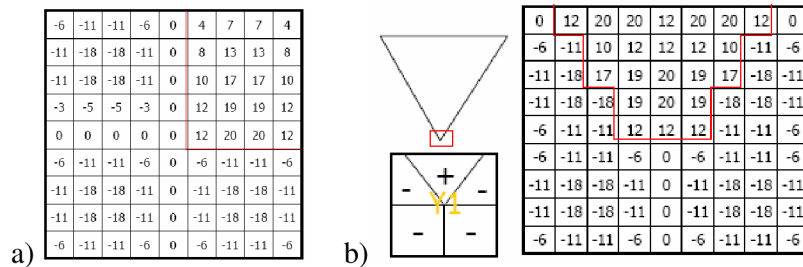
Algoritmus zpracování binárního obrazu je přibližně následující. Vypočítá se rohovitost v každém pixelu použitím rohové masky a následně se provede prahování. Dojde k vyřazení bodů, které mají méně jak dva sousedy v okolí s podobnou změnou

úhlu. Tímto nám v obraze zbude velmi malé množství rohů, se kterými lze nadále pracovat. Každý roh je přidán na seznam rohů.

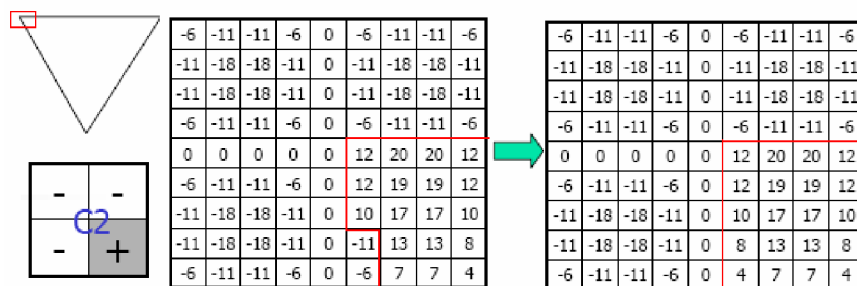
Metoda využívá celkem 5 masek o velikosti 9x9px (Obrázek 4-13):

- Y1 – maska pro nalezení spodního vrcholu trojúhelníka
- C1, C2, C3, C3 – 4 masky pro detekci 90° rohů

Tyto rohy jsou nadále prověřeny, zda jejich kombinace neodpovídá nějakému známému geometrickému objektu.



Obrázek 4-13: Hassan Shojanian – a) maska C1 pro detekci rohu, b) maska Y1 [8]



Obrázek 4-14: Hassan Shojanian – a) maska pro levý horní roh trojúhelníka, b) její aproximace [8]

4.5.1 Detekce trojúhelníku

Postup je patrný z Obrázek 4-16 a).

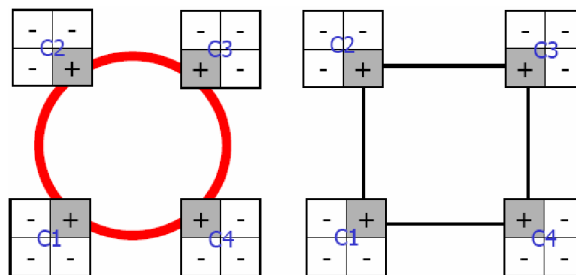
- nalezne se bod **p1** pomocí **Y1** rohového operátoru. Poté se hledá body **p2** v obraze tvořeným rohy **C2** a bod **p3** tvořeným rohem **C3**. Hledání probíhá pouze v úzkém pásmu úhlů, které odpovídají trojúhelníku.

- kontrola správnosti trojúhelníku je kontrolována existencí hrany pomocí operátoru C1 a C4.
- zkontroluje se, zda vnitřní část trojúhelníku je červená a ve výsledku dojde k ohraničení celé platné oblasti.

4.5.2 Detekce kruhu, čtverce

Postup je patrný z Obrázek 4-15.

- nalezneme bod **p1** pomocí **C1** a bod **p4** pomocí **C4**.
- následně se hledají body **p2** (maska **C2**) a bod **p4** (maska **C3**) opět v úzkém pásmu úhlové oblasti. Body jsou také hledány v určitém pásmu ve vzdálenosti h_{\min} až h_{\max} , určené úměrně poměrem vzdálenosti mezi **p1** a **p4**.

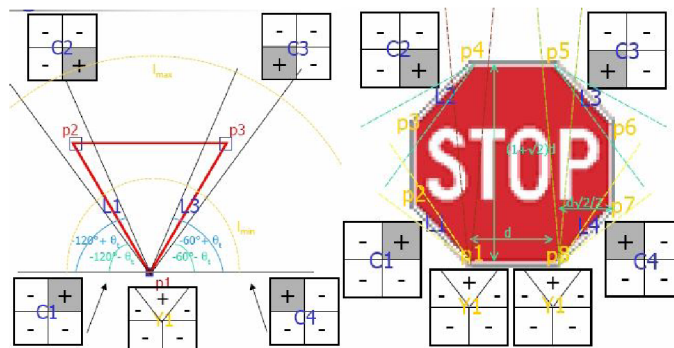


Obrázek 4-15: Hassan Shojania – a) detekce kruhu, b) detekce čtverce [8]

4.5.3 Detekce osmiúhelníku

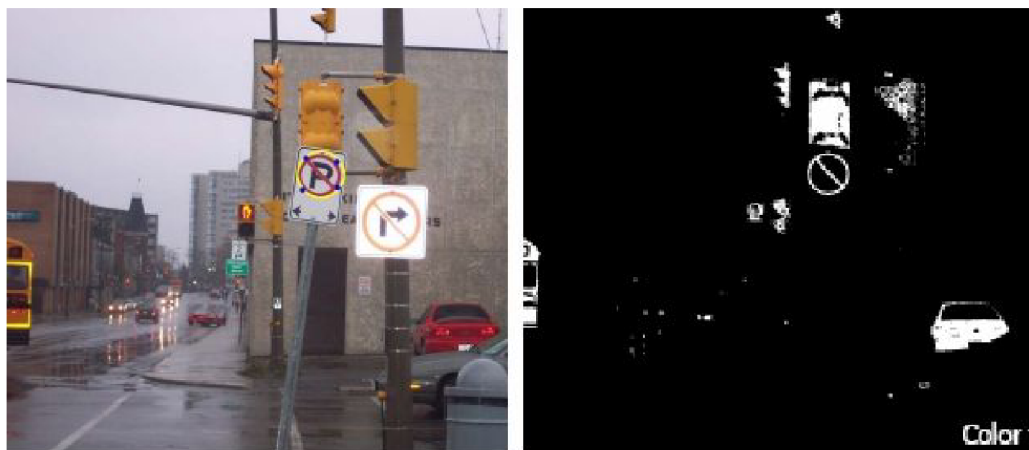
Postup je patrný z Obrázek 4-16 b).

- od bodu **p1** a **p8** nalezené maskou **Y1** hledáme **p4** a **p5** (masky **C2**, **C3**) v určité úhlem a vzdáleností vymezené oblasti.
- nalezneme zbývající rohy opět v oblastech vymezené patřičnou úhlovou oblastí.



Obrázek 4-16: Hassan – a) detekce trojúhelníka, b) detekce osmiúhelníku [8]

Na Obrázek 4-17 je zobrazena ukázka výsledku detekce. Je z něj patrné, že značka s významem **Zákaz odbočení vpravo** nebyla detekována, jelikož obsahuje barvu, která je mimo barevný práh. To by mohlo v případě odbočení řidiče stát pokutu vlivem porušení zákona. Proto tato metoda není příliš silná a doporučitelná pro další studium.



a)

b)

Obrázek 4-17: Hassan Shojania – ukázka výsledku detekce dopravních značek
a) originál s vyznačenými výsledky, b) barevná segmentace [8]

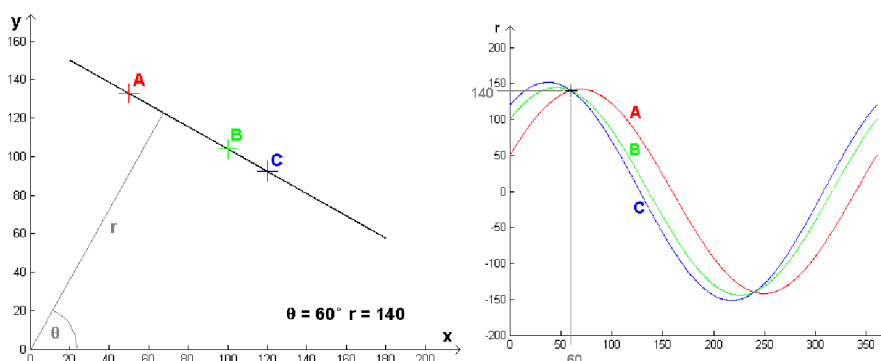
4.6 DETEKCE POMOCÍ HOUGHOVÝCH TRANSFORMACÍ

Houghova transformace [15] je metoda pro nalezení objektů v obraze, které lze analyticky popsat. Hodí se pro detekci přímek, kružnic a případně ji lze i modifikovat pro jiné tvary. Detekce houghových přímek se používá velmi často k detekci natočení objektů apod, jelikož jejím výstupem je úhel natočení a vzdálenost přímky (velikost normály) od počátku. Vstupním snímkem je obvykle obrázek na něž byl aplikován Cannyho hranový filtr [1]. Mezi výhody Houghovy transformace patří bezesporu jeho malá citlivost na šum, necitlivost k porušení hranice objektu, je schopný detekovat objekt i při částečném zakrytí jiným objektem. Nevýhodou je jeho časová náročnost a necitlivost k detekci blízkých rovnoběžných čar.

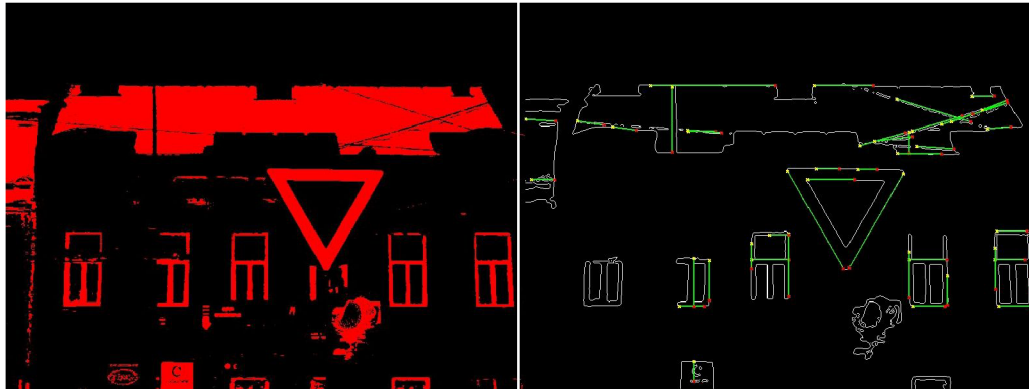
4.6.1 Detekce přímek

Předpokládejme rovnici přímky ve tvaru (6), kde r je délka normály přímky od počátku, φ je úhel mezi normálou a osou x . Přímka se mapuje na bod a bod na křivku. Intervaly hodnot řešení jsou $\varphi \in (0-360)$, $r \in (0, \text{uhlopříčka obrazu})$. Souřadnice x a y nalezeného bodu ve snímku jsou dosazeny do rovnice (6) a pro celý interval φ jsou vypočteny kombinace r . Data jsou postupně ukládána do akumulátoru přičemž výsledek s nejvyšší hodnotou odpovídá hledané kombinaci rovnice přímky.

$$r = x \cdot \cos \varphi + y \cdot \sin \varphi \quad (6)$$



Obrázek 4-18: Houghova transformace: mapování přímky na křivky



Obrázek 4-19: Houghova transformace: ukázka detekce přímek

Jak je zobrazeno na Obrázek 4-19 lze detekovat geometrické tvary i tímto způsobem. Po Houghově transformaci vznikne v obraze velké množství přímek a je nutné mezi nimi nalézt spojitosti, které by odpovídaly nějakému tvaru. Metoda je však zdlouhavá na výpočetní čas a tudíž není doporučeno ji aplikovat do real-time aplikací.

Houghova transformace pro přímky je využita i v navrženém rozhodovacím stromu pro detekci tvaru (kap. 4.2). Zde nejsou hledány Houghovy přímky přímo v celém obraze, ale pouze v prohledávaném a právě zpracovávaném objektu. Pakliže je objekt určen jako čtverec, obdélník či trojúhelník jsou v dalším algoritmu nalezeny body, kde se tyto přímky protínají. Ty jsou potřebné k dalšímu kroku, kterým je normování nalezené oblasti vzhledem k měřítku a perspektivě.



- `houghtest()`
- `houghvar()`

4.7 POROVNÁNÍ SE ŠABLONOU

Poslední metodou, kterou lze detekovat geometrické tvaru je velmi jednoduché porovnání se šablonou [23]. Vychází z metody, kdy je každá zkoumaná

oblast prověřena, zda splňuje podmínky poměru jejich velikostí stran (podobnost s příznakovým stromem 4.2). Pokud je poměr stran vyhovující a velikost objektu je dostatečná, je celá oblast převedena na normovanou velikost 50x50 px. Poté, je dvourozměrný snímek převeden na vektor a ten je porovnán se známými vektory tvarů, které jsou v paměti počítače.

Na počátku bylo nutné sestavit několik různých šablon v potřebném množství, které by odpovídaly jednotlivým tvarům dopravních značek. Byly vytvořeny šablony pro natočené trojúhelníky odpovídající značkám Dej přednost v jízdě (trojúhelník s vrcholem dole), trojúhelníky pro příkazové značky, kruhové masky pro detekci Zákazových apod. Každá maska je vyhotovena v několika provedeních s případnými deformacemi, aby byla opravdu zajištěna maximální shoda. Současně byla vytvořena i množina patvarů, které představují nechtěné segmentované objekty – člověk, automobil, odpadkový koš, strom apod (Obrázek 4-20). Tyto masky byly opět převedeny do normovaného tvaru 50x50px a jako jednotlivé vektory uloženy do matice.

Jak již bylo řečeno výše, nyní je zkoumaný objekt vektorového tvaru řádek po řádku přikládán k matici a je vypočtena jeho odchylka od zkoumaného řádku. Je předpokladem, že pokud zkoumáme objekt, který se podobá naučenému, bude rozdíl mezi těmito maticemi nejmenší. Trénovacích vzorů je proto pro každý tvar několik, aby se shoda skutečně našla. Odchylka od jednotlivých vzorů je vypočítávána podle (7) a ukládána do paměti.

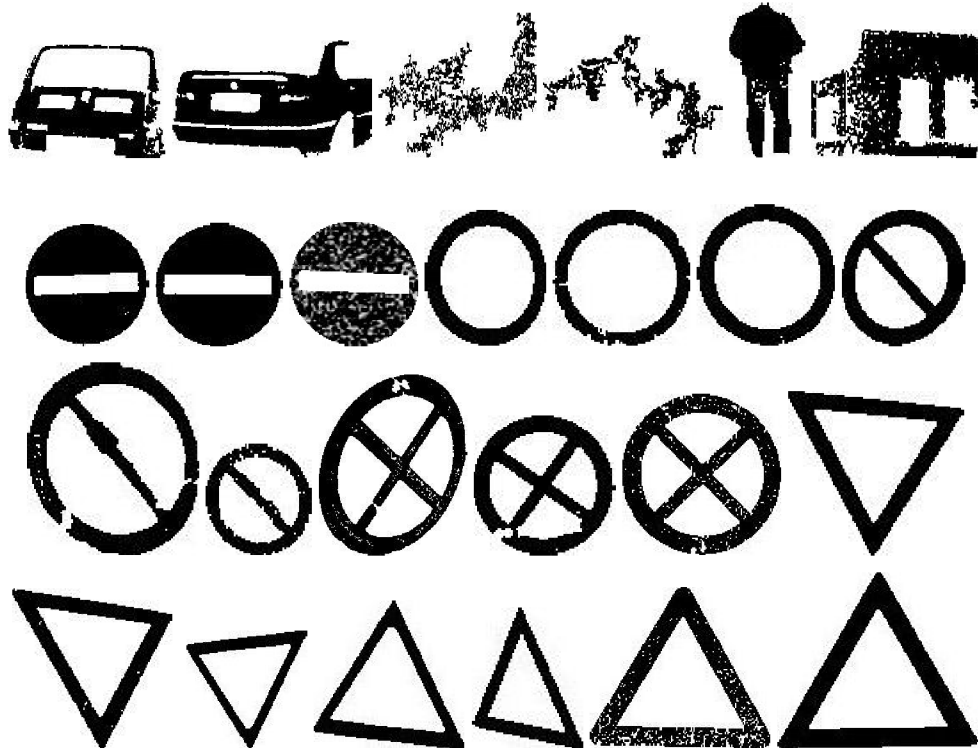
$$D(n) = |X - T(n)|^2 \quad (7)$$

Pouze pokud je rozdíl menší než stanovený práh jsou objekty považovány za shodné. Při nalezené shodě se vzorem například otočeného trojúhelníku tedy víme informace i o jeho rotaci a tak je poté velmi jednoduché tyto informace předat dalšímu kroku pro afinní operace. Systém rozlišuje tvary uvedené v příloze této práce na Obrázek 11-8 (číslo tvaru odpovídá programovému internímu pojmenování ID tvaru)



- rozpoznej_kohonen ()
- nauc_kohonen ()

Tato metoda porovnávání se šablonou, ač se zdá velmi primitivní, je nejspolehlivější a nejrychlejší z výše uvedených. Tento algoritmus je použit i k vyhodnocení vnitřních piktogramů, ale o tom až později.



Obrázek 4-20: Ukázka trénování množiny šablon, v horní řadě uvedeny patvary

5. NORMOVÁNÍ OBLASTI

Výstupem předchozího kroku je nalezení místa pravděpodobného výskytu dopravní značky. Již víme, že daná oblast odpovídá nějakému geometrickému tvaru a nyní je nutné rozhodnout, zda se nejedná o planý poplach, tedy falešnou detekci. Toho je docíleno tak, že detekovaná oblast je vybrána a zpracována pomocí afinní transformace souřadnic. Afinní zobrazení je geometrické zobrazení, při kterém odpovídají rovnoběžným přímkám opět rovnoběžné přímky, které však nemusí být rovnoběžné s původními přímkami. Ty se mohou skládat z jednotlivých operací jako je rotace, translace a zkosení, které jsou vyjádřeny v matici transformace.

Aby se nemusely jednotlivé hodnoty pracně vypočítávat využívá systém perspektivní transformace, které stačí zadat 4 body ze vstupního snímku a 4 body výstupního obrazu.

Pokud v předchozím kroku došlo k vyhodnocení tvaru pomocí rozhodovacího stromu s využitím příznaků, má každý tvar v paměti nalezené Houghovy přímky.

Když se jedná o trojúhelník, je nalezena vodorovná přímka a průniky dvou ostatních s touto přímkou dají dva body základny. Třetí bod (vrchol) je vypočten jako průsečík dvou stran mimo základnu. Poté je určen vektor spojující body základny a přenesen na vrcholový bod. Z parametrických rovnic pro úsečky jsou tak dopočteny další dva body.

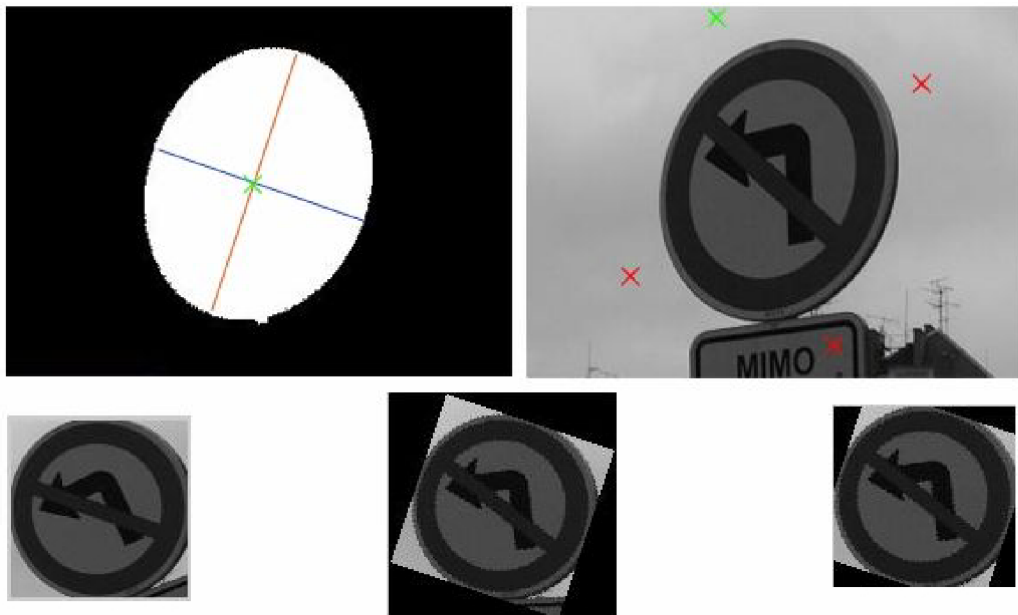
Pakliže se jedná o čtverec či obdélník jsou vypočteny 4 body jako průsečíky navzájem kolmých (nebo alespoň přibližně kolmých) přímek. Poté je nutné určit body v levém horním rohu, aby byla zaručena vždy stejná transformace. Tento bod se nalezne snadno – má nejkratší vzdálenost k počátku souřadnic. Zbylé tři body je nutné vhodně seřadit tak, aby jejich pořadí bylo po směru hodinových ručiček. Vlastní algoritmus na jejich řazení využívá velikost svírajícího úhlu s vodorovnou osou. Výsledkem jsou opět čtyři body, tentokrát ve správném pořadí.

U elipsy a kružnice je získání čtyřech bodů poněkud složitější. Z předchozího již nevíme žádné Houghovy přímky, ale pouze velikost majoritní a minoritní osy a střed. Pomocí postupných výpočtů s využitím základních algebraických rovnic pro parametrické vyjádření přímky a vektorového počtu jsou opět tyto čtyři body

nalezeny. Po aplikování perspektivní transformace je však výsledek orotován, což je pro detekci piktogramu nežádoucí. Proto je provedeno natočení obrazu s ořezem v záporném směru než bylo zjištěno natočení majoritní osy v prvním kroku. (Obrázek 5-1)



- `perspektiva_kruh()`
- `perspektiva_ctverec()`
- `perspektiva_trojuhelnik()`



- a) b)
c) d) e)

Obrázek 5-1: Ukázka normování nalezené elipsy

Na obrázku je uveden postup jak normování probíhá u nalezené elipsy: a) zobrazení majoritní a minoritní osy včetně středu, b) zobrazení nových bodů v originálním snímku (zelený bod je první), c) obrázek po aplikování perspektivní transformace, d) zpětně orotován o záporný úhel, e) finální normovaný snímek.

Pokud předchozím rozhodujícím krokem pro klasifikaci tvaru byla metoda porovnání se šablonou, je u každé šablony předem určena čtveřice bodů, které jsou vzhledem k měřítku šablony a skutečného objektu přepočteny a aplikovány na transformaci.



- `perspektiva_kruh2()`
- `perspektiva_ctverec2()`
- `perspektiva_trojuhelnik2()`

Z Obrázek 5-2 je vidět, že i trojúhelník s vrcholem dole, tedy Dej Přednost, je vždy transformován na trojúhelník s vrcholem nahoře. Je to kvůli dalšímu kroku, kdy dochází k extrakci piktogramu.



Obrázek 5-2: Normované oblasti 100x100 px

Doporučuje se, normovaným snímkům upravit jas a kontrast, aby vnitřní část byla pokud možno co nejvíce odpovídající skutečnému barevnému (či černobílému kontrastnímu) provedení.

6. IDENTIFIKACE PIKTOGRAMU

Na předchozí stránce je zobrazena ukázka několika normovaných oblastí, které s největší pravděpodobností obsahují dopravní značení. Nyní dochází k samotnému kroku extrakce vnitřního piktogramu, který rozhodne o jaký konkrétní druh dopravní značky se jedná či jde o falešnou detekci.

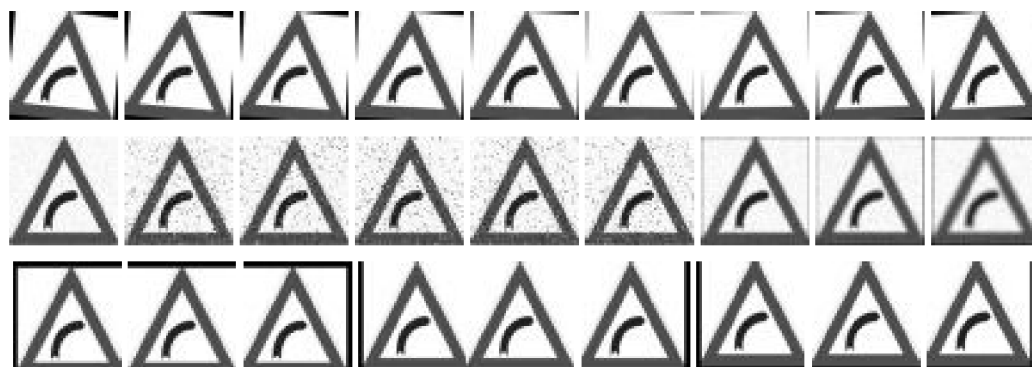
Jelikož je z předchozích kroků známo jaký tvar normovaná oblast má, je systém vybaven několika maskami pro přesnou extrakci. Masky jsou o trochu větší než je vnitřní oblast a do výsledku tak zanáší informaci i o části hrany značky. Toto rozšíření přináší lepší rozlišitelnost značek od falešných oblastí.

Metody, které identifikují vnitřní oblast jsou následující a byly během práce aplikovány k jejich vzájemnému porovnání. Testování bylo omezeno pouze na trojúhelníkové výstražné značky, kterých byla vytvořena velká počáteční množina tak, aby byla zajištěna univerzálnost algoritmů a naučení se na co největší množství dat patřících k jedné konkrétní značce.

Každý představitel značky měl vygenerováno celkem 40 vlastních mutací zahrnujících mírnou rotaci, translaci, přidání normálního šumu, přidání šumu pepř a sůl, rozmazání Gaussovským filtrem.



- `generuj_znacky()`

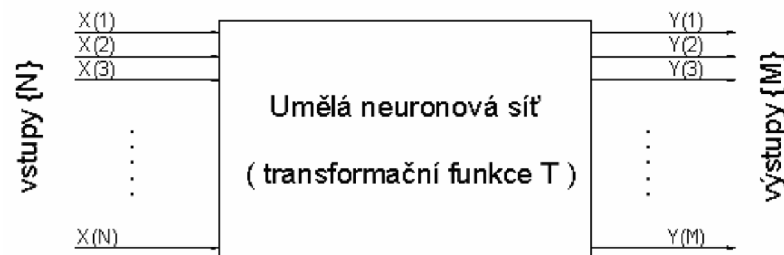


Obrázek 6-1: Ukázka trénovací množiny jednoho vzoru

6.1 UMĚLÉ NEURONOVÉ SÍŤE

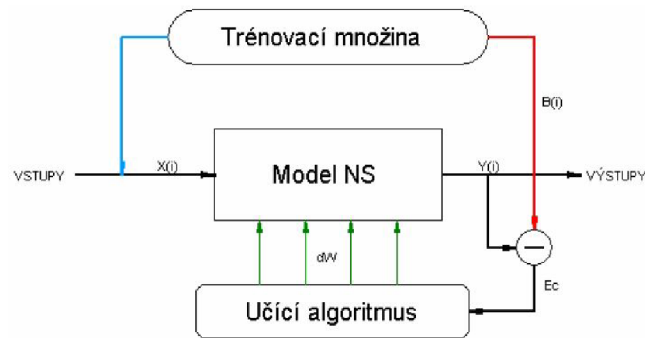
Umělá neuronová síť (UNS) [18], [20] je paralelní distribuovaný systém výkonných prvků modelujících biologické neurony účelně uspořádané tak, aby byl schopen požadovaného zpracování informací. UNS realizuje podle Obrázek 6-2 transformační funkci T zobrazující množinu vstupů $\{N\}$ na množinu výstupů $\{M\}$.

Principem výstupní funkce neuronové sítě tedy je aproximovat nějakou funkci podle vstupních vektorů. Klíčem k správné aproximaci je znát správně koeficienty, jež jsou vyjádřeny v matici transformační funkce T . Neuronovou síť lze naučit reagovat správně na vstupy a existují dva způsoby jejího naučení:



Obrázek 6-2: Umělá neuronová síť

1. **Učení neuronové sítě s učitelem** - při učení s učitelem se umělá neuronová síť učí tak, že srovnává svůj výstup s výstupem svého učitele. Poté nastavuje jednotlivé váhy mezi vstupy a vnitřními buňkami sítě tak, aby se snížil rozdíl mezi skutečným a požadovaným výstupem. Protože již při malém počtu vstupů je prostor definovaný koeficienty v matici obrovský, je nutno použít nějaký učicí algoritmus, efektivně hledající správné hodnoty koeficientů. Příkladem je například hledání řešení ve směru největšího gradientu. Algoritmus mění jednotlivé váhy a zjišťuje jejich vliv na výstupní vektor. Pokud se algoritmus zasekne na nějakém lokálním minimu tak zkusí rozhodit opět všechny koeficienty a zkusí najít tzv. lepší cestu.
2. **Učení bez učitele** - nemá žádný srovnávací signál k určení správnosti. Algoritmus je navržen tak, že hledá ve vstupních datech vzorky s určitými vlastnostmi tedy podle závislosti, korelace.



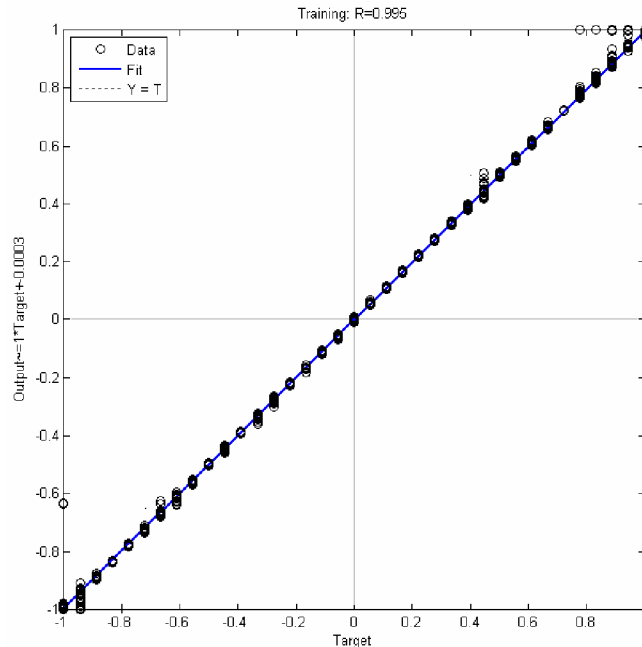
Obrázek 6-3: Umělá neuronová síť – učení s učitelem

UNS je schopná se naučit na jakákoliv vstupní data, která je nejprve nutné předem normalizovat do intervalu $\langle -1, +1 \rangle$ stejně tak i jeho výstup. V další části práce budou diskutovány ozkoušené metody, kdy byla měněna kombinace počtu vstupních i výstupních hodnot s jejich výsledky na separabilitu jednotlivých druhů. Byly testovány pouze piktogramy trojúhelníkových výstražných značek, kterých je dle vyhlášky 36. Do testovací množiny byla přidána i jedna množina obsahující falešná data – snímek automobilu, chodce apod.

6.1.1 Neuronová síť – M vstupů, 1 výstup, N hodnot

Jako první síť byla testována ta, která měla M vstupních hodnot a 1 výstup o 37 hodnotách. Jako vstupní data byly brány jednotlivé pixely pod maskou, jedná se tedy o metodu pixel-by-pixel. Výstup byl tvarován formou sigmoidy a to mělo za následek mírně nepřesné výsledky v krajních částech, kdy sigmoida dosahuje téměř horizontální hodnoty. Vnitřní vrstva byla vybavena pouze dvěma neurony vzhledem k rychlosti učení neuronové sítě.

Zvolený model neuronové sítě vykazuje separabilitu mezi jednotlivými prvky, přesto však není nejpřesnější.



Obrázek 6-4: Neuronové sítě: regresní funkce na výstupu sítě M-1-N

6.1.2 Neuronová síť – M vstupů, N výstupů, 1 hodnota

Další testovaná síť měla opět M vstupů představujících jednotlivé pixely obrazu, ale tentokrát byla vybavena 37 výstupy o stavech $\langle -1; +1 \rangle$. Výstupem byla opět sigmoida. Vnitřní vrstva sítě byla nyní vybavena 100 neurony.

Síť se po 10 000 epochách naučila rozpoznávat všech 37 druhů celkem správně. Může za to zejména to, že počet výstupu byl oproti minulému modelu zvětšen, takže byl požadován pouze jeden aktivní výstup celé sítě. Právě kvůli většímu modelu byla i vybavovací rychlost poněkud menší.

Přesto volba neuronové sítě s jednotlivými pixely na vstupu nebyla nejvhodnější, jelikož velmi závisí na světelných podmínkách a při učení sítě hrozí její přeučení na trénovací data. Pokud byly síti předkládány trénovací vzory, oba dva modely rozhodovaly velmi dobře. Pokud se síti předložil reálný snímek, úspěšnost rapidně klesla.



- NN_1x38.m
- NN_38x1.m

6.2 MOMENTOVÉ INVARIANTY

Momentové invarianty vycházejí z algebraických momentů a popisují základní geometrické charakteristiky obrazového segmentu jako statistické veličiny. S výhodou lze využít jejich rotační nezávislosti, na druhou stranu však neobsahují informace o prostorovém rozložení pixelů. Momentové invarianty mohou být geometrické resp. komplexní podle toho, zda se odvozují od geometrických resp. komplexních momentů. [6]

Jejich využití lze nalézt zejména u rozpoznávání objektů jednoduchých tvarů a rozdílné jasové barevnosti. Velikost momentového invariantu je totiž počítána váhově ke geometrickému těžišti daného objektu a jasové informaci bodů vlastním objektu. Z toho plyne, že jasnější body dále od těžiště vnášejí větší informace, proto se často pracuje s inverzním obrazem (potřebujeme-li detekovat převážně tmavé objekty). Takto získané momentové invarianty se poté porovnají se známými hodnotami a dle jejich podobnosti se určí o jaký známý, resp. neznámý objekt se jedná. Lze je tedy teoreticky využít i k základnímu rozpoznání některých druhů dopravních značek.

Geometrické momenty jsou fotometrické příznaky založené na regionech, jejich výpočet tedy vychází z tvaru objektu a jasových hodnot jeho pixelů. Geometrických momentů je pro každý objekt možné sestavit teoreticky nekonečně mnoho a čím více momentů popisuje daný objekt, tím je tento objekt popsán věrněji (je přesněji rekonstruovatelný). Základní geometrický moment řádu $p+q$ se stanoví podle vzorce:

$$m_{pq} = \sum \sum x^p \cdot y^q \cdot s(x, y) \quad (8)$$

Pro výpočet momentových invariantů segmentovaných objektů se používají centralizované geometrické momenty, vztažené zpravidla k těžišti objektu:

$$\mu_{pq} = \sum_Y \sum_X (x - x_t)^p \cdot (y - y_t)^q \cdot s(x, y) \quad (9)$$

Těžiště objektu lze vypočítat přímo z obrazových dat za použití základních geometrických momentů nultého a prvního řádu. Moment nultého stupně m_{00} je

prostý součet jasových hodnot segmentu, jehož mocnina slouží jako normalizační konstanta pro ostatní momenty. Normalizované momenty prvního stupně určují hodnoty souřadnic těžiště objektu. Pro souřadnici x_i těžiště platí vztah:

$$x_i = \frac{\sum_y \sum_x x \cdot s(x, y)}{\sum_y \sum_x s(x, y)} = \frac{m_{10}}{m_{00}}; \quad y_i = \frac{\sum_y \sum_x y \cdot s(x, y)}{\sum_y \sum_x s(x, y)} = \frac{m_{01}}{m_{00}} \quad (10)$$

Pro spolehlivé rozpoznávání je nutná normalizace hodnot geometrických momentů z hlediska velikosti objektů. Normalizovaný geometrický moment v_{pq} stupně $p+q$ je pak dán vztahem

$$v_{pq} = \frac{m_{pq}}{m_{00}^{\frac{p+q}{2}+1}} \quad (11)$$

U vztahu centrálního momentu μ_{pq} je výpočet vztažen ke geometrickému těžišti segmentu, je tedy prostorově nezávislý a výpočetní náročnost se zvýšila pouze o $x \cdot y$ odečetů. Z těchto centrálních geometrických momentů je pak pro rotačně nezávislé rozpoznávání vzoru nutné stanovit tzv. momentové invarianty. Momentové invarianty jak název napovídá jsou funkce geometrických momentů invariantní vůči určité třídě degradací jako je např. rotace, šum a obecně afinní transformace.

Sada sedmi invariantů sestavených na základě geometrických momentů třetího řádu mají tvar dle následujících vztahů.

$$\begin{aligned} \phi_1 &= \mu_{20} + \mu_{02} \\ \phi_2 &= (\mu_{02} - \mu_{20})^2 + 4 \cdot \mu_{11}^2 \\ \phi_3 &= (\mu_{30} - 3 \cdot \mu_{12})^2 + (3 \cdot \mu_{21} - \mu_{03})^2 \\ \phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\ \phi_5 &= (\mu_{30} - 3 \cdot \mu_{12}) \cdot (\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3 \cdot (\mu_{21} + \mu_{03})^2] + \\ &\quad + (3 \cdot \mu_{21} - \mu_{03}) \cdot (\mu_{21} + \mu_{03}) \cdot [3 \cdot (\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\ \phi_6 &= (\mu_{20} - \mu_{02}) \cdot [(\mu_{30} + \mu_{12})^2 - (\mu_{21} - \mu_{03})^2] + 4 \cdot \mu_{11}^2 \cdot (\mu_{30} + \mu_{12}) \cdot (\mu_{21} + \mu_{03}) \\ \phi_7 &= (3 \cdot \mu_{21} - \mu_{03}) \cdot (\mu_{30} + \mu_{12}) \cdot [3 \cdot (\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] - \\ &\quad - (\mu_{30} - 3 \cdot \mu_{12}) \cdot (\mu_{21} + \mu_{03}) \cdot [3 \cdot (\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (12)$$

6.2.1 Neuronová síť s momentovými invarianty

V práci je využito všech 7 momentových invariantů k rozpoznávání trojúhelníkových vnitřních piktogramů. Vstupem do algoritmu je vždy snímek společně s maskou pro extrakci. Vnitřní snímek je záměrně invertován, jak již bylo doporučeno výše. Je totiž nutné si uvědomit, že značky jsou převážně bílé a piktogram černý, takže by vzdálené bílé body zanesly do systému příliš mnoho nežádoucích informací a hodnota momentu by byla příliš velká.

Jelikož jsou hodnoty jednotlivých momentů normovány vzhledem k těžišti jsou výsledné momentové invarianty naopak velmi malé, řádově až $10e^{-15}$, proto je vhodné je zlogaritmovat.



- `moment()`
- `geo_teziste2()`
- `geom()`

Celá množina byla použita jako trénovací data pro neuronovou síť, která měla 7 vstupů, 7 vnitřních neuronů a 1 výstup. Bohužel se neuronová síť nebyla schopna naučit přesně rozlišovat jednotlivé piktogramy. Vliv na to má zejména použití pouze jednoho výstupu se sigmoidální výstupem a velmi malá separabilita vstupních dat.

Proto byla množina rozdělena na tři nové části, z nichž každá obsahovala 12 značek a jednu chybovou množinu dat. S menším počtem hodnot na výstupu se již neuronová síť byla schopná naučit s velmi přesnými výsledky. Při pokusech byly testovací obrazce předkládány třem neuronovým sítím a dvě z nich dle předpokladu odpověděly v převážné většině, že značku neznají přičemž třetí ji dokázala identifikovat. Ne vždy, ale tento postup byl platný. V mála případech odpověděly všechny tři sítě druhem značky a již možné rozeznat, která má pravdivé tvrzení.

6.2.2 Statistické zpracování momentových invariantů

Jelikož neuronová síť si momentovými invarianty příliš dobře neporadila, byl vyzkoušen další postup, tentokrát se statistickým zpracováním dat. Byl k tomu využit program RapidMiner [24, 25], který je schopný na základě vstupních dat a výstupního vektoru sestavit model odpovídající vzájemným vztahům mezi daty a výstupem. Tento pokus o rozpoznávání dopravních značek byl ozkoušen na základě zjištěných znalostí o tomto programu. Výstupem by měl být model, který lze snadno aplikovat na stávající data a vytvořit tak funkční rozpoznávací systém.

Celkový postup zpracování dat je uveden na Obrázek 6-5. V první fázi dojde k načtení a předzpracování dat jako je normování do intervalu. Následuje ohodnocení jednotlivých a výběr nejsilnějších atributů (v tomto případě momentových invariantů). Následně jsou data předána modelu. K naučení je použito metody CrossValidation, která z trénovacích dat vytvoří N stejně velkých množin a ty postupně předává modelu k naučení. Výsledkem je N různě nastavených modelů. Celková chyba se získá jako aritmetický průměr chyb jednotlivých modelů.

Jako modely byly v práci testovány tyto:

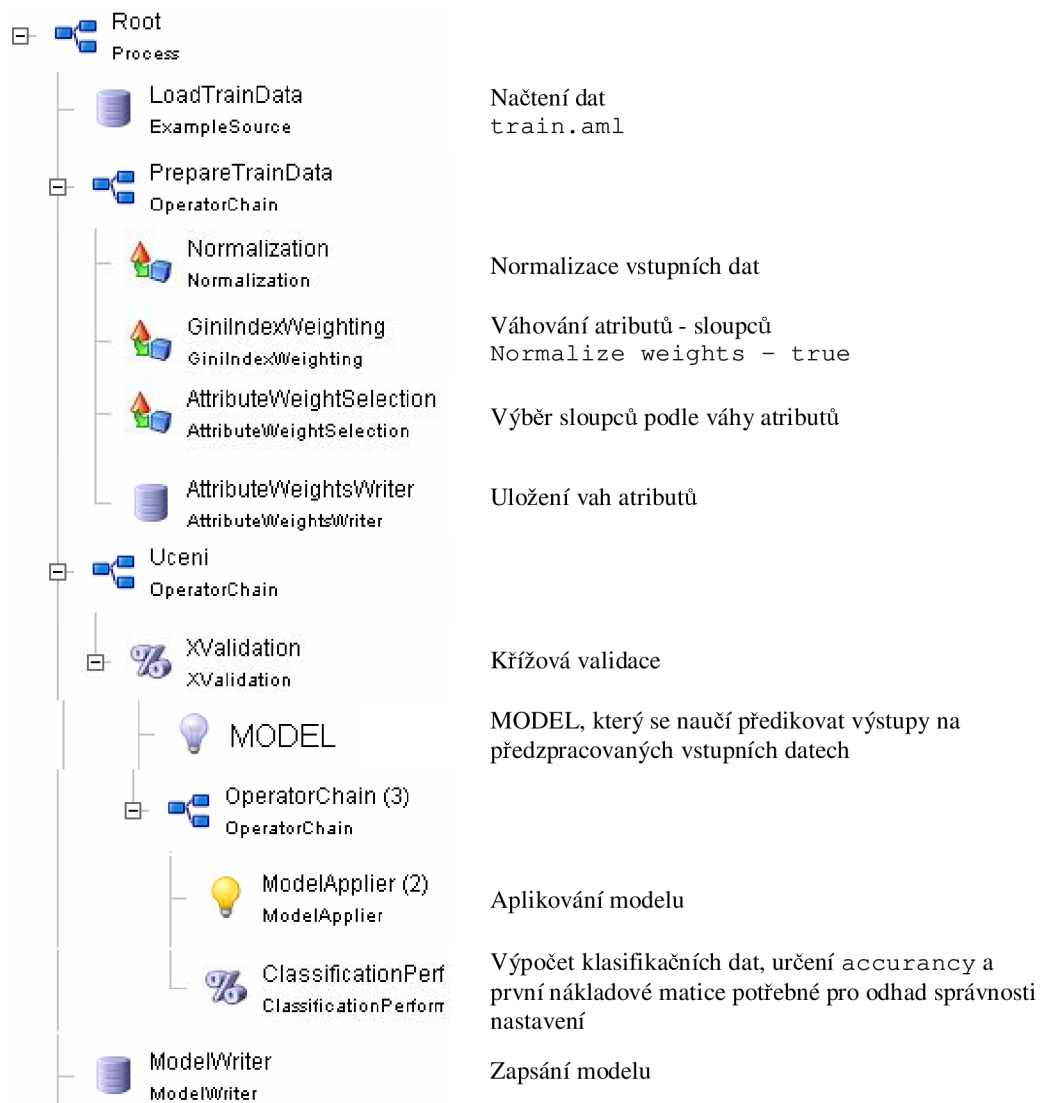
- Bayesovská síť, RandomTree, RandomForest

Metody pro naučení byly v práci testovány tyto:

- AdaBoost, Boosting, Bagging, CrossValidation

Vysvětlení, jak který model a metoda funguje překračuje rámeček této práce, proto je zájemcům doporučeno hlubší studium literatury [24].

Nakonec jako nejlepší postup pro zpracování byla zvolena metoda CrossValidation s RandomTree. Program RapidMiner vytvořil model rozhodovacího stromu na základě pěti momentových invariantů. Dosahoval úspěšnosti 84.5% jak je uvedeno v tabulce 6 umístěné v příloze této práce. Písmeno P značí předikovanou hodnotu a T značí správnou hodnotu. Z posledního řádku je možné vidět, které dopravní značky jsou lépe či hůře detekovatelné. Číslo 1-37 odpovídá pořadí značky v přehledu výstražných značek na Obrázek 11-1.



Obrázek 6-5: RapidMiner – struktura programu pro zpracování dat

Výstupní model stromu z programu RapidMiner má vlastní charakteristickou syntaxi. Proto byl navržen převodník, který dokáže tuto syntaxi transformovat pro jazyk MATLAB či C++. Převodník je naprogramován v jazyce JavaScript a je vložen do HTML stránky. Tu lze nalézt v příloze na datovém nosiči této práce.

Ukázka převodu syntaxe je patrná z Obrázek 6-6.

RAPID MINER – W-RandomTree	MATLAB
<pre> q6 < 0 q3 < 0 q1 < 0.2 : 4 (3/0) q1 >= 0.2 : 36 (2/0) q3 >= 0 : 4 (35/0) q6 >= 0 q6 < 0 : 36 (32/0) q6 >= 0 q1 < 0.22 q1 < 0.21 : 4 (2/0) q1 >= 0.21 : 14 (2/2) q1 >= 0.22 : 36 (25/0) </pre>	<pre> if (q6 < 0) if (q3 < 0) if (q1 < 0.2) FIN = 4; end; if (q1 >= 0.2) FIN = 36; end; end; %% maxlevel if (q3 >= 0) FIN = 4; end; end; %% maxlevel if (q6 >= 0) if (q6 < 0) FIN = 36; end; if (q6 >= 0) if (q1 < 0.22) if (q1 < 0.21) FIN = 4; end; if (q1 >= 0.21) FIN = 14; end; end; %% maxlevel if (q1 >= 0.22) FIN = 36; end; end; end; </pre>

Obrázek 6-6: RapidMiner – vlevo je výstup W-RandomTree z programu RapidMiner, vpravo transformovaný kód pro MATLAB pomocí JavaScriptu

Transformovaný kód je poté použit v programu MATLAB k rozhodování. Je však nutné mít na paměti, že je nutno dodržet stejné pořadí předzpracování vstupních dat jako tomu bylo v programu RapidMiner.



- `rozpoznej_randomTreeTroj()`

Strom sestavený na tomto principu dosahuje stejné přesnosti tedy 84.5% na vzorových datech. S reálnými snímky zde byl opět menší problém, který již byl zmiňován v části s momentovými invarianty.

6.3 POROVNÁNÍ SE VZOREM

Neuronové sítě vzhledem ke své náchylnosti na šum a nevelké separabilitě momentových invariantů se nejeví jako zcela vhodné řešení problému, proto byla na základě Occamova ostří¹ zvolena zcela jednoduchá metoda, která již byla diskutována ve stati o Rozhodování o tvaru (kap. 4.7). [23]

Vychází opět z principu, kdy jsou data z normovaného obrázku vybrána na základě masky a převedena na řádkový vektor. Je zde snížen počet rozlišení odstínů šedé na 10, jednoduchým vydělením hodnoty číslem 25.5 při 8-bitové reprezentaci barev.

Následně je vybrána vhodná matice na základě zjištěného tvaru – každý tvar má vygenerovanou vlastní matici pro rozhodování. Například je vytvořena matice pro výstražné trojúhelníky společně s chybovou množinou, značka Dej přednost v jízdě má opět vlastní množinu společně s chybami. Toto rozdělení je možné si dovolit, jelikož z předchozích kroků jsou známy orientace objektů. Další matice je uložena pro kruhové značky, další potom pro značky Hlavní silnice.

Postupně se hledá nejmenší kvadratická odchylka od vzoru. V algoritmu použitým v této práci je zanesena ještě podmínka, že se testovaný vektor musí alespoň částečně shodovat s více jedinci (řádky) u jednoho reprezentanta.

Metoda se ukázala jako nejvíce schopná a invariantní vůči jasovým změnám vstupního obrázku od vzoru. Proto taky byla aplikována ve finální verzi pro rozhodování.

Výjimku tvoří tvar odpovídající značení Zákazu zastavení, kdy je díky masce kontrolován počet modrých pixelů vně objektu. Pokud je počet úměrný ploše (cca 85%) je oblast prohlášena za Zákaz zastavení. Obdobně to probíhá s objektem odpovídajícím ID = 10 (Obrázek 11-8), kde je opět sečten počet modrých pixelů a pokud množství vyhovuje je oblast prohlášena za Zákaz stání. Pokud ne, je oblast podrobena rozhodování se vzory jako ostatní značky.

¹ Occamovo ostří - *nejjednodušší řešení bývá obvykle to nejsprávnější*

Každá skupina dopravního značení má unikátní barevnou kombinaci a tvar a díky této skutečnosti je možné výše uvedené způsoby rozpoznávání aplikovat. Vytvořením více rozpoznávacích matic (jedné ke každému tvaru) je velmi přínosné pro výpočetní výkon. Nemusí se totiž zbytečně počítat kvadratická odchylka pro všechny vzory.



- `rozpoznej_se_vzorem()`

6.4 POROVNÁNÍ METOD

Metody diskutované v předchozích částech byly během práce naprogramovány a vyzkoušena jejich funkčnost na reálných snímcích pořízených z ulic města. Neuronové sítě s metodou pixel-by-pixel vykazovaly velký stupeň separability a byly na rozhodování velmi rychlé. Přeci jen se jedná o obyčejné násobení a sčítání několika hodnot. Přesto však nedosahovaly příliš vysoké přesnosti na reálných snímcích. Pokud byly vypočteny momentové invarianty byla situace poněkud obdobná. Momentové invarianty se bohužel při malém rozlišení snímku stávají hůře separabilní což vede k časté chybovosti.

Jako nejlepší se ve výsledku ukázala metoda porovnávání se vzorem diskutována v kap. 6.3. Co se týče rychlosti, tak je závislá na velikosti trénovací množiny.

Metoda		Trénovací data	Reálné snímky
Pixel-by-pixel	M vstupů, 1 výstup, N hodnot	81 %	68%
	M vstupů, N výstupů, hodnota <-1;+1>	89%	79%
	Porovnání se vzorem	97%	91%
Momentové invarianty	7 vstupů, 1 výstup, N hodnot	82%	72%
	7 vstupů, 1 výstup 13 hodnot	79%	69%
	Rozhodovací strom v RapidMineru	85%	71%

tabulka 5: Porovnání metod pro rozpoznání piktogramu

7. SOFTWAREVÉ ŘEŠENÍ

Výsledkem této práce je prototypová aplikace vytvořená v programu MATLAB, který se v počátku řešení jevil jako nejsnazší možné řešení pro práci s obrazem. Postupem času byly jednotlivé algoritmy předělávány i do jazyka C++ programu Visual Studio rozšířeného o knihovnu OpenCV od firmy Intel [21]. Vzhledem k časové náročnosti přeprogramování algoritmů z jazyka jedné syntaxe na syntaxi druhou, nebyla aplikace v jazyce C++ úspěšně dokončena před termínem vyhotovení této práce. Co se týče časového porovnání, tak např. segmentace v HSV prostoru byla v jazyce C++ až 10x rychlejší. Přesto by bylo vhodné finální program rozdělit na vlákna a využít tak víceprocesorové jednotky na plný výkon a dosáhnout tak ještě rychlejšího zpracování.

Algoritmy pod programem MATLAB jsou funkční a pokud možno časově optimalizované pro maximální možnou rychlost detekce a rozpoznávání. Prototypové řešení v programu MATLAB je uloženo v souboru `finale.m` a je připraveno pro načítání jak videosekvence tak i jednotlivých snímků. Nastavení algoritmů je optimalizováno na snímky o rozlišení 640x480px.. Uživatel má možnost ve zdrojovém kódu volit z možnosti segmentace pomocí RGB, YCBCR a HSV diskutovaných v kap.3.3. Následuje krok předzpracování dat pomocí binárních operací eroze a dilatace. Dalším krokem je možnost volby rozhodovací metody o tvaru – lze využít buď rozhodovací strom s využitím radiometrických deskriptorů nebo porovnání se šablonou. Druhá zmiňovaná možnost je výrazně rychlejší. Každému nalezenému tvaru je ve vnitřní paměti dat uložena jeho poloha, popisná radiometrická data a transformovaný obraz, který je normován na vždy stejnou velikost. Normovaný obraz je ve finální fázi proložen maskou a dojde tak k extrakci pouze vnitřního piktogramu. Vzhledem k diskuzi metod uvedené v kap. 6 je možnost volby metody rozpoznávání na bázi momentových invariantů s využitím rozhodovacího stromu získaného v programu RapidMiner nebo využitím porovnávání se vzorem a hledání nejmenší kvadratické odchylky.

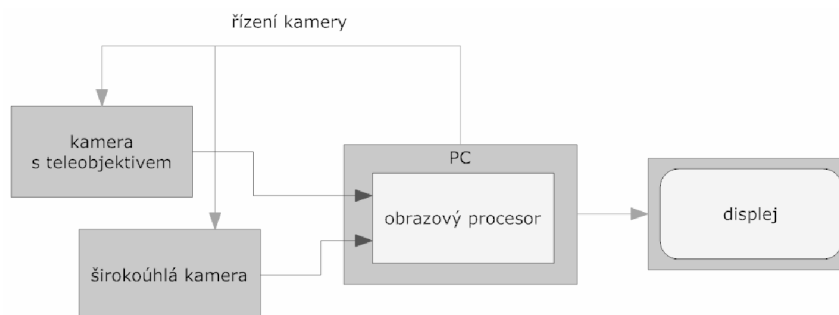
Finální částí aplikace je vhodné zobrazení a interpretování výsledků řidiči. Pro videosekvence je zde využita predikce značky v budoucích obrazech.

Geometrické tvary v červené a žluté vrstvě jsou rozpoznávány klasickou metodou porovnání se vzorem. Problém nastal s modrou vrstvou, kde mnohdy dochází k neúplnému výběru oblasti a tak je značka (např. IP4b - Jednosměrná ulice, Obrázek 11-4) detekována jako dvě nezávislé oblasti, přičemž tato detekci velmi ztěžuje a značka tak není rozpoznána. Dalším případem je hlavně to, že modré značky nemají v dané barvě pouze kruhový prstenec jako červené značky, ale jejich barva pokrývá celou plochu a vnitřní piktogram je pouze bílé barvy. Proto rozpoznávání tvarů v modré vrstvě přináší rovnou porovnávání se šablonami a vzorem všech dopravních značek. Již při tomto kroku je tedy rozpoznán druh značky a nemusí se, tak jako u červených a žlutých, provádět krok extrakce piktogramu a identifikace. Tento přístup se jeví dle testování na snímcích a testovací jízdě jako vhodná volba.

8. NÁVRH SNÍMACÍ SOUSTAVY

Jak již bylo zmíněno v úvodu této práce, tak se podpůrné systémy pro řidiče skládají ze základních částí jako je kamera a PC s displejem.

Miura [7] přichází se zajímavým konceptem snímací soustavy pro detekci dopravních značek. Jeho DAS pracuje na jednoduchém principu (Obrázek 8-1). Využívá dvě kamery, přičemž jedna sleduje po celou dobu obraz před kapotou automobilu (širokoúhlý objektiv) a druhá (telegrafický objektiv) pouze oblasti, které jsou zajímavé. Program v připojeném PC detekuje dopravní značky a rozpoznává jejich základní tvary. Pokud je značka v obraze detekována je obraz porovnán s uloženými obrazy v databázi a vyhodnocen. Poté systém čeká jak se zachová řidič na danou situaci. Systém je připojen k tachometru automobilu, proto lze vyhodnocovat, zda řidič zpomalil v případě omezené rychlosti. Jestliže systém nezaznamená změnu chování řidiče vyšle varovný signál - blikající světlo, výstražný zvuk, jinou signalizaci apod.



Obrázek 8-1: Schéma DAS

V tomto DAS systému je potřeba, aby kamery byly vhodně umístěny. Obvykle se montují do oblasti zpětného zrcátka. To umožňuje oběma kamerám, zejména té širokoúhlé, dobrý výhled vpřed a nebrání řidiči ve výhledu. Kamery jsou tvořeny obyčejnými digitálními fotoaparáty, PC je reprezentováno notebookem s displejem.

Širokouhlá kamera – využívá širokouhlého objektivu a sleduje dění před vozidlem. Neustále posílá snímky ke zpracování do PC, které rozhodne, zda je již ve snímku nějaká značka.

Fototelegrafická kamera – používá se ke spolehlivému rozpoznání barev a symbolů dopravní značky. Kamera je vybavena teleobjektivem a produkuje tak mnohem více informace potřebných k výsledné klasifikaci a tak si PC může přečíst jednotlivé informace na značkách. Kamera může měnit směr a zaostření k cílovému objektu, funguje pouze tehdy, pokud je k tomu z PC vydán příkaz.

PC a grafický procesor – má na starost zpracování obrazových údajů z obou kamer, detekuje přítomnost dopravní značky, klasifikuje její význam. Dle její polohy ve snímku ze širokouhlé kamery upraví nastavení telegrafické kamery a vyžádá si snímek v detailním záběru.

Displej – graficky reprezentuje výstupní data řidiči. Má na starost v konečné fázi signalizovat řidiči možná nebezpečí a varovat jej.

Gavrilla [4] zpracoval DAS se třemi kamerami. Dvě zajišťují prostorové vidění (lze tak odhadnout vzdálenost dopravní značky nebo chodce) a třetí je otočná a sleduje pouze zajímavá místa [7], stejně tak jak tomu bylo výše.



Obrázek 8-2: Koncept se třemi kamerami, Gavrilla [4]

9. VÝSLEDKY MĚŘENÍ A REALIZACE

9.1 OPTIMÁLNÍ NASTAVENÍ SNÍMACÍHO ZAŘÍZENÍ

V průběhu vývoje této práce bylo nutné stanovit omezující podmínky pro provoz algoritmu. Proto bylo pomocí snímacího zařízení a laserového dálkoměru provedeno terénní měření, kde se zjišťoval vliv vzdálenosti na velikost značky. Snímací zařízení společně s laserovým dálkoměrem byly umístěny na stativu a postupně posouvány blíže ke značce od vzdálenosti 50 do 5 metrů v 1 a 2 metrových intervalech.

Laserový dálkoměr

Leica DISTO A5



Stativ

Velbon Sherpa 600R



Digitální fotoaparát

CANON S850



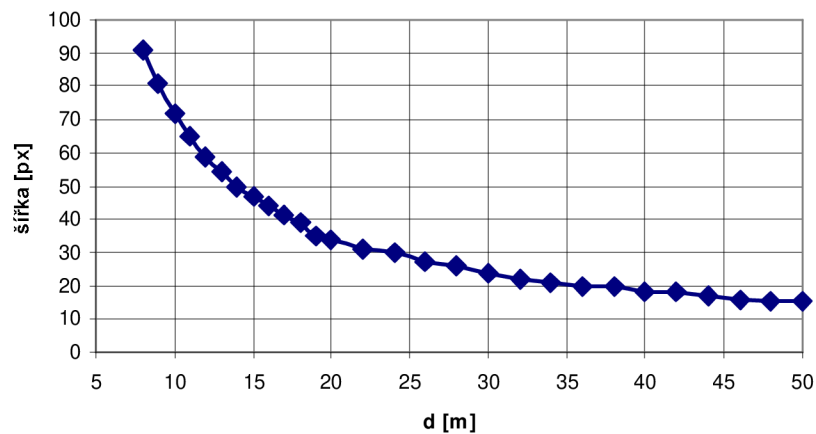
Obrázek 9-1: Měřicí zařízení

Digitální fotoaparát snímal scénu se zamknutým automatickým ostřením, se zamknutou expozicí nastavenou na čas 1/500 s při ohniskové vzdálenosti 7,8mm (38mm ekvivalentní 35mm objektivu) a snímky ukládány ve velikostech 800x600px.

Jak vyplývá z průběhu grafu (Obrázek 9-2), kde je uvedena změna šířky dopravní značky měřené v pixelech v závislosti na změně vzdálenosti, je tato změna nelineární a exponenciálně roste s klesající vzdáleností. Při určování tvaru a identifikaci piktogramu se osvědčila jako hraniční minimální rozlišovací hodnota

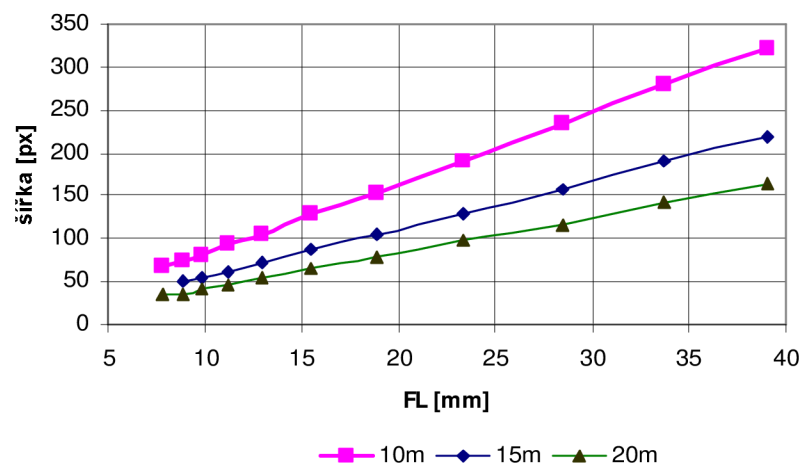
50px, které objektiv dosáhne při vzdálenosti 12 metrů od dopravní značky což je docela pozdě.

Průběh šířky dopravní značky při změně vzdálenosti automobilu



Obrázek 9-2: GRAF: průběh šířka značky při změně vzdálenosti automobilu

Průběh šířky dopravní značky vlivem změny ohniskové vzdálenosti objektivu



Obrázek 9-3: GRAF: průběh šířka značky vlivem změny ohniskové vzdálenosti objektivu

Posuv může nastat především změnou ohniskové vzdálenosti o jejímž vlivu na změnu šířky je další graf (Obrázek 9-3). Bylo provedeno obdobné měření s tím rozdílem, že se z jedné vzdálenosti měnilo nastavení objektivu a jeho ohnisková vzdálenost. Celkem pro tři různé vzdálenosti. Opět si lze všimnout, že tentokrát je změna lineární a hraniční oblasti šířky 50px lze dosáhnout i ze vzdálenosti 20ti metrů při ohniskové vzdálenosti 13mm. Po tomto zjištění bylo pořízeno několik videosekvencí již s novým nastavením objektivu.



Obrázek 9-4: Ukázka měřicích dat pro změnu ohniskové vzdálenost

9.2 UKÁZKA DETEKCE

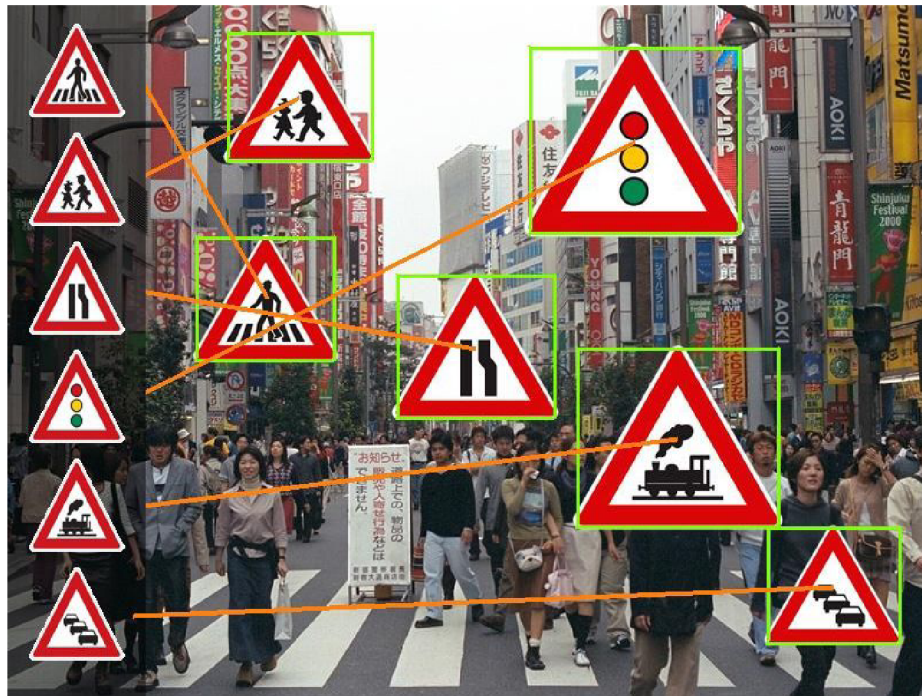
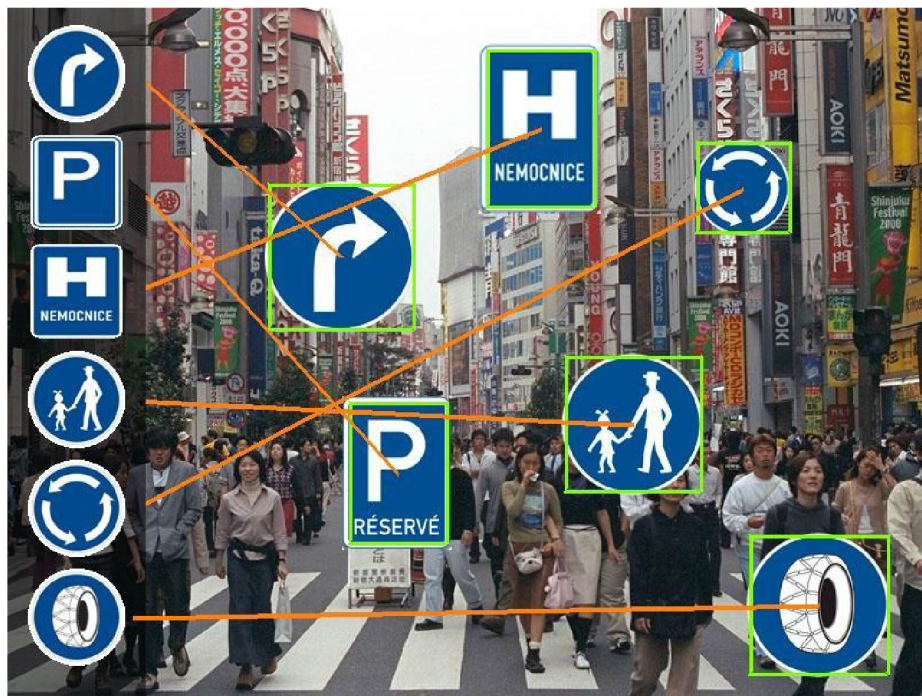
Připravený algoritmus je optimalizován na RGB vstupní snímky, které si automaticky převede na velikost 640x480px. Jako nejlepší zvolená kombinace metod se jeví segmentace v HSV prostoru, vyhodnocení tvaru pomocí shody se šablonou a klasifikace pomocí porovnání se vzorem.

Navržený algoritmus byl testován nejen na snímcích, které byly pořízeny v městském i mimoměstském prostředí (viz. Kap. 3.2), (Obrázek 9-6), ale i na videosekvencích pořízených z jedoucího automobilu. Testovací jízda probíhala tak, že skrze střešní okno byl na stativu umístěn digitální fotoaparát (Obrázek 9-1) a vysunut mírně nad horizont automobilu. Nastavení objektivu bylo provedeno s přiblížením (větší ohniskovou vzdáleností) a mírným natočením směrem k pravé straně vozovky, kde se předpokládá největší výskyt dopravních značek. Během testovacích jízd bylo pořízeno několik desítek minut záznamu v městské zástavbě.

Následně na videosekvence byl aplikován rozpoznávací algoritmus s velmi dobrými výsledky. Ty lze shlédnout jako videosoubory přiložené na datovém nosiči této práce. Úspěšnost celkové detekce byla přibližně 93% ve všech barevných kombinacích. Systém je velmi přesný co se týče rozpoznávání piktogramu trojúhelníkových tvarů. Modré značky se bohužel hůře rozpoznávají především proto, že jejich modrá barva vyplňuje celý obrazec. Přesto se na testovacích snímcích (Obrázek 9-5) povedlo rozpoznat všechny symboly modré barvy.

Testovací snímek je záměrně vyprodukován na reálném pozadí, aby byla vidět i síla algoritmu co se týče segmentace.

Časová náročnost celého rozpoznávacího algoritmu je při daném rozlišení 1.2 vteřiny. Pokud se v nastavení algoritmu nastaví pouze červený kanál, lze celý snímek zpracovat již za 500ms. Přesto i tato hodnota je bohužel vysoká, jelikož se jedná o prototypové řešení programu MATLAB. Je jisté, že pokud by byly algoritmy napsány v jazyce C++ zmiňovaném výše, jistě by se dosáhlo i vyšší rychlosti zpracování dat a přiblížili bychom se tak, alespoň na hodnotu 100ms na jeden snímek, což by bylo adekvátní.



Obrázek 9-5: Ukázka výstupu při detekci na testovacích snímcích



Obrázek 9-6: Ukázka výstupních snímků jak je algoritmus reprezentuje řidiči

9.3 OMEZUJÍCÍ PODMÍNKY

V předchozí části byla nakousnuta problematika rychlosti zpracování detekce dopravní značky. Čas je velice kritické měřítko. Pro názornost se podívejme na rovnici (13), kde je výpočet dráhy automobilu jedoucí konstantní rychlostí 50km/h po dobu 1 vteřiny. Za tento čas automobil urazí téměř 17 metrů.

$$v = 60 \text{ km/h} = 16,66 \text{ m/s}, t = 1 \text{ sec},$$

$$v = \frac{s}{t} \Rightarrow s = v \cdot t \quad (13)$$

$$\underline{\underline{s = 16,66m}}$$

Proto je velmi nutné zpracovat několik snímků za vteřinu, aby nebyla žádná značka vynechána. Přeci jen rozlišovací schopnost systému je taková, že dokáže zachytit a přesně určit značku až v blízkosti cca 15m.

Dalším omezením pro funkčnost algoritmů jsou světelné podmínky. Prozatím byl systém vyvíjen pro denní světlo. Při nočním osvětlení není jeho funkčnost testována a tedy ani garantována.

10. ZÁVĚR

Práce se zabývala detekcí dopravních značek v rozmanitém prostředí a omezila se pouze na dopravní značení platné na území České republiky. Algoritmus využívá barevných snímků vzhledem k tomu, že jednotlivé tvary a barevné kombinace jsou dané vyhláškou a platnou normou.

Byly postupně diskutovány jednotlivé metody barevné segmentace společně s jejich hodnocením na kvalitu výsledku. Navržené metody pracují na principu segmentace v RGB, YCBCR a HSV barevných prostorech. Jako nejspolehlivější se ukázala metoda pomocí HSV. Po detekci vznikly tři masky, které vyjadřují pravděpodobná místa výskytu značky.

V dalším kroku bylo rozhodnuto, které tvary mohou skutečnými značkami být, a které tvoří pouze falešnou detekci. Byla navržena vlastní metoda pomocí rozhodovacího stromu založeného na radiometrických deskriptorech jednotlivých oblastí. Dále byly v práci uvedeny metody pro detekci geometrických tvarů, které byly nalezeny v použité literatuře a směřují tak čtenáře k dalším alternativám. Jako nejspolehlivější metoda se ukázala ta, která jednotlivé objekty porovnávala s maskami známých tvarů uloženými v databázi. Tímto krokem v obraze zůstaly pouze místa s nejvyšší pravděpodobností výskytu a tak se z originálního snímku tyto oblasti extrahovaly a normovaly perspektivní transformací na pevný rozměr.

Důležitým krokem je samozřejmě rozpoznání vnitřního piktogramu dopravní značky. Byly zde uvedeny a ozkoušeny metody rozhodující pomocí naučených neuronových sítí se vstupními informacemi v podobě jednotlivých pixelů nebo se vstupními hodnotami momentových invariantů získaných z obrazu. Pro momentové invarianty byl využit externí program RapidMiner, který je vhodný pro statistické zpracování dat. Za jeho pomoci byl navržen rozhodovací strom. Bohužel RapidMiner má pro výstupní data velmi specifickou syntaxi, proto byl vytvořen v JavaScriptu převodník do jazyka MATLAB a C++ přiložený na datovém nosiči této práce. Další metodou, ve finální práci aplikovanou a ve výsledku nejúspěšnější, se osvědčilo porovnání se vzorovými daty a hledání nejmenší kvadratické odchylky od vzoru.

Výsledkem celé práce je funkční algoritmus napsaný v jazyce programu MATLAB. Jeho výstupem jsou jednotlivé snímky, na kterých je vizuálně zobrazena korespondence se vzorovými znaky. Jelikož je v programu uložen i název dopravní značky není problém algoritmus rozšířit i o jeho využití například o akustické předčítání významu. Korespondence uvedená na Obrázek 9-5, kde jsou zobrazeny oranžová vodítka mezi nalezeným vzorem a symbolem v obrazovce, jsou zobrazena algoritmem automaticky zejména pro demonstraci výstupu. Pakliže by byl algoritmus nasazen ve skutečném provozu navrhuji čáry vynechat – při větším počtu může být jejich spleť matoucí. Při detekci videosekvence je systém obohacen o mírnou predikci, kde se bude značka v budoucnu vyskytovat. Pro vylepšení navrhuji využití například Kalmanova filtru.

Systém je v tuto chvíli schopen přesně rozpoznat 37 druhů výstražného značení, 30 zákazových značky a částečně i modré informativní.

Ukázka výstupní videosekvence je umístěna na datovém nosiči přiloženém v této práci ve složce „DEMO“. Doporučuji její shlédnutí pro demonstraci výsledků.

Při tvorbě práce byla využita znalost hned z několika předmětů vyučovaných na VUT v Brně. Jedná se zejména o MPOV (Počítačové vidění), MAPV (Aplikace počítačového vidění), MSTU (Strojové učení), MUIN (Umělá inteligence), MIAP (Internetové aplikace).

Tato práce se zúčastnila konference STUDENT EEICT 2009 a umístila se na 2. místě v kategorii Magisterských projektů v části Zpracování signálů a obrazu [26].

11. POUŽITÉ ZDROJE

- [1] HLAVÁČ, ŠONKA, *Počítačové vidění*. Praha: Grada, 1992, ISBN 80-85424-67-3
- [2] HLAVÁČ, V. *Hledání hran* [online]. Fakulta el. ČVÚT v Praze, 2006. [cit. 15.12.2006]. Dostupné z <<http://cmp.felk.cvut.cz/~hlavac>>.
- [3] WIKIPEDIA, *Otevřená encyklopedie, heslo: Barevný prostor* [online], poslední revize 20. 4. 2008. [cit. 1.5.2008]. Dostupné z <<http://cs.wikipedia.org/wiki/Kategorie:Barva>>.
- [4] GAVRILA, D. M. *Traffic Sign Recognition Revisited* [online], poslední revize 10. 3. 1999, [cit. 5. 5. 2008]. Dostupné z <<http://www.gavrila.net/Publications/dagm99.pdf>>.
- [5] WIKIPEDIA, *Otevřená encyklopedie, heslo: Distance transform* [online], poslední revize 29. 4. 2008. [cit. 5.5.2008]. Dostupné z <http://en.wikipedia.org/wiki/Distance_transform>.
- [6] HORÁK, K. *Aplikace metod rozpoznávání obrazu v defektoskopii*, disertační práce, Brno VUT 2007
- [7] MIURA, J. *An Active Vision System for Real-Time Traffic Sign Recognition*, Dearborn USA: IEEE, 2000, ISBN 0-7803-5971-2
- [8] SHOJANIA, H., *Real-time traffic sign detection* [online], poslední revize 12. 1. 2007. [cit. 5.5.2008]. Dostupné z <<http://hassan.shojania.com>>.
- [9] HORÁK, K. *Počítačové vidění – přednáška č. 9, Popis objektů, příznakový prostor* [online], poslední revize 26.11.2007. [cit. 8.5.2008]. Dostupné ze školní sítě VUT.
- [10] WIKIPEDIA, *Otevřená encyklopedie, heslo: Pozemní komunikace v Česku* [online], poslední revize 30. 9. 2007. [cit. 5.5.2008]. Dostupné z <[http://cs.wikipedia.org/wiki/Kategorie:Pozemní komunikace v Česku](http://cs.wikipedia.org/wiki/Kategorie:Pozemní_komunikace_v_Česku)>.
- [11] *Ministerstvo dopravy České republiky* [online], Dostupné z <<http://www.ministerstvodopravy.cz>>
- [12] VOJÁČEK, A. [online], poslední revize 3. 5. 2007. [cit. 10.12.2008]. Dostupné z <<http://automatizace.hw.cz/view.php?cislocclanku=2007042901>>.

- [13] WINKLER, Z. Měření rychlosti [online], poslední revize 12. 5. 2003. [cit. 10.12.2008]. Dostupné z <<http://robotika.cz/guide/filtering/en>>.
- [14] DUDA, R. O., HART, P. E. *Use of the Hough Transformation to Detect Lines and Curves in Picture*, Comm. ACM, Vol. 15, pp. 11–15 (January, 1972)
- [15] PERKINS, S. F., WOLFART, E. *Hough transform* [online], poslední revize 12. 5. 2003. [cit. 10.12.2008]. Dostupné z <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>>.
- [16] SOETEDJO, A., YAMADA, K. *An Efficient Algorithm for Traffic Sign Detection*, Journal of Advanced Computational Intelligence, Vol. 10, No. 3, 2006, pp. 408-418
- [17] VOMEL, J. *Genetické algoritmy* [online], Praha VŠE 2008, poslední revize 29. 4. 2008. [cit. 19.12.2008]. Dostupné z <<http://staff.utia.cas.cz/vomel/>>.
- [18] ŠÍMA, J., NERUDA, R. *Teoretické otázky neuronových sítí*, Praha: MatfyzPress 1996
- [19] SHNEIER, M. *Road Sign Detection and Recognition* [online], 2005, 6 s., IEEE Computer Society International Conference on Computer Vision and Pattern Recognition, June 2005, Dostupné z <http://www.isd.mel.nist.gov/documents/shneier/Road_Sign_Detection.pdf>.
- [20] LORSAKUL, A., SUTHAKORN, J. *Traffic Sign Recognition Using Neural Network on OpenCV* [online], 2006, Faculty of Engineering, Mahidol University, Thailand, Dostupné z <<http://www.bartlab.org/publication.html>>
- [21] *Intel OpenCV* [počítačový program, online] verze 1.1., knihovna pro Visual Studio umožňující práci s obrazem v jazyce C++, open source, Dostupné z <<http://opencv.willowgarage.com/wiki/>>
- [22] *Přehled dopravního značení* [online], poslední revize 10. 5. 2009. [cit. 15.5. 2009]. Dostupné z <<http://www.dopravni-znaceni.eu/>>
- [23] ZÁMEČNÍK, D. *Rozpoznávání dopravních značek s využitím neuronové sítě*, Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 90s. Vedoucí diplomové práce doc. Ing. Václav Jirsík, CSc.

- [24] HONZÍK, P. *Strojové učení*, Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2006. 85 s.
- [25] *RapidMiner* [počítačový program, online, příloha CD], Program pro datamining a statistické zpracování dat, open source.
Dostupné z <<http://www.rapidminer.com>>
- [26] ČÍP, P. *Traffic signs detection and recognition*. In *Proceeding of the 15th Conference STUDENT EEICT 2009 : Volume 2*. Brno VUT : [s.n.], 2009. Zpracování signálů a obrazu. s. 69-71.
Dostupný z WWW: <<http://feec.vutbr.cz/EEICT/>>.
ISBN 978-80-214-3868-2.

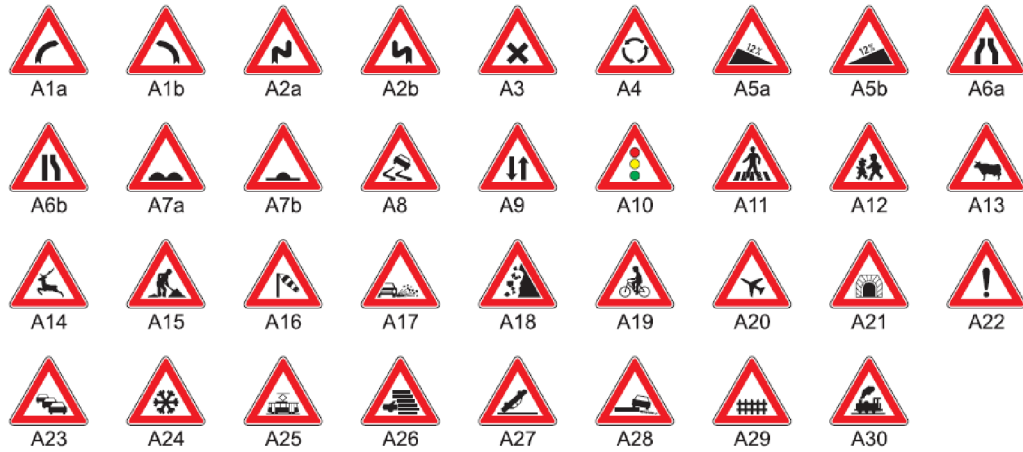
SEZNAM ZKRATEK A SYMBOLŮ

- **DAS** – z angl.: driving asistance system – systém pro podporu řízení
- **DT** – z angl.: distance transform – vzdálenostní transformace
- **RGB** – z angl.: red-green-blue – červený-zelený-modrý, barevný prostor
- **HSV** – z angl.: hue-saturation-value – odstín-saturace-jas, barevný prostor

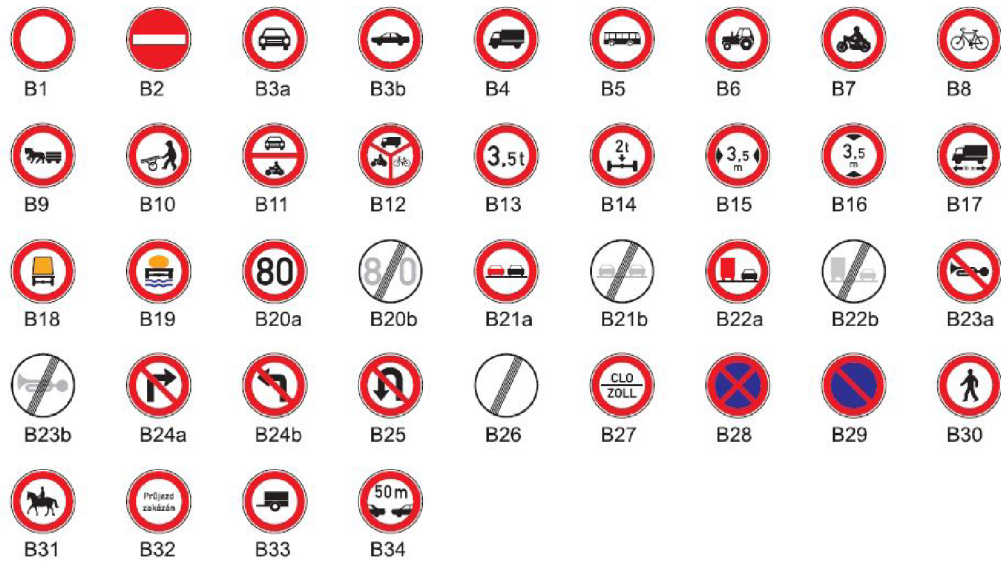
SEZNAM PŘÍLOH

- 4x strany grafických příloh (strany 88-91)
- 1x DVD 4.7GB
 - DEMO – ukázkové videosekvence s výslednou detekcí
 - VIDEO – složka s video soubory
 - ZNAČKY – obrazová data symbolů značek
 - ZNAČKY REÁLNÉ – snímky reálných značek
 - MATLAB – programové řešení aplikace
 - OPENCV – programové řešení aplikace
 - PŘEVODNÍK – převodník pro syntaxi RapidMiner

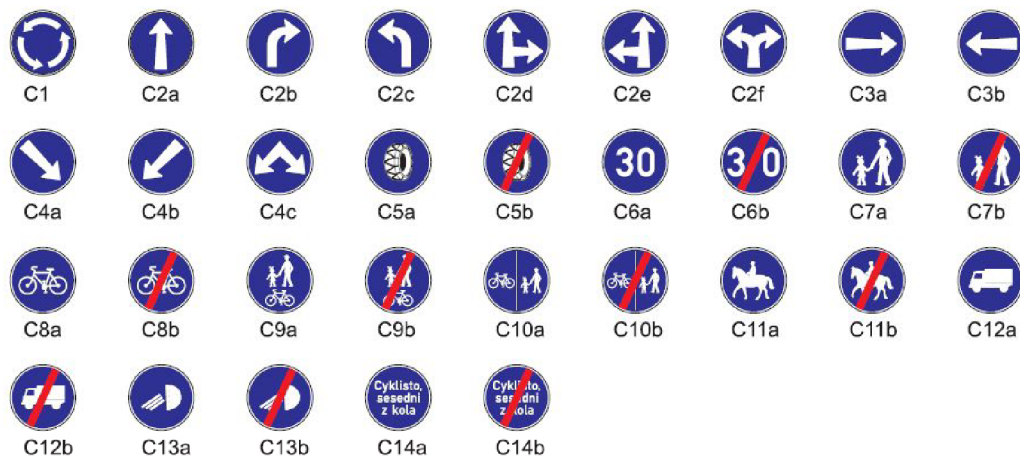
PŘÍLOHY



Obrázek 11-1: Dopravní značení – výstražné značky [10]



Obrázek 11-2: Dopravní značení – Zákazové značky [10]



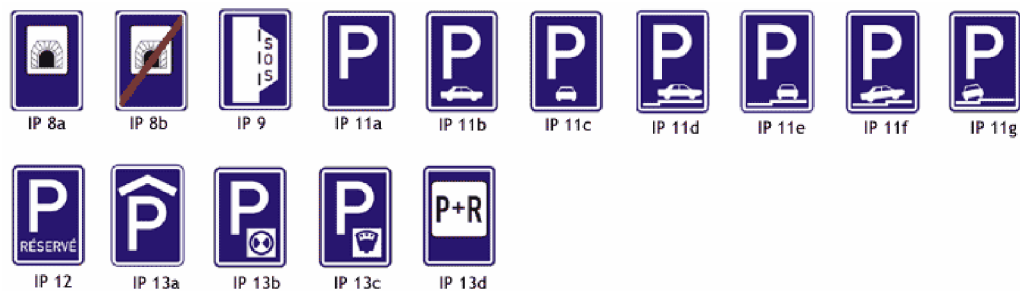
Obrázek 11-3: Dopravní značení – Příkazové značky [10]



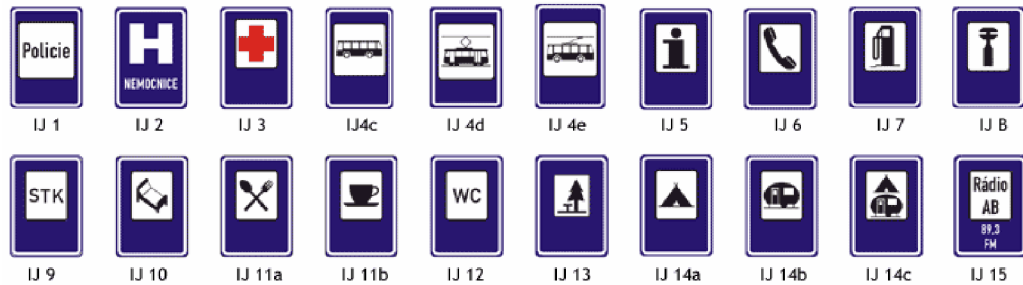
Obrázek 11-4: Dopravní značení – Informativní značky [10]



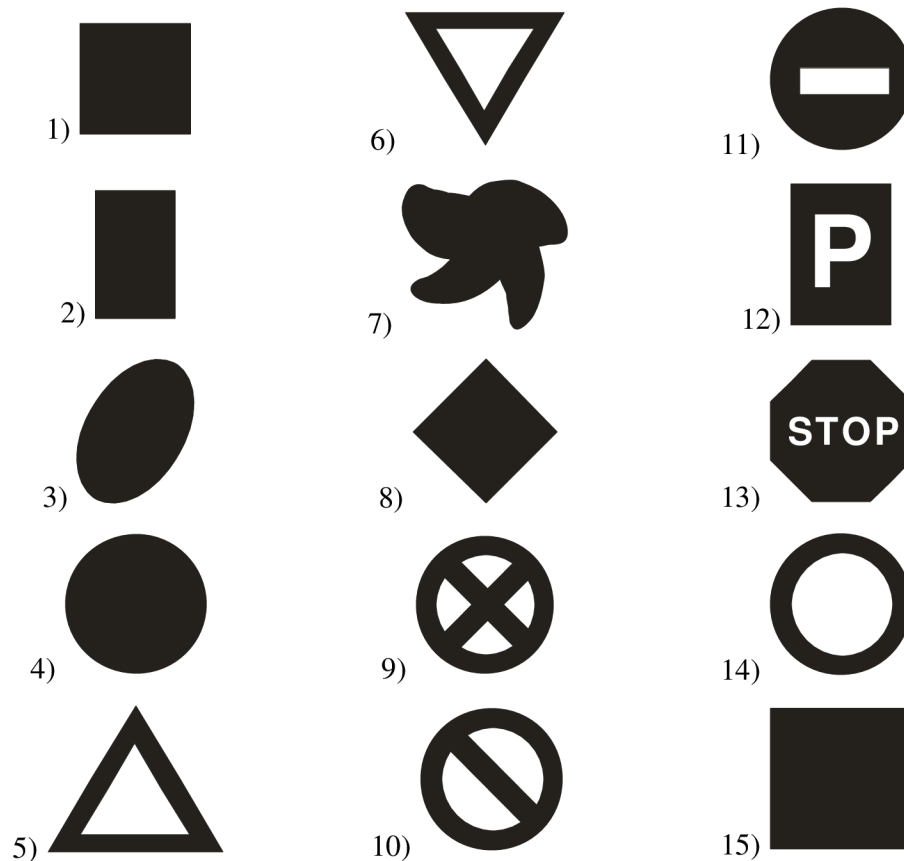
Obrázek 11-5: Dopravní značení – značky upravující přednost vozidel [10]



Obrázek 11-6: Dopravní značení – Informační značky provozní [10]



Obrázek 11-7: Dopravní značení – Informativní značky [10]



Obrázek 11-8: Tvary, které systém rozlišuje – číslo odpovídá proměnné ID_tvar

