

# 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

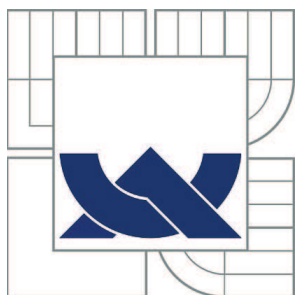
DETEKCE JÍZDNÍCH PRUHŮ A PŘEKÁŽEK

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

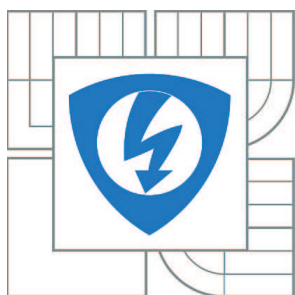
Bc. MARIAN DOJAVA

BRNO 2011



**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 JÍZDNÍCH PRUHŮ A PŘEKÁŽEK**

TRAFFIC LANES AND INTERRUPTIONS DETECTION

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

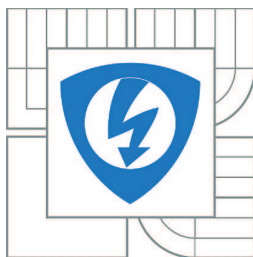
**Bc. MARIAN DOJAVA**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. KAREL HORÁK, Ph.D.**

BRNO 2011



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. Marian Dojava

**ID:** 73065

**Ročník:** 2

**Akademický rok:** 2010/2011

**NÁZEV TÉMATU:**

**Detekce jízdnicích pruhů a překážek**

**POKYNY PRO VYPRACOVÁNÍ:**

Úkolem studenta je teoreticky zpracovat možnosti použití kamer ve vozidle nebo na vozidle jako součást asistenčního systému řidiče. Praktická část spočívá v detekci vozovky, krajnice a hranice jízdnicích pruhů před vozidlem na snímcích pořízených kamerou z interiéru vozidla. Součástí vizuálního systému je detekce jízdnicích překážek před vozidlem.

**DOPORUČENÁ LITERATURA:**

Hlaváč, V., Šonka, M.: Počítačové vidění. Praha: Grada, 1992.

Haußecker H., Geißler P.: Handbook of Computer Vision and Applications. San Diego: Academic press, 1999.

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 23.5.2011

**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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Úkolem této diplomové práce je popsat využití kamery jako snímače pro asistenční systém automobilu. Byly navrženy způsoby nalezení silnic, jízdních pruhů a překážek na vozovce. To vše za použití pouze jedné kamery. Řešení je realizováno metodami založenými na barvě a gradientu obrazu. Metody byly použity jak jednoduché tak i s matematickým modelem. Výsledkem práce je souhrn často používaných metod a jejich otestování a vzájemné porovnání. V závěru práce je představena realizace vlastního programu.

## **KLÍČOVÁ SLOVA**

Inverzní perspektivní mapování, gradientní vektorové pole, statistické spojování regionů, mean-shift, detekce překážek, silnice, jízdní pruhy, lokální entropie, detekce hran, HSV, RGB, LAB, morfologické operace, prahování

## **ABSTRACT**

This master's thesis deals with depiction application of camera like sensitive element for assisting system of car. It was proposed, how find a road, a lane and a obstacle on roadways. Only one camera was application for it. Solution is realized by methods, that are based on color and gradient of image. It applies simple methods and methods with mathematical model. Result is sum of method and its test and comparing. Realization of my program is present at resume of this thesis.

## **KEYWORDS**

Inverse perspective mapping, Gradient vector flow, statistical region merging, mean-shift, interruptions detection, obstacle detection, road, traffic lanes, local entropy, edge detection, HSV, RGB, LAB, mathematical morphology, thresholding

DOJAVA, Marian *Detekce jízdních pruhů a překážek* : diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2011. 66 s. Vedoucí práce byl Ing. Karel Horák, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Detekce jízdnic pruhů a překážek“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno .....

.....

(podpis autora)

Děkuji tímto Ing. Karlovi Horákovi, Ph.D. a  
Ing. Iloně Kalové, Ph.D za cenné rady a  
vedení při vypracovávání této diplomové  
práce.

# OBSAH

Úvod	11
<b>1 Obraz</b>	<b>12</b>
1.1 Reprezentace obrazu	12
1.1.1 Barevný obraz	12
<b>2 Operace s obrazem</b>	<b>16</b>
2.1 Diskrétní konvoluce	16
2.2 Detekce hran	16
2.3 Matematická morfologie (základní operace)	17
2.3.1 Dilatace	18
2.3.2 Eroze	18
2.3.3 Otevření, uzavření	18
<b>3 Inverzní perspektivní mapování</b>	<b>20</b>
3.1 Geometrický model snímacího zařízení	20
3.2 Mapování $P(u, v)$ do $P(x, y, 0)$	22
<b>4 Gradientní vektorové pole</b>	<b>24</b>
4.1 Postup při výpočtu GVF	24
<b>5 Jednoduché filtry pro nalezení vozovky</b>	<b>26</b>
5.1 Filtr na základě detekce hran	26
5.2 Filtr na základě barev	27
5.3 Detekce krajnice z výsledků jednoduchých filtrů	28
<b>6 Hledání silnice pomocí hran a modelu krajnice</b>	<b>31</b>
6.1 Detekce krajnice pomocí hyperbolického modelu	31
6.1.1 Věrohodnost modelu	31
6.1.2 Optimalizace parametrů $s$	32
6.1.3 Výsledky metody	33
6.2 Model krajnice s využitím křivky B-splajn	34
6.2.1 Kubický b-splajn	34
6.2.2 B-splajn jako model jízdního pruhu	35
6.2.3 Adaptace parametrů modelu	36
6.2.4 Výsledky metody	39

<b>7</b>	<b>Detekce vozovky na základě analýzy barev</b>	<b>40</b>
7.1	Detekce vozovky pomocí metody mean-shift . . . . .	40
7.1.1	Princip metody mean-shift . . . . .	40
7.1.2	Shlukování blízkých bodů . . . . .	41
7.1.3	Realizace, výsledky . . . . .	42
7.2	Detekce vozovky na základě statistického spojování regionů . . . . .	43
7.2.1	Algoritmus . . . . .	44
7.2.2	Výsledky metody . . . . .	45
<b>8</b>	<b>Detekce překážek</b>	<b>46</b>
8.1	Možné metody pro hledání překážek . . . . .	46
8.2	Základní metody pro detekci překážek . . . . .	47
8.2.1	Metoda založená na vertikálních a horizontálních hranách . . .	47
8.2.2	Detekce překážek pomocí stínu . . . . .	49
8.2.3	Detekce překážek pomocí entropie . . . . .	52
8.3	Kombinace metod pro detekci překážek . . . . .	54
<b>9</b>	<b>Program pro detekci silnice a překážek</b>	<b>56</b>
<b>10</b>	<b>Závěr</b>	<b>58</b>
10.1	Porovnání metod . . . . .	58
10.2	Budoucí práce . . . . .	58
	<b>Literatura</b>	<b>60</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>62</b>
	<b>Seznam příloh</b>	<b>65</b>
<b>A</b>	<b>DVD s elektronickou verzí této práce, zdrojovými texty a testovacími soubory</b>	<b>66</b>



# SEZNAM OBRÁZKŮ

1.1	Geometrické znázornění RGB prostoru . . . . .	13
1.2	Geometrické zobrazení prostoru HSV a HSL . . . . .	14
1.3	Geometrický prostor CIE 1976 (Lab) . . . . .	15
2.1	Základní strukturní elementy . . . . .	17
3.1	Perspektivní zobrazení . . . . .	20
3.2	Geometrický model snímání, pohled $y, z$ . . . . .	21
3.3	Geometrický model snímání, pohled $x, z$ . . . . .	21
3.4	Ukázka inverzního perspektivního mapování . . . . .	23
4.1	Výsledek hranového detektoru . . . . .	24
4.2	Jednoduché vektorové pole . . . . .	25
4.3	Gradientní vektorové pole . . . . .	25
5.1	Ukázka filtru pro detekci vozovky pomocí hran . . . . .	26
5.2	Filtr s maskou o velikosti 11 . . . . .	27
5.3	Snímek v barevném prostoru HSV . . . . .	27
5.4	Snímek v barevném prostoru $L^*a^*b^*$ . . . . .	27
5.5	Ukázka filtru $S$ – složky . . . . .	28
5.6	Vstupní obraz pro detekci krajnic . . . . .	29
5.7	Nalezené a vyfiltrované krajnice . . . . .	29
5.8	Výsledný obraz s nalezenými krajnicemi . . . . .	30
6.1	Ukázky výsledků metody . . . . .	33
6.2	Závislost věrohodnostní funkce na počtu iterací . . . . .	34
6.3	Kubická b-splajn křivka . . . . .	35
6.4	Vliv působících externích sil na parametr $k$ . . . . .	38
6.5	Vliv působících externích sil na deformaci středového b-splajnu . . . . .	38
6.6	Výsledek modelu b-splajn . . . . .	39
7.1	Obraz jako 3-rozměrná data . . . . .	40
7.2	Ukázka posuvu bodů pomocí mean-shift . . . . .	41
7.3	Ukázka supergridu . . . . .	42
7.4	Ukázka výsledku metody mean-shift . . . . .	43
7.5	Ukázka obrazů . . . . .	44
7.6	Ukázka výsledků metody pro různé hodnoty $q$ . . . . .	45
8.1	Dvojice hypotéza-potvrzení hypotézy . . . . .	46
8.2	Základní dělení metod pro detekci překážek . . . . .	46
8.3	Histogram sum hran v řádcích . . . . .	47
8.4	Nalezená maxima . . . . .	48
8.5	Kaskada subobrazů . . . . .	49
8.6	Ukázka stínu pro různé hodnoty prahu . . . . .	50

8.7	Volná silnice . . . . .	51
8.8	Aproximace histogramu normálním rozložením pravděpodobnosti . .	51
8.9	Ukázka stínu pro různé hodnoty prahu . . . . .	52
8.10	Nalezená překážka pomocí stínu . . . . .	53
8.11	Výsledek po prahování výsledku lokální entropie . . . . .	53
8.12	Postup při hledání příznaků pomocí hran . . . . .	54
8.13	Možné oblasti překážek podle příznaku stínu . . . . .	55
8.14	Výsledky kombinované metody (červená 3, zelena 2 shody) . . . . .	55
9.1	Ukázka výsledků algoritmu . . . . .	57
10.1	Ukázky výsledků jednotlivých metod . . . . .	58

# ÚVOD

Úkolem této práce je použití kamery jako snímače pro asistenční systém vozidla. Dalším cílem je navrhnout část tohoto systému. Systém by měl být schopen určit sjízdnou vozovku, případně i samostatné jízdní pruhy. Součástí systému má být také detekce překážek.

Tyto systémy jsou již u mnoha automobilek používány, problém ovšem je, že jsou licenčně ošetřené, nelze tak zjistit jak pracují.

Tyto systémy ovšem používají nejenom kameru (náš případ), ale také další senzory např.: ultrazvukové radary pro měření vzdálenosti (detekce překážek), infra senzory (použití za zhoršených světelných podmínek) a podobně.

Naší prací tedy také ověříme, jak bude použití jen kamery komplikovat situaci.

# 1 OBRAZ

Obraz je matematicky popsán jako spojitá funkce  $O(u,v)$  dvou argumentů v rovině. Této funkci se říká obrazová funkce. Hodnotou obrazové funkce je většinou jas. Jas je veličina, která souhrnně vyjadřuje vlastnosti obrazového signálu způsobem, který odpovídá jeho vnímání člověkem. Někdy může obrazová funkce reprezentovat i jinou fyzikální veličinu (tlak, vzdálenost, teplotu atd.). [4]

## 1.1 Re prezentace obrazu

V této práci se zabýváme pouze diskrétními obrazy, tzn. obraz je reprezentován diskrétní obrazovou funkcí  $O(u, v)$ . Základní dělení reprezentace obrazu může být:

1. Podle prostoru
  - 3D-Odpovídá tomu, jak vnímáme obraz my, tj. trojrozměrně. Nese informaci, kde přesně se snímáný objekt nachází, a to v prostoru.
  - 2D-Jedná se o perspektivní zobrazení části 3D prostoru.
2. Podle jasové informace:
  - Černobílý obraz (binární)-Obrazová funkce nabývá pouze dvou hodnot
  - Šedotónový obraz-Obrazová funkce nabývá více hodnot, obvykle 0-255.
  - Barevný obraz-Obrazová funkce nese informaci nejen o jasu, ale také o barvě.

### 1.1.1 Barevný obraz [10]

Barevný obraz obsahuje jak informaci o jasu tak i informaci o barvě. Podle toho, jakým způsobem jsou tyto informace v obraze uloženy, rozlišujeme různé barevné modely.

#### RGB

RGB model je jeden ze základních, každý obrazový bod je určen vektorem o 3 položkách. V tomto vektoru jednotlivé položky reprezentují velikost jasu určité základní barvy (R – červená, G – zelená, B – modrá). Velikosti jasu jednotlivých barev jsou vyjádřeny celočíselným intervalem 0-255 (8bitů = 1byte), používají se i jiné rozsahy 16bitů 24bitů apod.

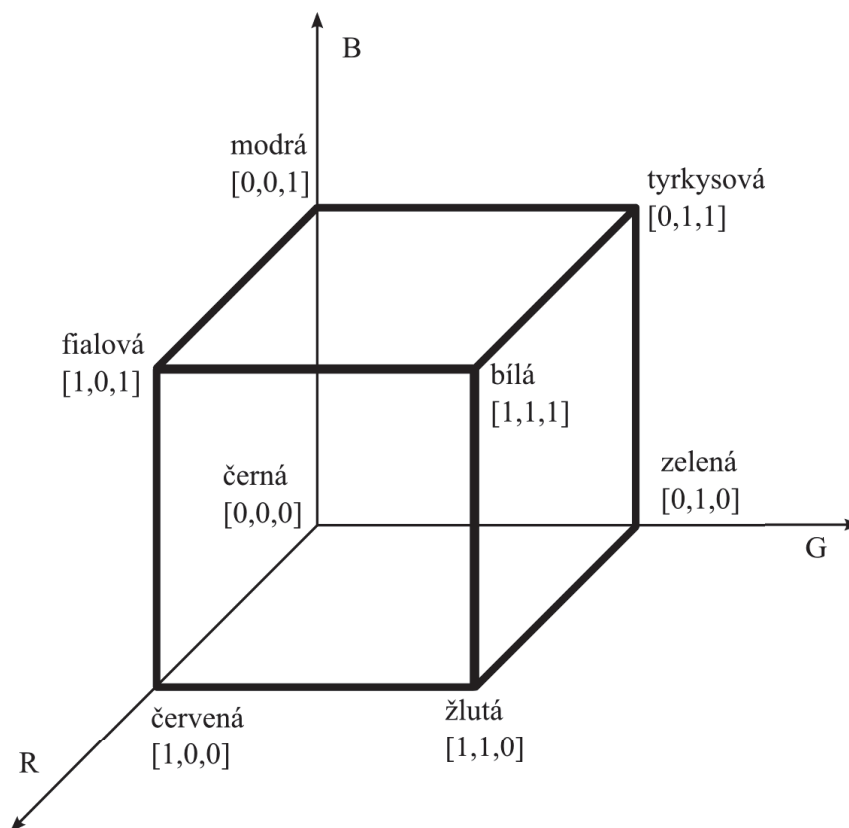
Počet barevných odstínů, které lze reprezentovat trojicí bytů je  $256^3 = 16777216$ .

Složíme-li červenou, zelenou a modrou složku v plné intenzitě získáme barvu bílou. Snižováním intenzit všech jednotlivých složek stejně, získáme stupně šedi. Této vlastnosti lze využít k převodu barevného obrazu na šedotónový (průměr).

Tento převod ovšem neodpovídá tomu, jak je lidské oko citlivé na jednotlivé barvy. Následující vztah koresponduje s vlastnostmi oka

$$I = 0,299R + 0,587G + 0,114B. \quad (1.1)$$

Barevný rozsah v RGB prostoru můžeme zobrazit jako jednotkovou krychli. Počátek krychle reprezentuje černou barvu, její vrchol pak bílou. Vrcholy krychle na osách představují základní barvy.



Obr. 1.1: Geometrické znázornění RGB prostoru

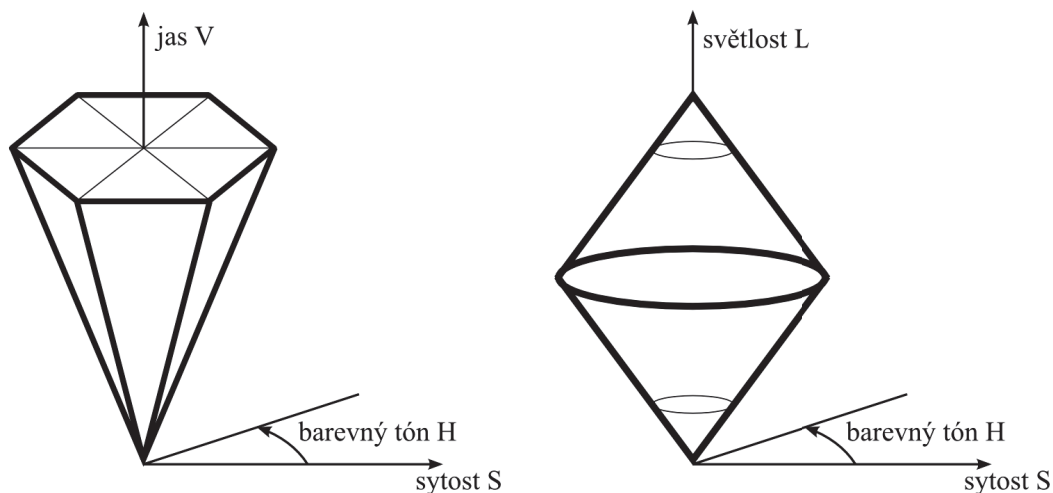
## HSV, HLS

HSV i HLS nerepresentují barvy v základních barvách. Parametry prostoru HSV jsou: barevný tón (H – hue), sytost (S – saturation) a jasová složka (V – value). Barevný tón udává převládající spektrální barvu, sytost určuje příměsí jiných barev a jas je dán množstvím bílého světla. Pro zobrazení HSV prostoru se používá šesti-boký jehlan s vrcholem v počátku souřadnic. Souřadnice S a V se mění v rozsahu  $\langle 0; 1 \rangle$ , souřadnice H představuje úhel  $\langle 0; 360 \rangle$ , vrchol reprezentuje černou barvu.

HSV model trpí některými nedostatky. Tyto nedostatky mohou ztěžovat přesné určení barvy. Jedním z nich je jehlanový tvar, kvůli kterému se bod při změně

H nepohybuje po kružnici (což by bylo přirozené). Další nedostatek je nesymetrie prostoru z hlediska jasu. Tyto nedostatky odstraňuje model HLS.

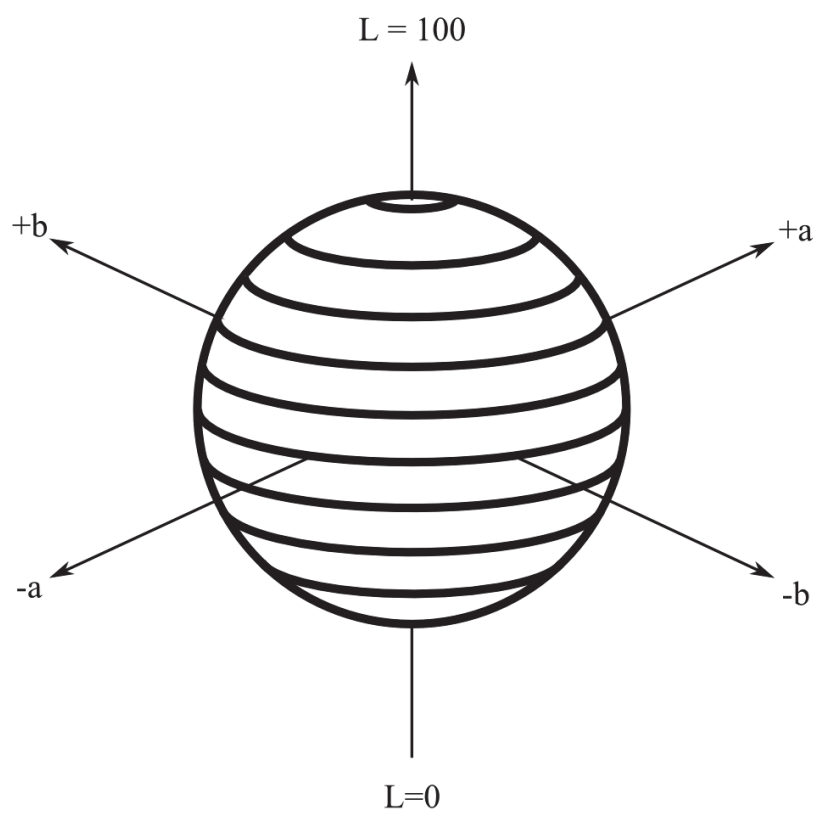
HLS model je obdobou HSV. Složka V je nahrazena složkou L (světlost, lightness). Jehlan je nahrazen dvojicí kuželů, dolní vrchol reprezentuje černou, horní pak bílou barvu. Nejjasnější barvy leží na podstavě kuželů,  $L = 0, 5$ .



Obr. 1.2: Geometrické zobrazení prostoru HSV a HSL

### CIE 1976 ( $L^*a^*b^*$ )

Prostor CIE 1976 slouží k popisu barev materiálu (povrchové barvy). Barva je určena souřadnicemi uvnitř kruhu. Osa  $L^*$  (světlost) o rozsahu  $\langle 0; 100 \rangle$  určuje velikost jasu (0 – černá, 100 – bílá), osy  $a^*$  a  $b^*$  reprezentují rozsahy barev: zelená ( $-a$ ), červená ( $+a$ ), modrá ( $-b$ ) a žlutá ( $+b$ ). Geometrický prostor modelu CIE 1976 je na 1.3.



Obr. 1.3: Geometrický prostor CIE 1976 (Lab)

## 2 OPERACE S OBRAZEM

V rámci vypracování zadání byly použity některé operace s obrazem. Jednalo se především o bodové jasové transformace a matematickou morfologii. Použité metody jsou popsány níže. Podrobnější informace o nich můžeme nalézt v [4].

### 2.1 Diskrétní konvoluce [10]

Konvoluce je základem mnoha operací s obrazem jako jsou např.: odstranění šumu, ostření, detekce hran apod. Konvoluce dvou funkcí je definována jako:

$$I(x) * h(h) = \int_{-\infty}^{\infty} I(x - \alpha)h(\alpha)d\alpha. \quad (2.1)$$

Funkce  $h(x)$  se nazývá konvoluční jádro. Jeho hodnoty určují způsob výpočtu výsledné hodnoty obrazového bodu. Diskrétní podoba konvoluce pro dvourozměrný signál (obraz) je v (2.2). Konvoluční jádro lze popsat jako tabulku o rozměrech  $\langle -k, k \rangle \langle -k, k \rangle$ .

$$I(x) * h(h) = \sum_{i=-k}^k \sum_{j=-k}^k O(u - i, v - j).h(i, j) \quad (2.2)$$

### 2.2 Detekce hran

Hrana je vektorová veličina určená velikostí a jejím směrem. Při jejím hledání se vychází z gradientu obrazové funkce na vybraném obrazovém elementu.

Hranové operátory vycházejí z parciálních derivací obrazové funkce, její gradient je pak v (2.3), směr je určen podle (2.4).

$$|\nabla g(x, y)| = \sqrt{\frac{\partial g^2}{\partial x} + \frac{\partial g^2}{\partial y}} \quad (2.3)$$

$$\psi = \tan^{-1} \left( \frac{\frac{\partial g^2}{\partial y}}{\frac{\partial g^2}{\partial x}} \right), \text{ pro } \frac{\partial g^2}{\partial x} \neq 0. \quad (2.4)$$

Pokud nás zajímá jen velikost nikoli směr, používá se často Laplaceův operátor  $\nabla^2$ . Laplaceův operátor je určen pomocí:

$$\nabla^2 = \frac{\partial^2 g^2}{\partial x} + \frac{\partial^2 g^2}{\partial y} \quad (2.5)$$

Pro diskrétní obrazy jsou derivace nahrazeny diferencemi.

Hledání hran probíhá pomocí diskrétní konvoluce s vhodným konvolučním jádrem. Konvoluční jádra (hranové operátory) lze dělit do dvou skupin. První skupina



realizuje derivaci aproximovanou pomocí diference. U směrových jader lze jejich otočením získat jiná jádra, reagující na jiný směr hrany. Druhá skupina hledá hranu jako průchod nulou druhé derivace. V (2.6 – 2.8) jsou ukázky některých konvolučních jader pro výpočet hran.

Robertsův operátor

$$h_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, h_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \quad (2.6)$$

Prewittův operátor

$$h_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, h_2 = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}. \quad (2.7)$$

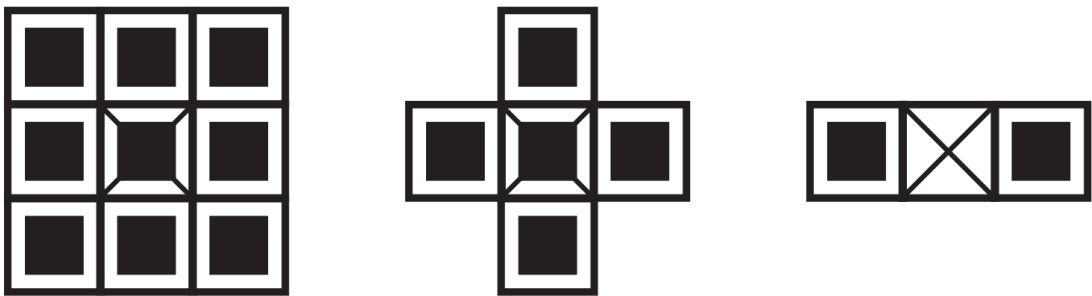
Sobelův operátor

$$h_1 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, h_2 = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}. \quad (2.8)$$

## 2.3 Matematická morfologie (základní operace)

Matematická morfologie vychází z vlastností bodových množin. Je jí možné použít jak pro obrázky ve stupních šedi tak pro binární.

Pro popis je použito jen binárních obrazů. Množina  $X$  reprezentuje objekt a  $X^C$  pozadí. Za pozadí jsou považovány i díry v objektu. Morfologické transformace jsou realizovány jako relace obrazu s jinou menší bodovou množinou  $B$ , této množině se říká strukturní element. Základní strukturní elementy jsou na obrázku 2.1.



Obr. 2.1: Základní strukturní elementy

Body příslušející objektům jsou vyznačeny čtverečky, počátek je značen přeškrtnutím.

### 2.3.1 Dilatace

Dilatace skládá body množin pomocí vektorového součtu.

$$X \oplus B = \{\mathbf{d} = \mathbf{x} + \mathbf{b}, \mathbf{x} \in X, \mathbf{b} \in B\} \quad (2.9)$$

Dilatace způsobuje rozšíření původního obrazu v závislosti na strukturním elementu. Dilatace se používá k zaplnění malých děr a úzkých zálivů, rozšíření úzkých čar apod.

### 2.3.2 Eroze

Eroze skládá množiny jako vektorový rozdíl, je duální operací k dilataci (není inverzní). Je definována jako

$$X \ominus B = \{\mathbf{d} + \mathbf{b} \in X, \text{pro} \forall \mathbf{b} \in B\} \quad (2.10)$$

Eroze zvětšuje díry v objektu. Zároveň ho zmenšuje v závislosti na strukturním elementu. Používá se pro zjednodušení struktury objektů. Způsobuje také rozdělení objektů na více částí, jsou-li spojeny čarou tloušťky srovnatelné s elementem.

Pomocí eroze lze nalézt obrys objektu a to např. tím, že od původního obrazu odečte jeho eroze.

Eroze na rozdíl od dilatace není komutativní!

### 2.3.3 Otevření, uzavření

Eroze a dilatace nejsou navzájem inverzní, jejich kombinací dostaneme další morfologické operace.

Operace otevření je definována jako eroze následovaná dilatací:

$$X \circ B = (X \ominus B) \oplus B. \quad (2.11)$$

Uzavření je definováno jako eroze dilatovaného obrázku:

$$X \bullet B = (X \oplus B) \ominus B. \quad (2.12)$$

Otevření rozdělí objekty spojené úzkým přechodem podobně jako eroze, ale na rozdíl od ní nemění jejich tvar.

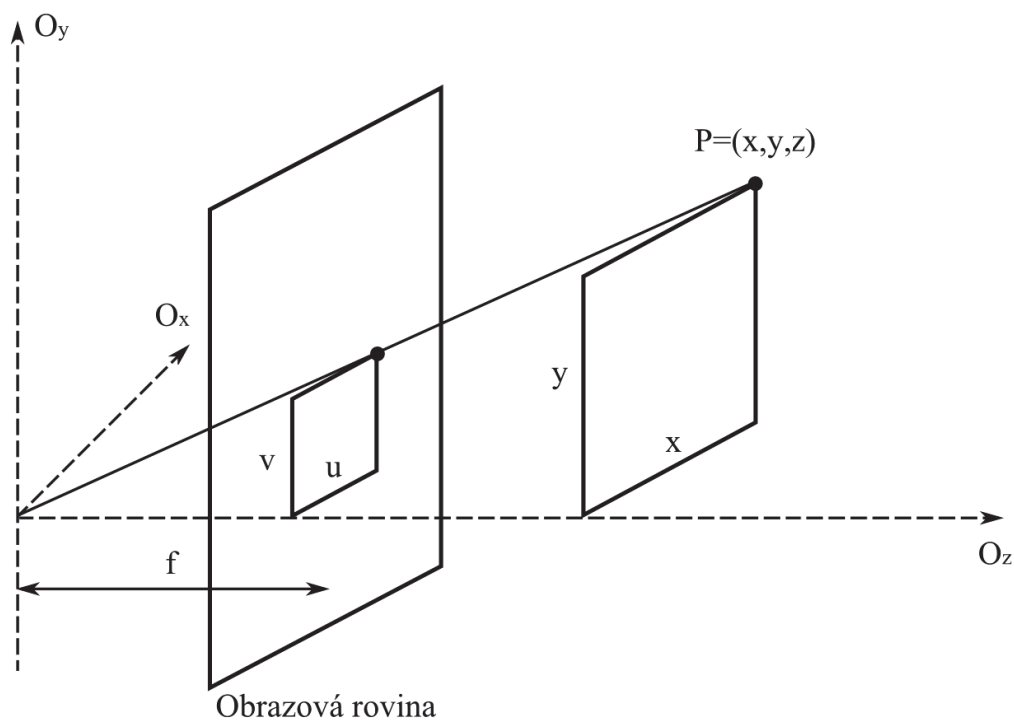
Uzavření spojí objekty oddělené úzkou mezerou, zaplní malé díry a vyhladí obraz zaplněním malých nerovností. Malý, úzký jsou relativní vzhledem k velikosti strukturního elementu.

Uzavření a otevření mají důležitou vlastnost: idempotentnost. Tato vlastnost znamená neměnnost objektu po opětovném provádění operace. Díky této vlastnosti

mohou být množiny  $A$  otevřené nebo uzavřené vzhledem k  $B$ . Těto vlastnosti lze využít pro otevření nebo uzavření oblastí s libovolnou velikostí (operace se opakuje až do stavu idempotentnosti).

### 3 INVERZNÍ PERSPEKTIVNÍ MAPOVÁNÍ

Obraz snímáný kamerou je perspektivním zobrazením scény. Na obrázku 3.1 je grafické znázornění perspektivního zobrazení kamerou. Rovnice (3.1) pak udávají vztahy mezi jednotlivými veličinami.



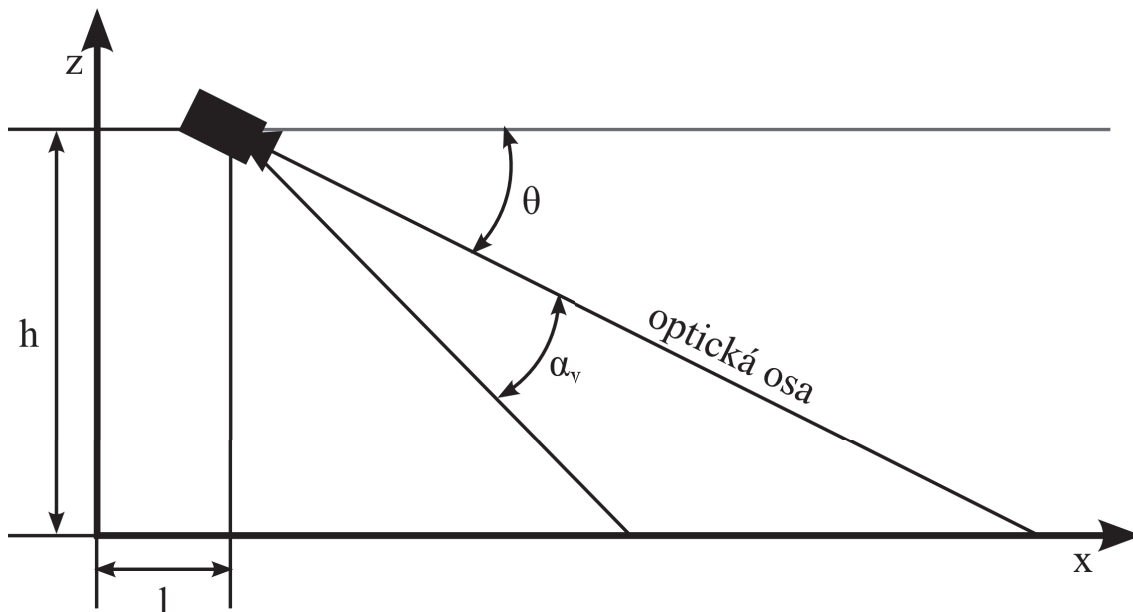
Obr. 3.1: Perspektivní zobrazení [4]

$$u = \frac{xf}{z}, v = \frac{yf}{z} \quad (3.1)$$

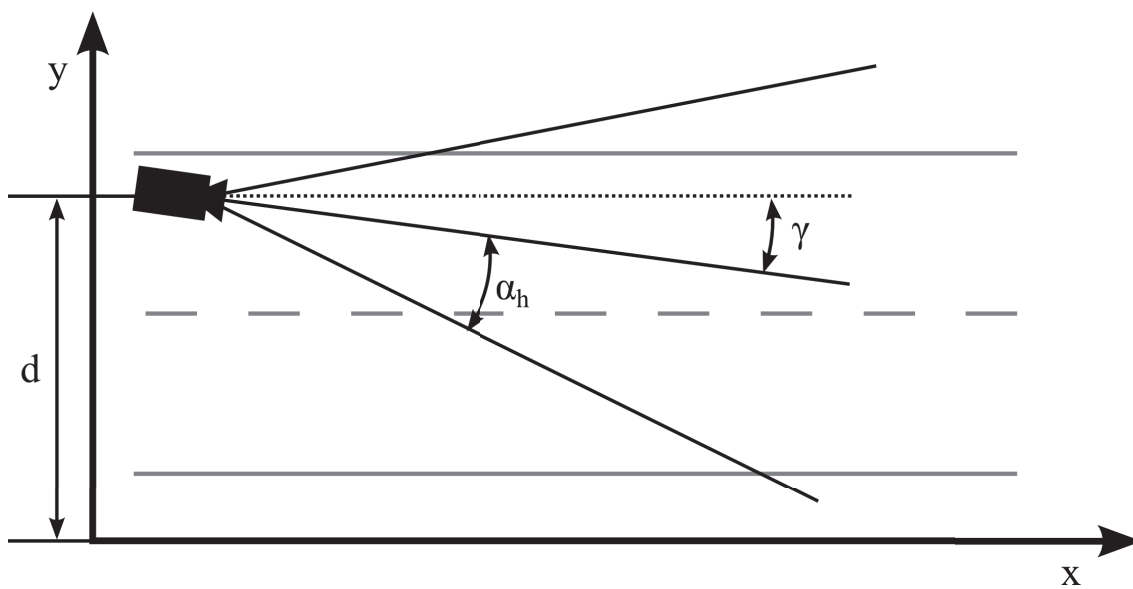
Pomocí geometrického modelu snímacího aparátu a skutečnosti, že kamera snímá perspektivně, lze nalézt transformaci, která převede obraz ze souřadnic kamery na souřadnice  $x$  a  $z$  snímané scény. Tato transformace se nazývá inverzní perspektivní mapování.

#### 3.1 Geometrický model snímacího zařízení

Pomocí geometrického modelu podle obrázků 3.2 a 3.3 můžeme odvodit rovnice pro přepočítání pozice každého bodu obrazu  $P(u, v)$  na polohu bodu v souřadnicích reálného světa, konkrétně  $P(x, y, 0)$ . Převádíme tedy perspektivní pohled v rovině kamery na pohled zhora a bez perspektivy (letadlový snímek).



Obr. 3.2: Geometrický model snímání, pohled  $y, z$



Obr. 3.3: Geometrický model snímání, pohled  $x, z$

V takto vzniklém snímku je možné hledat krajnice jízdních pruhů jako rovnoběžky. Také je možné použít tento pohled pro odhad vzdálenosti a pro nalezení překážek na vozovce.

## 3.2 Mapování $P(u, v)$ do $P(x, y, 0)$

Podle obrázku 3.2 a 3.3 odvodíme následující rovnice:

$$x(u, v) = \frac{h}{\tan((\theta + \alpha_h) + u \frac{2\alpha_h}{m-1})} \cos((\gamma - \alpha_v) + v \frac{2\alpha_v}{n-1}) + l \quad (3.2)$$

$$y(u, v) = \frac{h}{\tan((\theta + \alpha_h) + u \frac{2\alpha_h}{m-1})} \sin((\gamma - \alpha_v) + v \frac{2\alpha_v}{n-1}) + d \quad (3.3)$$

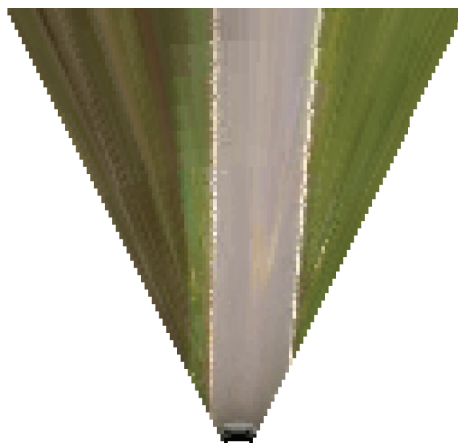
$$z = 0 \quad (3.4)$$

Je dán obraz  $O(u, v)$  o rozměrech  $m, n$ , jeho promítnutí do roviny  $(x, y, 0)$  provedeme podle rovnic (3.2–3.4). Úhel  $\theta$  udává velikost úhlu mezi optickou osou kamery a osou  $x$ . Zorný úhel kamery je reprezentován pomocí  $\alpha$ , rozeznáváme dva zorné úhly, horizontální ( $\alpha_h$ ) a vertikální ( $\alpha_v$ ). Parametry  $h, l, d$  udávají pozici kamery v prostoru  $K(h, l, d)$ ,  $\gamma$  udává úhel osy kamery proti ose  $y$ .

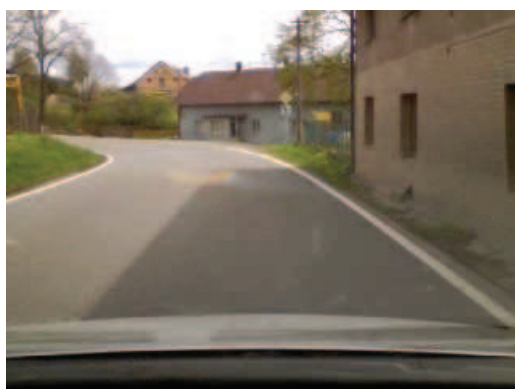
Ukázka inverzního perspektivního mapování je na obrázku 3.4.



(a) originál



(b) inverzní perspektivní transformace



(c) originál



(d) inverzní perspektivní transformace

Obr. 3.4: Ukázka inverzního perspektivního mapování

## 4 GRADIENTNÍ VEKTOROVÉ POLE [16]

Gradientní vektorové pole (GVF) je vektorové pole  $\vec{v}(x, y) = (u(x, y), v(x, y))$ , které minimalizuje energetickou funkci:

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2 dx dy \quad (4.1)$$

Funkce  $f(x, y)$  je gradient obrazu (v našem případě silnice),  $\mu$  je parametr udávající míru vlivu integrální složky a složky parciální derivace vektorového pole.

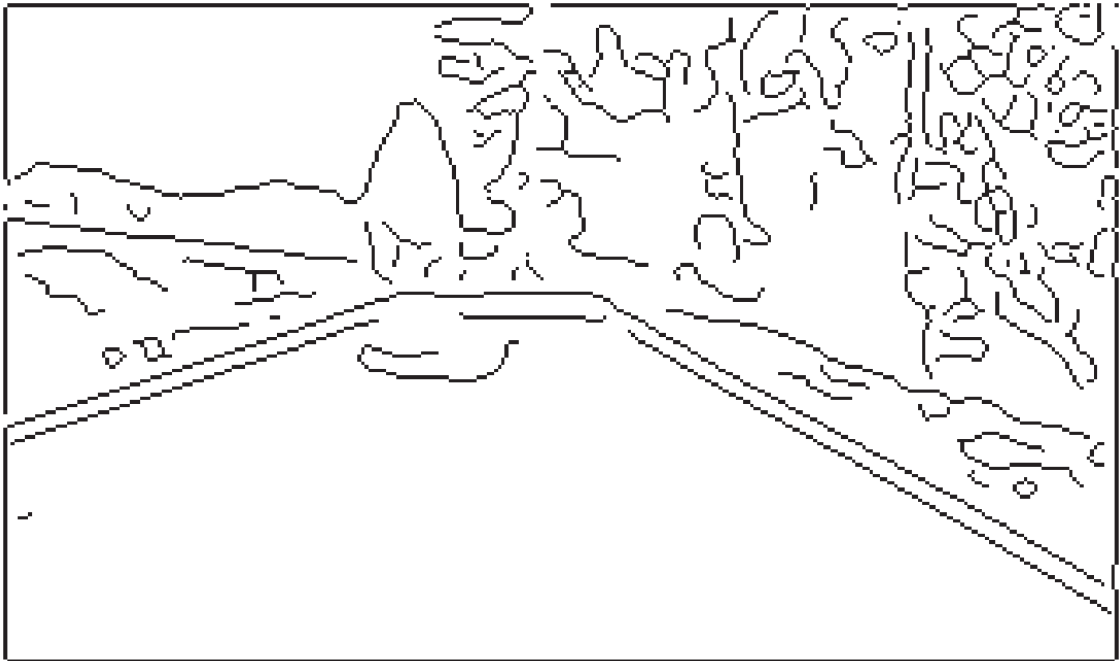
Výpočet GVF můžeme provést pomocí Eulerovi rovnice:

$$\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \quad (4.2)$$

$$\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \quad (4.3)$$

### 4.1 Postup při výpočtu GVF

Nejprve vypočítáme gradient obrazu (libovolný hranový operátor) obrázek 4.1.

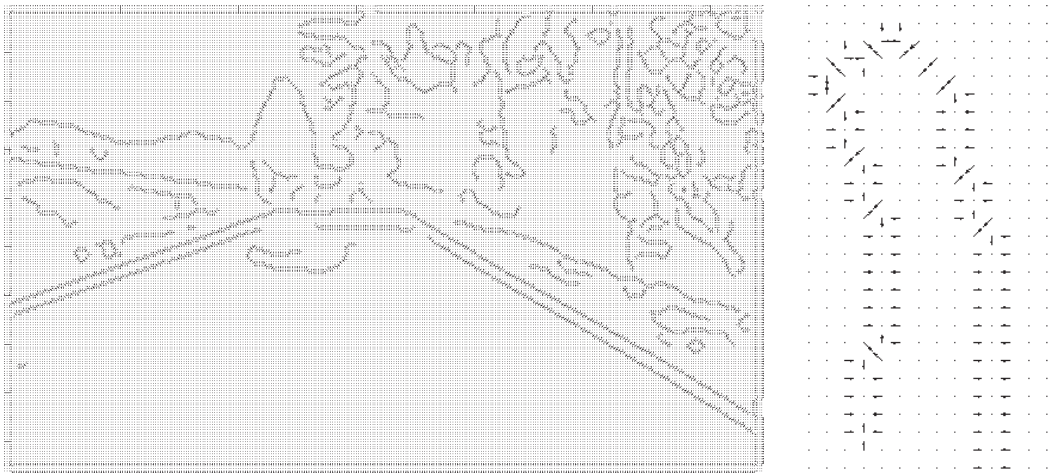


Obr. 4.1: Výsledek hranového detektoru

Z takto získaných hran se vypočítá další gradient, tentokrát se však spočte zvlášť pro směr v ose  $u$  a  $v$ . Dostaneme tak  $f_u(x, y)$  a  $f_v(x, y)$ , tyto funkce dohromady tvoří jednoduché vektorové pole (obrázek 4.2).

Z tohoto pole pak již pomocí rovnic (4.2, 4.3) vypočítáme GVF obrázek 4.3.

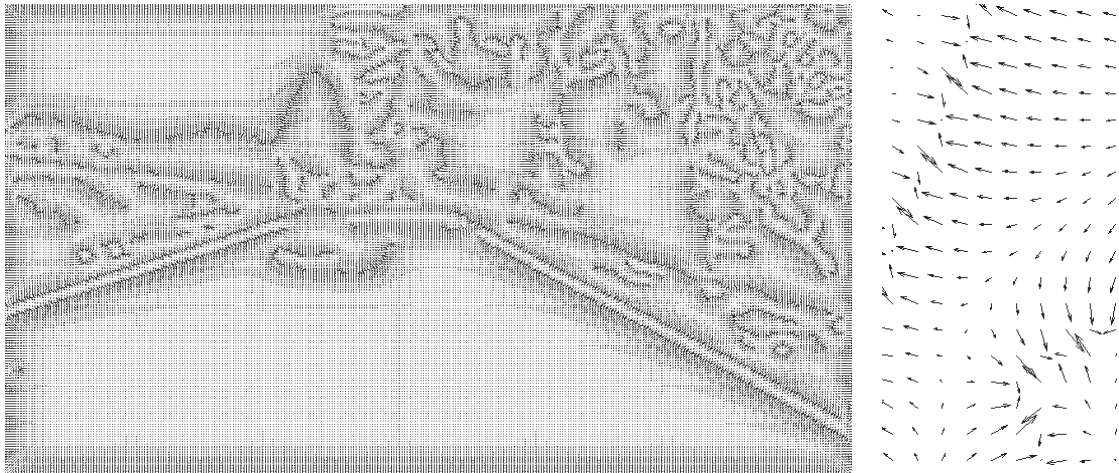




(a) úplné vektorové pole

(b) detail

Obr. 4.2: Jednoduché vektorové pole



(a) úplné vektorové pole

(b) detail

Obr. 4.3: Gradientní vektorové pole

GVF má proti jednoduchému vektorovému poli velkou výhodu v tom, že je toto pole podstatně širší. To umožňuje zachytit směr pole v mnohem větší vzdálenosti od původní hrany.

## 5 JEDNODUCHÉ FILTRY PRO NALEZENÍ VOZOVKY

### 5.1 Filtr na základě detekce hran

Tento filtr pracuje na základě detekce hran v obraze, hrany je možné nalézt pomocí libovolného hranového detektoru (v našem případě se jedná o Prewittovy detektory). Vychází z úvahy, že asfalt na rozdíl od okolí (tráva, pole, dlažba, stromy apod.) je kompaktní, tzn. obsahuje mnohem méně hran.

Obraz, který představuje výsledek hranového detektoru, se dále upravuje. Úprava spočívá v jeho procházení zvoleným elementem o určené velikosti (čtvercový element o velikosti 8 pixelů). Je-li v oblasti, kterou překrývá tento element nalezena hrana, je tento element prohlášen za neúčastníci se silnice. Pokud se ovšem v elementu žádná hrana nenachází, je považován za součást silnice.

Na takto získaný obraz se následně aplikují morfologické operace. Mezi ně patří především eroze následovaná odstraněním příliš malých (nevýznamných oblastí). Poté je obraz dilatován, aby se nezměnily rozměry oproti vstupnímu obrazu a výsledná oblast se částečně zjednodušila. Nakonec jsou vyplněny všechny díry. Ukázka výsledku je na obrázku 5.1



Obr. 5.1: Ukázka filtru pro detekci vozovky pomocí hran: (a) výsledek filtru; (b) originál.

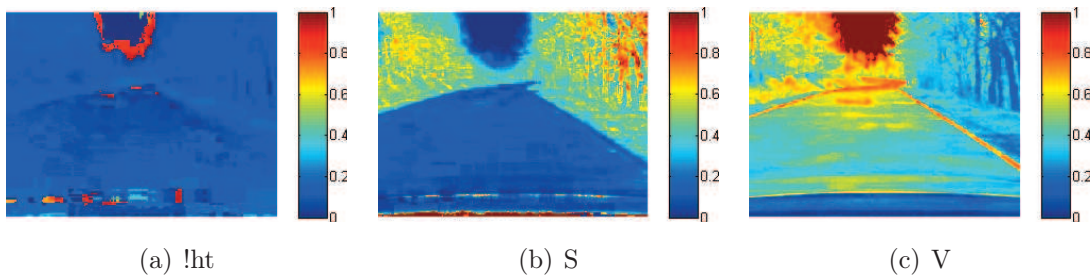
Jak je vidět z obrátek 5.1, je tento filtr náchylný na různé stíny, chyby na vozovce apod. Kvůli těmto chybám je výsledný objekt rozdělen na části a často není jednoduché určit, které objekty jsou součástí vozovky a které ne. Lepších výsledků je dosaženo pomocí hranového detektoru s větším jádrem (použito Prewittovo jádro s velikostí 11). Výsledek je na obrázku 5.2.



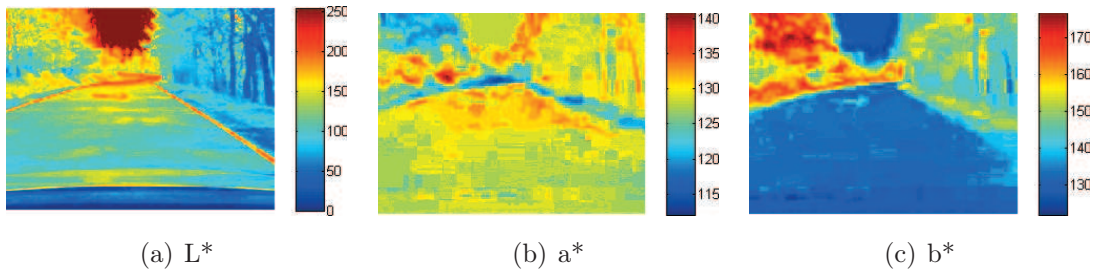
Obr. 5.2: Filtr s maskou o velikosti 11

## 5.2 Filtr na základě barev

Další možnost jak najít v obraze vozovku jsou barvy. Pro filtr byly použity různé barevné modely, barevný prostor se složkami: červená, zelená, modrá – red, green, blue (RGB) byl vyloučen hned v počátku, neboť jeho jednotlivé barevné složky jsou silně závislé na jasů. Pro filtr byly zvoleny modely, u kterých barva na jasů nezávisí. Vybrané modely jsou barevný prostor se složkami: L – světelnost, a – osa zelená - červená, b – osa modrá - žlutá ( $L^*a^*b^*$ ) a barevný prostor se složkami: H – barevný tón, S – sytost barvy, V – hodnota jasu (HSV).



Obr. 5.3: Snímek v barevném prostoru HSV



Obr. 5.4: Snímek v barevném prostoru  $L^*a^*b^*$

Jak je patrné z obrázků 5.3 a 5.4, nejvhodnější pro detekci silnice je model HSV a to ve složce S – saturace. V tomto zobrazení je nejvíce patrná hranice mezi vozovkou a okolím. Další dobře odlišitelné hranice jsou v modelu  $L^*a^*b^*$  a to ve složce b.

Jednotlivé barevné filtry (HSV i  $L^*a^*b^*$ ) pracují na principu prahování. Jako práh je volen průměr ze vzorku. Tento vzorek je brán z obrazu v určené barevné složce. Vzorek je brán z oblasti s největší pravděpodobností výskytu silnice (předpokládá se, že jedeme po silnici). Tato oblast se nachází na ose obrazu cca v jedné třetině na osy y (kamera je nainstalována na ose vozidla). Lepších výsledků lze dosáhnout průměrováním z více po sobě jdoucích snímků.

Z barevných filtrů dává nejlepší výsledky filtr HSV, konkrétně filtr složky S, její ukázka je na obrázku 5.5.



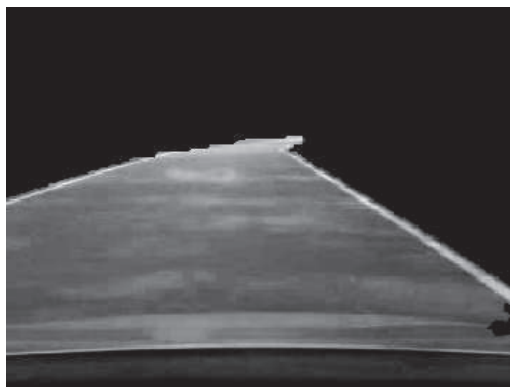
Obr. 5.5: Ukázka filru S – složky

### 5.3 Detekce krajnice z výsledků jednoduchých filtrů

Vstupem detektoru krajnice je obraz vzniklý kombinací obrazu z kamery, převedeným na stupně šedi a následnou kombinací s obrazem z detektoru silnice. V obrazu silnice se vynechají oblasti, které jsou příliš vysoko na to, aby mohli být skutečně silnicí. Ukázka je na obrázku 5.6.

Pro hledání krajnice je vstupní obraz naprahován s prahem 70 % maxima. Tím získáme pouze oblasti s bílou barvou (krajnice). Následuje popsání takto nalezených oblastí hraničními body. Popisování probíhá od středu ke stranám, tímto postupem získáme vždy nejbližší krajnici (hledáme hranice aktuálního jízdního pruhu).

Získaná data je třeba vyfiltrovat. V datech se nalézají značné chyby (špatně naprahování), krajnice také nejsou lineární (velká chyba při následné aproximaci).



Obr. 5.6: Vstupní obraz pro detekci krajnic

Filtrace je založena na počítání úhlu nalezených bodů k x-ové souřadnici (jednoduchá Pythagorova věta). Body jsou použité jen tehdy, je-li spočtený úhel v rozmezí  $\pm 5\%$  mediánu ze všech nalezených souřadnic.

Selže-li hledání krajnic podle naprahovaného obrazu, přejde se od naprahování k detekci pomocí hran. Získávání souřadnic krajnic je stejné jako v předchozím případě.

Ukázka nalezených a vyfiltrovaných dat o krajnicích je na obrázku 5.7.



Obr. 5.7: Nalezené a vyfiltrované krajnice

Na závěr jsou získaná data o krajnicích aproximována přímkami a jsou spočteny jejich rovnice.



Obr. 5.8: Výsledný obraz s nalezenými krajnicemi

## 6 HLEDÁNÍ SILNICE POMOCÍ HRAN A MO- DELU KRAJNICE

Největším problémem detekce krajnice pomocí hran je velké množství hran nepatřících krajnicím, jde především o stíny, vady vozovky a podobně. Tento problém se snaží odstranit nebo alespoň potlačit model krajnice.

Model krajnice vychází z několika skutečností jako: počet krajnic, krajnice jsou rovnoběžné, zkoumaný obraz je perspektivní. Pomocí nich lze určit několik vlastností modelu:

1. model má nejméně dvě krajnice (levá, pravá – L,R)
2. jednotlivé krajnice mají jeden společný počátek takzvaný vanishing point, tento bod leží na přímce horizontu
3. model by měl umět obsáhnout i zatáčky nejen přímou silnici

### 6.1 Detekce krajnice pomocí hyperbolického modelu

Tento model je přezván z [6], vychází z rovnice hyperboly. Pro naše účely je tato rovnice ve tvaru:

$$c = \frac{k}{(r - hz)} + b(r - hz) + vp \quad (6.1)$$

#### 6.1.1 Věrohodnost modelu

Definujme funkci  $f(\alpha, x)$  jako:

$$f(\alpha, x) = \frac{1}{(1 - \alpha x^2)} \quad (6.2)$$

Jako model krajnic nám poslouží matematický popis hyperboly. Model pak bude popsán rovnicemi:

$$E_L(s) = \frac{s_1}{(r - hz)} + s_2(r - hz) + s_3 \quad (6.3)$$

$$E_R(s) = \frac{s_1}{(r - hz)} + s_4(r - hz) + s_3 \quad (6.4)$$

Primární věrohodnostní funkce pak bude:

$$P(s) = \frac{\pi}{2} \arctan(\alpha_{width}(s_4 - s_2)) \quad (6.5)$$

Tato pravděpodobnost plní dva úkoly. První je podmínka, aby levá krajnice byla vlevo a pravá vpravo, není-li tomu tak, je tato pravděpodobnost záporná. Druhý úkol odráží skutečnost, že jízdní pruh má určitou minimální šířku. Tuto minimální šířku ovládá parametr  $\alpha_{width}$ .

Rovnice (6.3, 6.4) jsou rovnicemi jednotlivých krajnic (v tomto modelu jsou dvě), tyto rovnice společně tvoří model krajnice. Tento model je určen pomocí parametru  $s$ , kde  $s = \{s_1, s_2, s_3, s_4\}$ , obě rovnice mají společný parametr  $vp$  ( $s_3$ ), tím je dosaženo podmínky na společný počátek (vanishing point).

Z rovnic (6.2 – 6.5) můžeme sestavit celkovou věrohodnostní (hodnotící) funkci modelu. Tato věrohodnost je dána vztahem:

$$\begin{aligned}
L(s) = & P(s) \cdot \sum_{r,c} \left( gm(r, c) \cdot f(\alpha_{prox}, c - E_L(s)) \cdot \right. \\
& \cdot f(\alpha_{orient}, \cos( gd(r, c) - \arctan( \frac{d}{dr} E_L(s) ) ) ) + \\
& + gm(r, c) \cdot f(\alpha_{prox}, c - E_R(s)) \cdot \\
& \left. \cdot f(\alpha_{orient}, \cos( gd(r, c) - \arctan( \frac{d}{dr} E_R(s) ) ) ) \right)
\end{aligned} \tag{6.6}$$

Tento věrohodnostní model počítá sumu všech bodů obrazu hran  $gm(r, c)$  takových, které leží na křivkách  $EL(s)$  a  $ER(s)$ . Model počítá kromě existence hrany také s jejím směrem  $gd(r, c)$ . Parametry  $\alpha_{prox}$  a  $\alpha_{orient}$  jsou konstanty určující požadovanou míru schody mezi modelem a hranovým obrazem, případně orientací hrany. Parametry  $r, c$  představují aktuální řádek a sloupec obrazu, s kterým se právě provádí výpočet.

### 6.1.2 Optimalizace parametrů $s$

Pro optimalizaci parametrů modelu krajnic je možné požit libovolnou metodu pro globální optimalizaci, hledání globálního optima. Testovány byli genetický algoritmus a algoritmus simulovaného žíhání. Genetický algoritmus nedával valné výsledky, to je dáno především velkým rozsahem možných hodnot parametrů. Naproti tomu simulované žíhání je mnohem vhodnější.

#### Simulované žíhání

Simulované žíhání je podrobně popsáno v [1] nebo [12]. Princip algoritmu je následující:

1.  $L(s_N)$  je věrohodnost modelu podle (6.6) s parametry  $s_N$  v  $N$ -té iteraci
2. Iterace  $N+1$  :
  - Výběr náhodného parametru  $s$  z okolí  $s_N$ ,

$$s = s_N + \Delta \tag{6.7}$$

- výpočet teploty  $T(n)$

$$T = T_{init} \left( \frac{T_{final}}{T_{init}} \right)^{\left( \frac{(N+1)}{maxiter} \right)}, \tag{6.8}$$



kde  $T_{init}$  je počáteční teplota,  $T_{final}$  koncová teplota a  $maxiter$  je maximální počet iterací

- výpočet rozhodovací funkce  $\rho$

$$\rho_{N+1} = e^{\left(\frac{L(s_N+\Delta)-L(s_N)}{T(N+1)}\right)} \quad (6.9)$$

- výběr  $s_{N+1}$  pomocí pravidel:

$$s_{N+1} = \begin{cases} s_N + \Delta & \text{pokud } \rho(N+1) \geq 0, \\ s_N + \Delta & \text{s pravděpodobností } \rho(N+1) \text{ pokud } \rho(N+1) \leq 0, \\ s_N & \text{jinak.} \end{cases} \quad (6.10)$$

1. Pokračovat krokem 2 do dosažení počtu iterací

Pomocí tohoto algoritmu hledáme maximální hodnotu funkce  $L(s)$ . V ukázkách je počítáno s  $-L(s)$ , to je proto že program MATLAB tento algoritmus obsahuje, hledá však minimum.

### 6.1.3 Výsledky metody

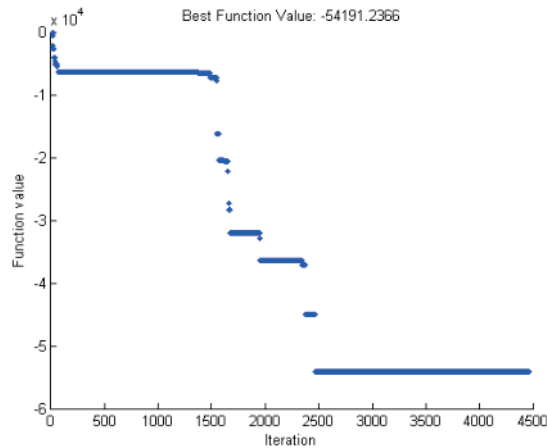
Na následujících obrázku je ukázka některých výsledků.



Obr. 6.1: Ukázky výsledků metody

Z obrázku 6.1 je vidět jak tato metoda aproximovala krajnici, je vidět že se skutečné blíží, není však totožná. Na obrázku 6.2 je zobrazen vývoj průběhu hodnotící pravděpodobnostní funkce  $L(s)$  v závislosti na počtu iterací algoritmu simulovaného žíhání.

Výsledky tohoto algoritmu jsou nedeterministické, to je dáno náhodným volením zkoumaných parametrů modelu  $s$ . Při každém spuštění algoritmu dostaneme různé výsledky, někdy ani není nalezené řešení podobné reálné podobě krajnic. Tento model aproximuje jen funkcí hyperboly, není tak možné obsáhnout všechny možné typy



Obr. 6.2: Závislost věrohodnostní funkce na počtu iterací

zatáček oblouků atd., jen jejich malé množství. Další nevýhoda spočívá ve značně velké oblasti možných parametru modelu a s tím spojená potřebná časová náročnost (algoritmus simulovaného žíhání toto částečně odstraňuje, ovšem za cenu nedeterministického chování).

Pro vylepšení výsledků a zrychlení metody by bylo vhodné zavést přidané informace. Užitečná by byla např. informace o vzdálenosti a směru nejbližší hrany. Tím by algoritmus hledání nemusel volit parametry zcela náhodně, ale s jistou předvídatelostí. Tyto a podobné přidané informace využívá následující model.

## 6.2 Model krajnice s využitím křivky B-splajn

Tento model se snaží nalézt krajnice pomocí křivky typu kubický b-splajn. Tento model potřebuje počáteční inicializaci, ta v této práci není samostatně realizována algoritmicky (je zadávána ručně). Existují však algoritmy, které toto dokáží, např. algoritmus CHEVP popsáný v [15], z této literatury také pochází tento model.

### 6.2.1 Kubický b-splajn

Splajn je křivka, která se skládá z polynomů a zaručuje spojitost derivací v navazovacích bodech těchto polynomů.

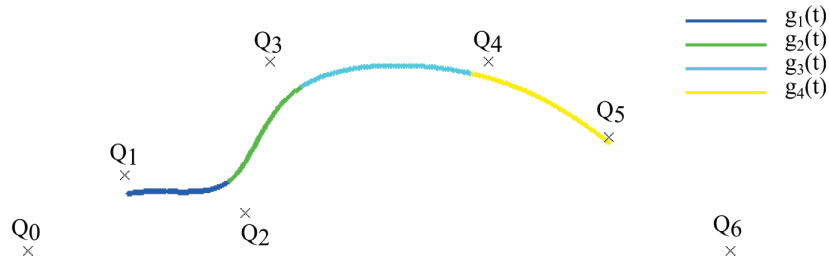
Kubický b-splajn je dán  $n+1$  kontrolními body  $\{Q_0, Q_1, \dots, Q_n\}$ , skládá se z  $n-2$  spojitých na sebe navazujících segmentů  $g_i(t)$ . Každý segment je dán lineární kombinací čtyř kubických polynomů, kde  $t$  je z rozsahu  $t \in \langle 0, 1 \rangle$ . V maticové formě jsou

tyto vztahy vyjádřeny jako:

$$g_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \quad (6.11)$$

$$g_i(t) = M_R(t) \cdot \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \quad (6.12)$$

$i = 1, 2 \dots n - 2, t \in \langle 0; 1 \rangle$ .



Obr. 6.3: Kubická b-splajn křivka

## 6.2.2 B-splajn jako model jízdního pruhu

Model se snaží aproximovat jízdní pruh jednou b-splajn křivkou (střed jízdního pruhu). Krajnice jsou hledány v závislosti na této středové křivce s přihlédnutím k faktu, že krajnice jsou ve většině případů rovnoběžné.

### Definice rovnoběžnosti

Pravá krajnice je vlastně jen posunutá levá krajnice, můžeme tedy určit jejich vzájemnou vzdálenost

$$D = y_R - y_L, \quad (6.13)$$

kde  $y_R$  je horizontální pozice pravého a  $y_L$  levého bodu ležícího na rovnoběžných krajnicích. Tyto dva body  $(P_L(y_L, x), P_R(y_R, x))$  ze souřadnic  $(x, y, z)$  převedeme pomocí perspektivní transformace na souřadnice obrazu  $(u, v)$ , respektive  $(r, c)$  (řádek, sloupec).

Tato dvojice bodů v souřadnicích obrazu mají také svou vzdálenost

$$d = c_R - c_L, \quad (6.14)$$

tato vzdálenost již ale není stejná, je závislé na souřadnici  $r$ . Vzdálenost je dána vztahem:

$$d = k(r - hz), \quad (6.15)$$

kde

$$k = \frac{f^2 D}{h(f^2 + hz^2)} \quad (6.16)$$

### Model

Definujme středovou křivku jako  $E_M(r_M, c_M)$ , levá krajnice pak bude  $E_L(r_L, c_L)$  a pravá  $E_R(r_R, c_R)$ , kde:

$$c_L = c_M - \frac{1}{2}d = c_M - \frac{1}{2}k(r_M - hz) \quad (6.17)$$

$$c_R = c_M + \frac{1}{2}d = c_M + \frac{1}{2}k(r_M - hz) \quad (6.18)$$

Středová křivka je námi definovaný b-splajn.

$$E_M(r_M, c_M) = M_R(t) \cdot \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \quad (6.19)$$

$$i = 1, 2 \dots n - 2, t \in \langle 0, 1 \rangle$$

Středový b-splajn může být deformován pomocí externí síly  $F_{M\_sum}$ , tato síla je suma sil působících na levou stranu  $F_L$  a pravou  $F_R$ .

$$F_{M\_sum} = F_L + F_R \quad (6.20)$$

Na parametr  $k$  působí síla  $F_{M\_diff}^c$  ( $c$  značí, že jde o velikost vektoru sil ve směru řádku) :

$$F_{M\_diff}^c = F_L^c - F_R^c \quad (6.21)$$

### 6.2.3 Adaptace parametrů modelu

Adaptace parametrů modelu tak, aby odpovídal původnímu obrazu vozovky probíhá jako minimalizace průměrné kvadratické odchylky externích sil působících na model.

$$F_{ext} = 0, \quad (6.22)$$

kde

$$F_{ext} = F_{M\_sum} \quad (6.23)$$

Pokud na středovou křivku nepůsobí žádná vnější síla jsou jeho změny tvaru a pozice nulové. Z tohoto předpokladu můžeme určit následující rovnici.

$$\begin{aligned} F_{ext} &= \eta(E_M(N) - E_M(N - 1)) \\ F_{ext} &= \eta M_R(t)(Q(N) - Q(N - 1)) \\ F_{ext} &= \eta M_R(t)\Delta Q(N) \end{aligned} \quad (6.24)$$

Velikost kroku změny řídicích bodů  $Q$  je vyjádřena pomocí  $\eta$ . Nová sada řídicích bodů v iteračním kroku  $N$  je dána vztahem:

$$Q(N) = Q(N - 1) + \Delta Q(N). \quad (6.25)$$

Externí sílu navzorkujeme pomocí středového b-splajnu a parametru modelu  $k$ . Úpravu řídicích bodů lze spočítat maticově:

$$\Delta Q(N) = \eta(MM^T)M^T F_{ext} \quad (6.26)$$

$$M = \begin{bmatrix} M_{-1} & 0 & \dots & \dots & 0 \\ 0 & M_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & M_n \end{bmatrix} \quad (6.27)$$

$$M_i = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 \\ t_2^3 & t_2^2 & t_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ t_m^3 & t_m^2 & t_m & 1 \end{bmatrix} \cdot \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad (6.28)$$

Parametr  $m$  udává počet vzorků externí síly v  $i$ -tém segmentu B-splajnu, parametr  $n$  je v našem případě roven  $n = 2$  (3 řídicí body).

Rozdíl mezi silami působících na levou a pravou stranu modelu určuje změnu parametru  $k$ . Velikost změny můžeme určit:

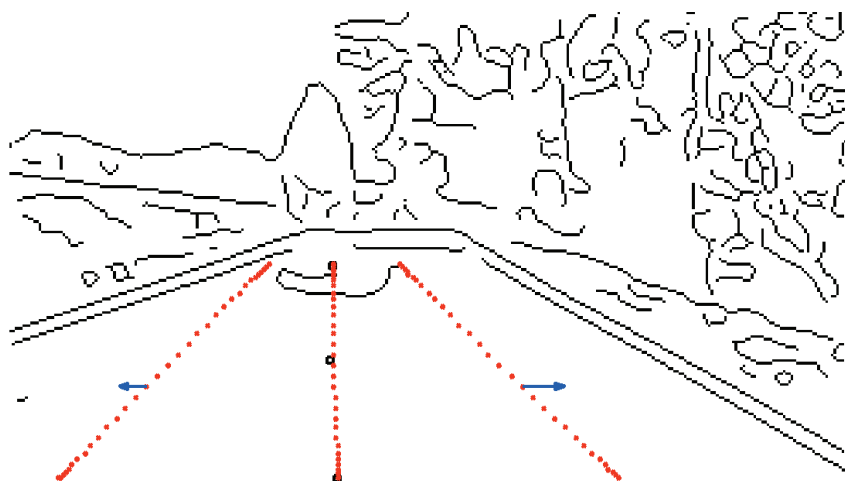
$$F_k = F_{Mdiff}^c \quad (6.29)$$

$$F_k = \tau(k(N) - k(N - 1)) = \tau\Delta k(N) \quad (6.30)$$

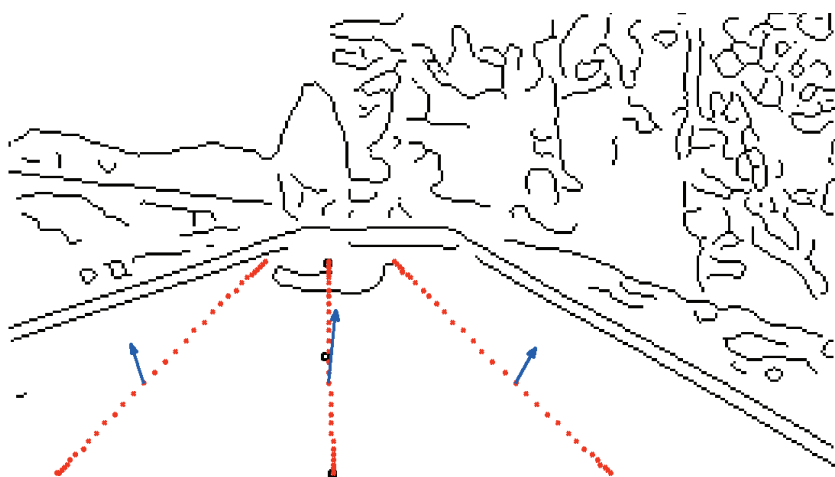
$$k(t) = k(t - 1) + \tau\Delta k(t). \quad (6.31)$$

Je-li  $\tau$  velikost kroku změny parametru  $k$ , získáme z předchozích rovnic:

$$\Delta k(t) = \frac{F_k}{\tau}. \quad (6.32)$$



Obr. 6.4: Vliv působících externích sil na parametr  $k$



Obr. 6.5: Vliv působících externích sil na deformaci středového b-splajnu

Jako externí síly používá tento model GVF, pro zrychlení lze použít jiné méně výpočetně náročná vektorová pole. Tyto pole však mají menší dosah.

Na obrázcích 6.4 a 6.5 jsou znázorněny působící síly na splajnový model.

Obrázek 6.4 ukazuje vliv sil na parametr  $k$ . Odečteme-li od sebe síly působící na bod pravé a levé strany (jen působících ve směru řádku obrazu) zjistíme, jak tento bod přispěje k celkové změně. Celková změna může být spočtena např. jako průměr dílčích změn.

Obdobně je zjišťována i síla manipulující se středovou křivkou, (vzorec 6.20). Celková změna pozice řídicích bodů však není jen jednoduchý průměr, ale musí se přepočíst na účinky jednotlivých příspěvků na konkrétní bod  $Q$  (vzorec 6.26).

## Výpočet

Jsou dány 3 řídicí body b-splajnu, pro výpočty jich však potřebujeme kvůli maticovým výpočtům 16, proto definujeme vektor řídicích bodů jako:

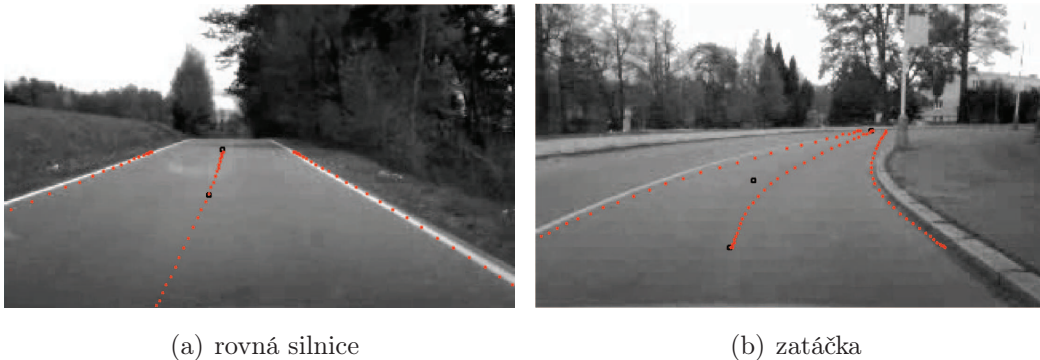
$$Q = [Q_1, Q_2, Q_3, Q_4, Q_2, Q_3, Q_4, Q_5, Q_3, Q_4, Q_5, Q_6, Q_4, Q_5, Q_6, Q_7]^T \quad (6.33)$$

$$Q_1 = Q_2 \quad (6.34)$$

Výpočet se řídí podle následujícího algoritmu:

1. Inicializace b-splajnu
2. Výpočet GVF aktuálního obrazu silnice
3. Výpočet změn parametrů  $\Delta k(N)$ ,  $\Delta Q(N)$  podle rovnic (6.32), (6.26)
4. Výpočet  $k(N)$  a  $Q(N)$
5. pokud je  $|\Delta k(N)| > \delta_1$ ,  $|\Delta Q(N)| > \delta_2$ , pokračuj krokem 3, jinak krok 6
6. konec algoritmu, další snímek

### 6.2.4 Výsledky metody



Obr. 6.6: Výsledek modelu b-splajn

Na obrázku 6.6 jsou ukázány výsledky této metody. Tato metoda dává obecně lepší výsledky než metoda hyperbol. To je dáno především možností b-splajnu obsáhnout mnohem větší škálu tvarů. Výsledky jsou také více předvídatelné, parametry modelu nejsou totiž určovány náhodně, ale za pomoci vektorového pole sil.

## 7 DETEKCE VOZOVKY NA ZÁKLADĚ ANALÝZY BAREV

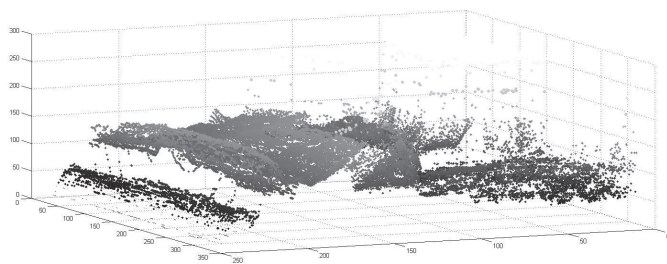
Silnice je charakteristická svou barvou. Tato barva je víceméně na celé vozovce stejná, vlivem parazitních vlivů je však její odstín různý (stín, odraz atd.). Další skutečnost je, že okolí má ve většině případů dosti odlišnou barvu, jsou zde však výjimky (např. Chodník) .

Z těchto skutečností vycházejí různé metody pro nalezení silnice pomocí barev.

### 7.1 Detekce vozovky pomocí metody mean-shift

Metoda mean-shift není primárně určena pro práci s obrazem, ale pro obecnou matematickou analýzu dat. Metoda pracuje s  $n$ -rozměrným prostorem. V prostoru se nacházejí data, tato data jsou reprezentována jako bod v daném prostoru. Metoda pracuje se skutečností, že data náležící do jedné skupiny jsou v tomto  $n$ -rozměrném prostoru blízko sobě. Mean-shift algoritmus je obecně iterativní postup pro nalezení lokálního maxima vzorků dat. Tento algoritmus je podrobně popsán v [2].

Pro realizaci mean-shift je tedy třeba obraz přetransformovat na tyto  $n$ -rozměrné body. Jednou z možností je reprezentace obrazu jako bodů v 3-rozměrném prostoru daný souřadnicemi v obraze a jeho intenzitou  $P(u, v, i)$ , na následujícím obrázku je zobrazen takto daný prostor.



(a) obrázek reprezentován 3D-prostorem



(b) původní obrázek

Obr. 7.1: Obraz jako 3-rozměrná data

#### 7.1.1 Princip metody mean-shift

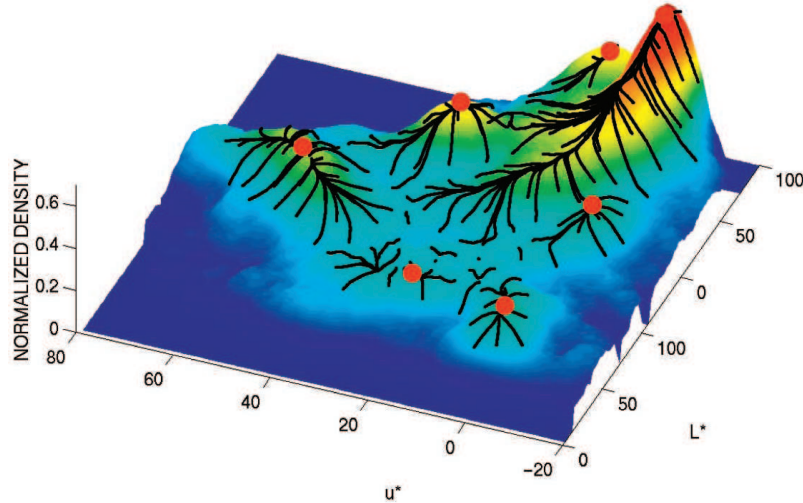
V daném bodu zkoumaného prostoru určíme jeho okolí. V tomto okolí vypočítáme gradient hustoty rozmístění všech bodů. Původní bod posune ve směru tohoto gradientu. Pokračujeme tak dlouho dokud se jeho pozice mění. Tyto operace provedeme se všemi body obrazu.



Pro odhad místa s nejvyšší hustotou lze použít obyčejný aritmetický průměr:

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i), \quad (7.1)$$

kde  $K$  je tzv. kernel, ten určuje okolí bodů, případně jejich váhy (pro aritmetický průměr  $K = 1$ , jinak libovolná funkce  $K = f(\mathbf{x})$ ) a  $n$  je počet bodů obsažených v okolí.



Obr. 7.2: Ukázka posuvu bodů pomocí mean-shift [2]

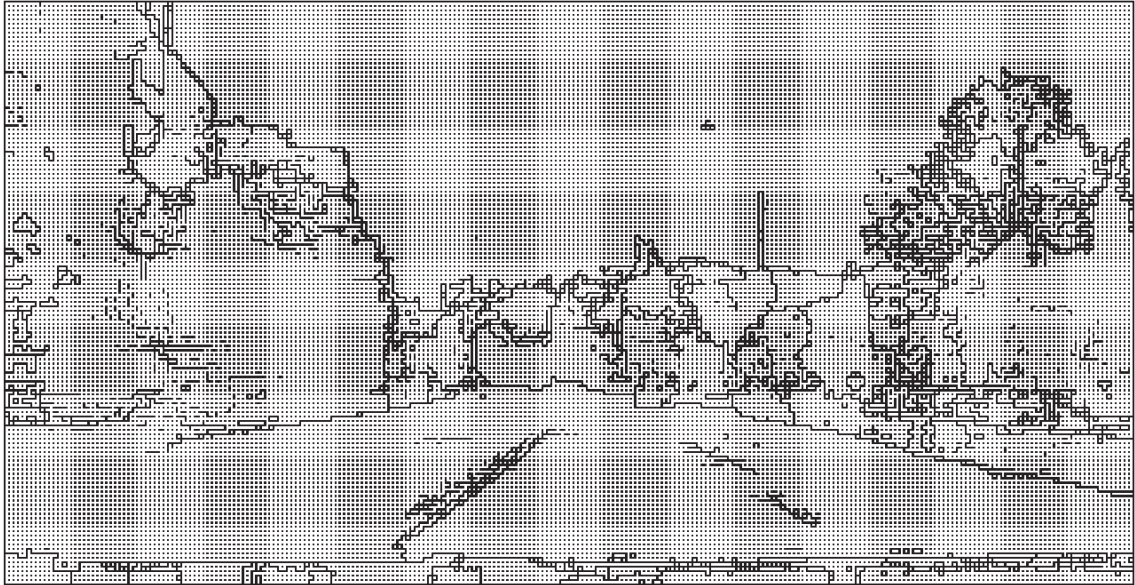
### 7.1.2 Shlukování blízkých bodů

Pro shlukování podobných bodů, tj. bodů které dokonvergovali k sobě pomocí mean-shiftu, existuje více způsobů. Můžeme pro každý bod hledat body ležící v nějakém malém okolí. Tato metoda je však velice výpočetně náročná. Pro každý bod musíme spočítat  $n - 1$  vzdáleností, kde  $n$  je počet všech bodu obrazu.

Další možností je použití supergridu [11] obzázek 7.3. Mezi každé sousední body obrazu vložíme prázdný bod. Postupně procházíme bod po bodu a zkoumáme, zda sousední body  $x_a, x_b$  splňují podmínku:

$$|\mathbf{x}_a - \mathbf{x}_b| \leq \varsigma, \quad (7.2)$$

kde  $\varsigma$  je zvolená největší vzdálenost dvou bodů. Pokud je podmínka (7.2) splněna tyto zkoumané body patří do téže skupiny.



Obr. 7.3: Ukázka supergridu

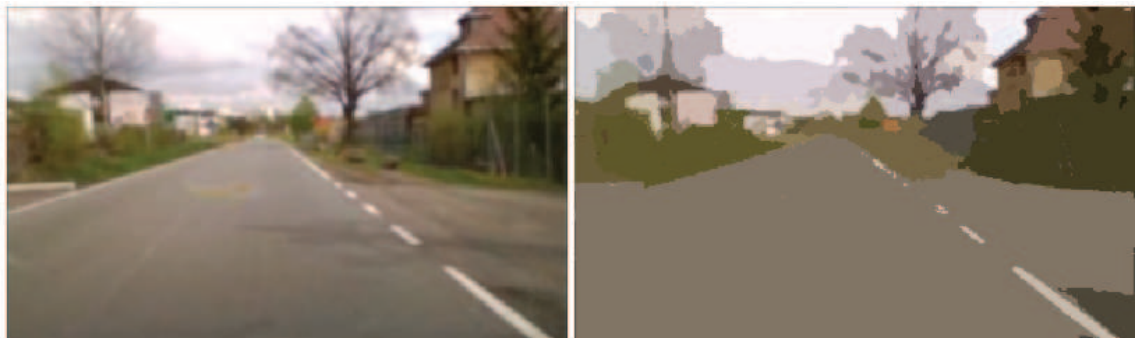
### 7.1.3 Realizace, výsledky

Pro nalezení vozovky byl použit 5-ti rozměrný prostor, kde každý bod je dán souřadnicemi  $P(u, v, H, S, V)$ . Jak je vidět ze souřadnic bodu, je použit barevný model HSV. Shlukování bodů je realizováno pomocí výše zmíněné metody supergrid. Na obrázku 7.4 jsou některé výsledky metody.

Metoda mean-shift poskytuje dosti přesné výsledky, je schopná rozlišit i chodník od vozovky, i když vypadají podobně. Přesnost a dobré výsledky jsou však vykoupěny značnou výpočetní náročností. Z každého bodu obrazu se vypočítává několik kroků mean-shift. Z testování metody lze vypočítat četnost provedení v jednom bodě, ta je rovna přibližně 100 krokům pro jeden pixel obrazu. Je-li obraz o rozlišení 320x240, pak je provedeno cca 8.106 kroků v 5-ti rozměrném prostoru, v každém kroku je navíc počítán gradient hustoty vzorků.



(a) test 1



(b) test 2

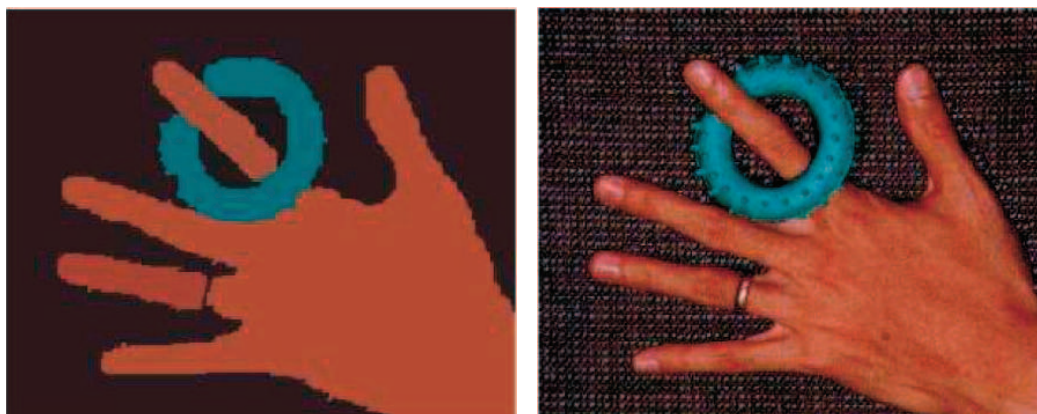
Obr. 7.4: Ukázka výsledku metody mean-shift

## 7.2 Detekce vozovky na základě statistického spojování regionů [9]

Tato metoda pracuje na principu spojování regionů. Spojování je prováděno na základě statistických charakteristik. Obraz je považován za pozorování neznámého teoretického obrazu, jehož statistické (*true*) regiony mohou být zrekonstruovány. Považujme obraz  $O(u, v)$  za pozorování teoretického obrazu  $O^*(u, v)$ . Obraz obsahuje  $|O(u, v)|$  obrazových bodů. Každý obrazový bod má 3 složky (RGB) z nichž každá může nabývat hodnot z množiny  $\{0; 1; 2 \dots; g\}$  (pro 24-bitový barevný obraz  $g = 256$ ).

V teoretickém obraze  $O^*(u, v)$  jsou obrazové body dokonale zastoupeny podle distribuční funkce, pomocí které jsou jednotlivé barevné složky vzorkovány. V takovémto obraze popisují statistické (*true*) regiony jednotlivé objekty sdílející stejné vlastnosti homogenity:

1. v statistickém regionu mají všechny pixely ve všech barevných složkách stejnou pravděpodobnost výskytu,
2. pravděpodobnost výskytu se v sousedních regionech liší alespoň v jednom z barevných kanálů.



(a) teoretický obraz  $O^*(u, v)$

(b) obraz  $O(u, v)$

Obr. 7.5: Ukázka obrazů [9]

Hodnoty barevných kanálů jsou v obraze  $O^*(u, v)$  nahrazeny  $q$  nezávislými náhodnými veličinami, nabývajících hodnot z rozsahu  $\langle 0; \frac{q}{q} \rangle$ .

Pro výpočty definujeme následující funkce:

$$P(R, R') = \begin{cases} true & \text{pokud } \forall a \in \{RGB\}, |\bar{R}'_a - \bar{R}_a| \leq \left| \sqrt{b^2(R) + b^2(R')} \right| \\ false & \text{jinak} \end{cases} \quad (7.3)$$

$$b(R) = g \sqrt{\frac{1}{2q|R|} \ln \left( \frac{|R|_{|R|}}{\psi} \right)}. \quad (7.4)$$

Funkce  $P(R, R')$  udává zda dvojice regionů  $(R, R')$  je ekvivalentní, to znamená jedná se o stejný statistický region. Parametr  $\psi$  udává povolenou toleranci pravděpodobnosti výskytu pixelů v jednom regionu. Jedna z možných hodnot je:

$$\psi = \frac{1}{|O(u, v)|^2} \quad (7.5)$$

### 7.2.1 Algoritmus

Algoritmus pracuje s 4-okolím, tomu odpovídá počet sousedících pixelů  $N < 2|O(u, v)|$ . Nechť  $S_O$  je soubor těchto párů. Dále definujeme reálnou funkci  $f(p, p')$ , kde  $p$  a  $p'$  jsou pixely obrazu  $O(u, v)$ . Postup algoritmu je následující:

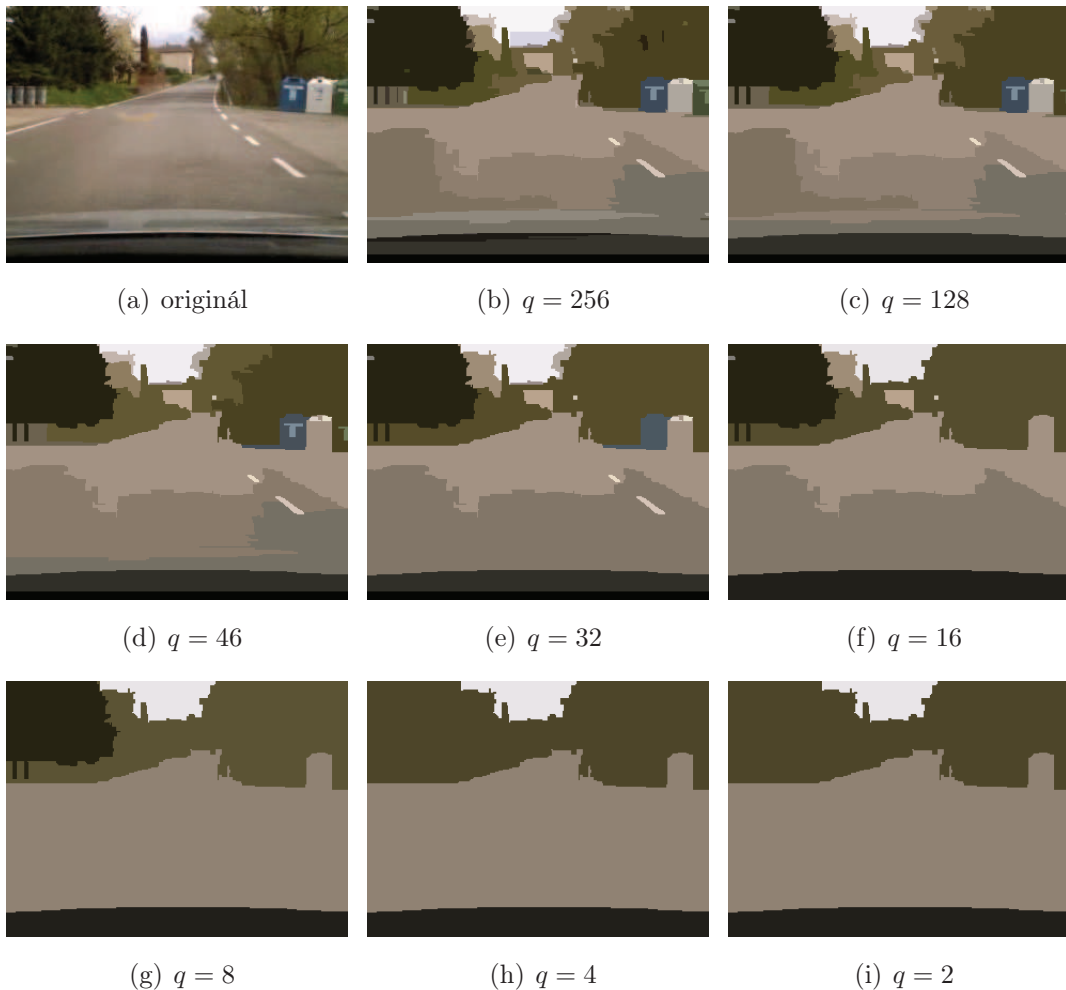
1. Vzestupné seřazení párů obsažených v  $S_O$  podle velikosti funkce  $f(p, p')$
2. Pro každý pár  $(p, p') \in S_O$  pro které platí:  $R(p) \neq R(p')$  provést test  $P(R(p), R(p'))$
3. Je-li test úspěšný spojíme  $R(p)$  s  $R(p')$

Pro volbu funkce  $f(p, p')$  můžeme použít různé metody např.:

1.  $f(p, p') = \max_a |p_a - p'_a|$
2. gradient obrazu  $O(u, v)$  a pod.

## 7.2.2 Výsledky metody

Tato metoda je velmi rychlá (oproti metodě mean-shift o několik řádů rychlejší), další výhodou je, že nepotřebuje žádné transformace vstupního obrazu. Mezi pozitivní vlastnosti můžeme uvažovat také jediný parametr a to počet náhodných veličin  $q$ . Tento parametr pak ve výsledku ovlivní do kolika tříd bude obraz rozdělen. Na obrázku jsou výsledky metody pro různé hodnoty  $q$ .



Obr. 7.6: Ukázka výsledků metody pro různé hodnoty  $q$

## 8 DETEKCE PŘEKÁŽEK

Tato práce má za úkol také zpracovat možnosti pro detekci překážek na vozovce. Překážek je nepřehledné množství, většina z nich ale má některé společné znaky, podle kterých je lze hledat. V této kapitole budeme pracovat s některými z nich.

Veškeré zde popsané metody pro hledání překážek pracují na principu generování a verifikaci hypotéz. Ukázka takovéto dvojice hypotéza-potvrzení hypotézy je na obrázku 8.1

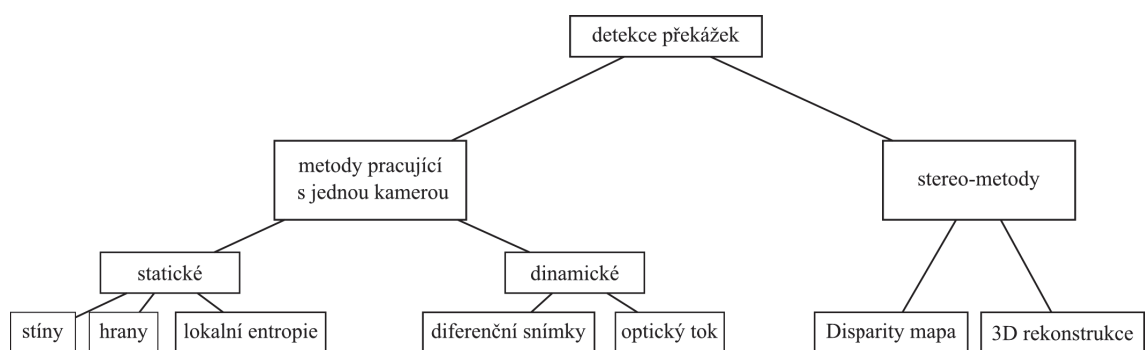


(a) všechny vygenerované hypotézy

(b) potvrzené hypotézy - překážky

Obr. 8.1: Dvojice hypotéza-potvrzení hypotézy

### 8.1 Možné metody pro hledání překážek



Obr. 8.2: Základní dělení metod pro detekci překážek

Na obrázku 8.2 je přehledové schéma možností při hledání překážek. V zásadě metody můžeme delit do dvou základních skupin:

1. Metody pracující s jednou kamerou, tzn. máme jen 2D informaci popřípadě 2D informaci v různých časech (video, sekvence snímků).

2. Metody vyžadující 2 a více kamer (stero-metody), jsme schopni zrekonstruovat úplný 3D model scény

Jelikož v této práci pracujeme jen s jednou kamerou stero-metody zde zabývat nebudeme.

## 8.2 Základní metody pro detekci překážek

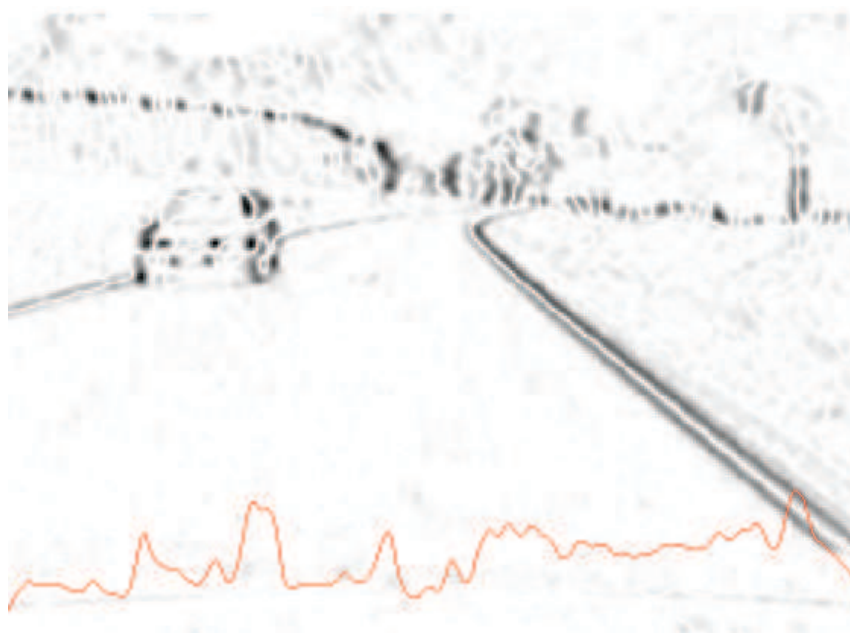
Staické metody pracují v podstatě na dvou různých principech:

1. Detekce na základe hran.
2. Detekce pomocí analýzy textur

Dynamické metody se pak zabývají změnami obrazu v čase. Těmito metodami se zda však zabývat nebudeme a to z důvodu srovnatelných výsledků se statickými, dále jede-li před námi auto stejnou rychlostí tyto metody ho nerozpoznají.

### 8.2.1 Metoda založená na vertikálních a horizontálních hranách

Tato metoda je založena na předpokladu, že překážky mají ostré hrany, tyto hrany pak mají většinou pravoúhlý průběh. Na základě tohoto předpokladu hledáme překážky jako určité uskupení vertikálních a horizontálních hran. Metoda je určena především pro detekci umělých předmětů jako jsou papř. vozidla. V [7] je podrobně popsána detekce vozidla pomocí vertikálních a horizontálních hran.

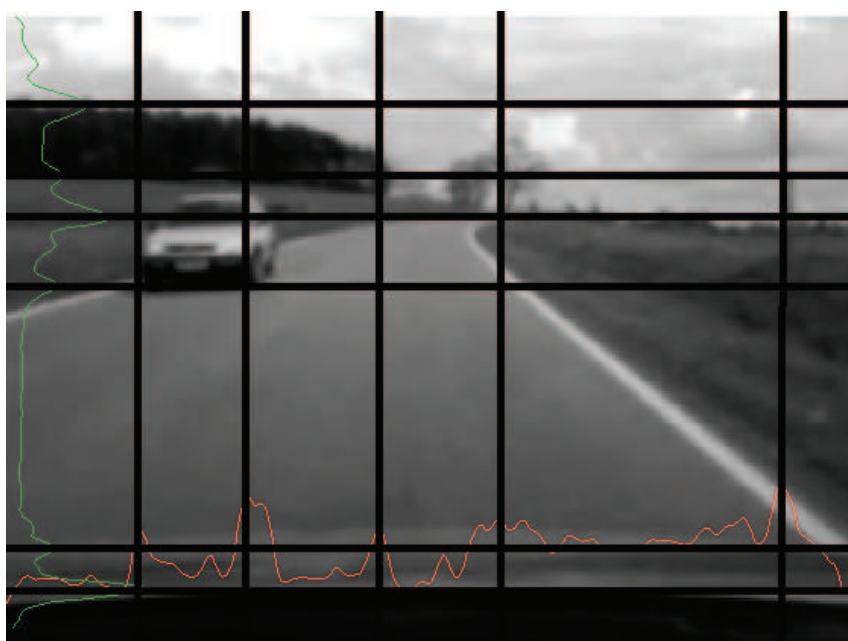


Obr. 8.3: Histogram sum hran v řádcích

Detekce krajů překážky probíhá jako hledání lokálních maxim ve vyhlazeném (plovoucí průměr) histogramu sum hran v jednotlivých sloupcích obrazu (obrázek 8.3).

Jednotlivá maxima seřadíme podle jejich velikosti, postupně bereme jednotlivá maxima a zkoumáme zda nové maximum není příliš blízko již vybranému, v takovém případě toto maximum ignorujeme.

Tento postup můžeme použít i pro horizontální směr. kombinací obou směrů dostaneme několik oblastí s potencionální překážkou (obrázek 8.4). Zpřesnění výsledků lze dosáhnout například tím že tuto metodu použijeme na obrazy s různým měřítkem [13].



Obr. 8.4: Nalezená maxima

### Použití obrazu ve více měřítkách

U obyčejné metody detekce pomocí hran je několik nedostatků. Pro správnou funkci detektoru musíme vybrat opravdu silné kandidáty, to v mnoha případech může být dosti obtížné, pro správné nalezení kandidátů na překážku je třeba právě vybrat maxima histogramu. Na nalezení maxim se ale podílí více faktorů: výběr operátora a prahů hranového detektoru, vyhlazovací filtr, nejmenší vzdálenost maxim a v neposlední řadě sám histogram, ten může být dosti zašuměný. Některá nevýhody odstraňuje právě více-měřítkový detektor.



Definujme vstupní obraz  $O$ , nechť  $O = O^{(K)}$ , kde  $K$  značí úroveň (stupeň měřítka) subobrazu. Vztah mezi obrazem a subobrazem můžeme vyjádřit jako:

$$O^{(K-1)} = REDUCE(O^{(K)}). \quad (8.1)$$

kde  $REDUCE$  značí libovolnou operaci redukce rozlišení obrazu. Jednu z možností vyjadřuje následující vzorec:

$$O^{(K-1)}(x) = O^{(K)}(2x). \quad (8.2)$$



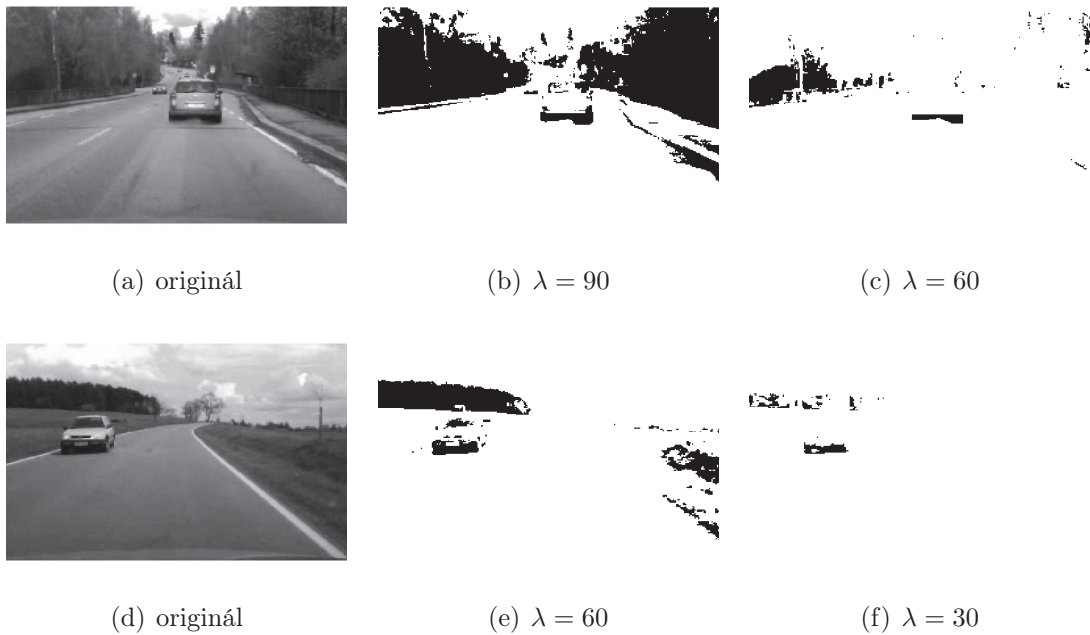
Obr. 8.5: Kaskada subobrazů (jemnost detailu  $\uparrow$ ) [13]

Maxima začneme hledat na nejnižší úrovni (obrázek 8.5 první řádek). Jak je vidět obraz postrádá většinu detailů, podstatné svislé a vodorovné hrany však zůstali. Z nejnižší úrovně pak postupujeme do úrovně s větší jemností, zde zpřesníme pozici již nalezených maxim. Tímto postupem se zbavíme falešných maxim, neboť jsme začali z obrazu, kde jsou jen velmi silné hrany. Pro ještě zvětšení robustnosti můžeme použít např. poměr po sobě jdoucích minim a maxim, absolutní hodnoty maxim a podobně.

### 8.2.2 Detekce překážek pomocí stínu [14]

Každá překážka na vozovce vrhá stín, této skutečnosti využijeme pro jejich detekci. V [8] jsou uvedeny základní postupy při detekci podle stínu.

Základní vlastnost stínu překážky je ta, že jeho hodnota jasu je vždy menší než hodnota jakéhokoliv bodu vozovky. Problémem je, která hodnota jasu patří stínu a jaká už ne. Nejjednodušší by bylo určit hodnotu prahu dostatečně nízkého, aby správně označoval stín v co nejvíce případech.



Obr. 8.6: Ukázka stínu pro různé hodnoty prahu

Jak je zřejmé z obrázku 8.6, volba pevného prahu je v reálných podmínkách neúčinná.

Jako první krok pro určení hodnoty prahu stínu  $\lambda$  potřebujeme zjistit jaké body obrazu náleží silnici. K tomuto úkolu můžeme s výhodou použít lokální entropii [5]:

$$H(x_k) = - \sum_{k \in C} p(x_k) \ln p(x_k), \quad (8.3)$$

kde  $p(x_k)$  udává pravděpodobnost výskytu hodnot  $x_k$ , sumace pak probíhá přes taková  $x_k$ , která patří do okolí  $C$ . Entropie obecně udává „míru neuspořádanosti“, toho využijeme k nalezení oblasti prázdné vozovky. Segmentace oblasti je jednoduché prahování s pevným prahem (cca 70 % maximální hodnoty entropie). Ukázka takové oblasti je na obrázku 8.7.

Dalším krokem k nalezení optimálního adaptivního prahu  $\lambda$  je vytvoření histogramu hodnot jasu oblasti volné vozovky. Máme-li tento histogram aproximujeme ho normálním rozložením pravděpodobnosti (obrázek 8.8):

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (8.4)$$

kde  $\mu$  představuje průměrnou hodnotu a  $\sigma$  rozptyl. Ideální práh pro stín pak vypočteme podle:

$$\lambda = \mu - 3\sigma. \quad (8.5)$$

Ukázky jsou na obrázku 8.9

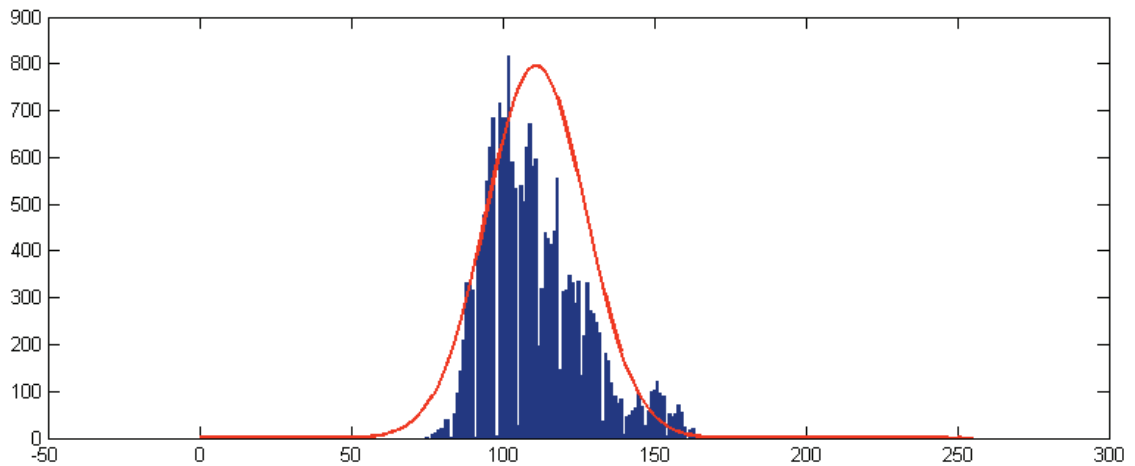


(a) volná silnice



(b) volná silnice

Obr. 8.7: Volná silnice



Obr. 8.8: Aproximace histogramu normálním rozložením pravděpodobnosti

Tímto postupem jsme se do značné míry zbavili nepříjemné vlastnosti pevného prahu, totiž silné závislosti na okolních světelných podmínkách. Naproti tomu adaptivní práh si se změnami světelných podmínek, barvy vozovky, typu scény apod. dokáže poradit.

Jestli-že aproximujeme oblast volné vozovky trojúhelníkem, zůstanou nám jen stíny na vozovce. Dále pak jsou-li stíny nad sebou, bereme v úvahu jen ten níže položený. Toto je poslední úprava u této metody, neboť nám zůstaly již jen stíny překážek. Z těchto stínů již můžeme určit jejich pozici (obrázek 8.10).



(a) originál



(b)  $\lambda = 90$



(c)  $\lambda = 60$



(d) originál

Obr. 8.9: Ukázka stínu pro různé hodnoty prahu

### 8.2.3 Detekce překážek pomocí entropie [13],[5]

Entropii jsme použili již u detekce pomocí stínů, kde jsme se pomocí ní snažili získat prázdné místo. Pokud myšlenku otočíme, můžeme hledat právě oblasti, které jsou oproti svému okolí značně odlišné - překážky. Postup je podobný jako u detekce prázdné vozovky, jen provedeme prahování opačným směrem. Výsledek nalezneme na obrázku 8.11.

Z tohoto výsledku však nelze jednoduše zjistit, kde je překážka a kde je pouze rozhraní dvou ploch. Proto se tato metoda samotná pro detekci překážek nepoužívá. Je to však silný indikátor jejich přítomnosti. Metoda je ideální pro kombinaci s jinými metodami.



Obr. 8.10: Nalezená překážka pomocí stínu



Obr. 8.11: Výsledek po prahování výsledku lokální entropie

## 8.3 Kombinace metod pro detekci překážek

Pro zlepšení výsledků detekce překážek a minimalizaci falešných hypotéz můžeme výše popsané metody zkombinovat. Možností, jak tyto metody zkombinovat, je celá řada, od prosté operace AND až po samo-organizační mapy [3]. My použijeme kombinaci metod podle následujícího postupu.

Jako první krok si vypočítáme hranový obraz Sobelovým operátorem v horizontálním a vertikálním směru pro každou složku (RGB). Následně provedeme jejich průnik. Na rozdíl od metody popsané v kapitole 8.2.1 nám bude stačit pouze zjistit oblasti, pro které platí, že v jejich oblasti se nachází dostatek silných vertikálních nebo horizontálních hran. Toho je dosaženo konvolucí binárního hranového operátoru s průměrovací maskou o rozměrech  $21 \times 21$  a následným prahováním hodnotou rovnou polovině maxima takto vzniklého obrazu.



(a) originál

(b) hranový obraz

(c) výsledek po průměrování a prahování

Obr. 8.12: Postup při hledání příznaků pomocí hran

Každá překážka vrhá svůj stín, toho využijeme jako dalšího příznaku. V kapitole 8.2.2 je popsán postup. Uvažujme překážku, která vrhá stín, její přítomnost tedy budeme hledat pouze nad tímto stínem.

Posledním použitým příznakem je entropie podle kapitoly 8.2.3, obrázek 8.11.

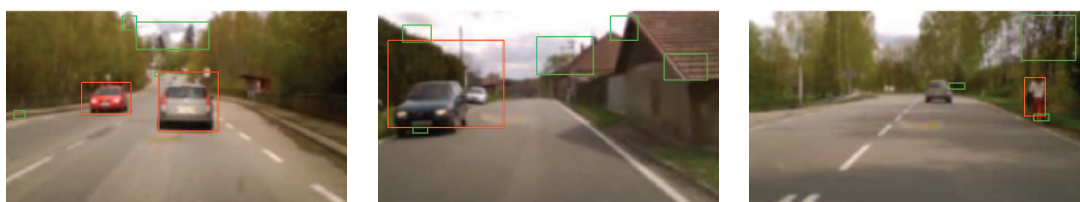
Vlastní kombinace metod spočívá v sečtení jejich výsledků a naprahování dvěma (to odpovídá jejich hlasování, kde se alespoň dvě metody shodnou).

Na obrázku 8.14 je ukázka výsledků takovéto kombinace metod.

Jako verifikaci všech nalezených kandidátů, tj. dostaly alespoň jeden hlas, slouží právě jejich kombinace (3 hlasy). Jak již bylo řečeno je možné použít i jiné přístupy, tento postup však dosahuje uspokojivých výsledků a je velmi jednoduchý a rychlý.



Obr. 8.13: Možné oblasti překážek podle příznaku stínu



(a)

(b)

(c)

Obr. 8.14: Postup při hledání příznaků pomocí hran

## 9 PROGRAM PRO DETEKCI SILNICE A PŘEKÁŽEK

Výsledný algoritmus pro detekci vozovky a překážek je kombinací některých výše popsaných metod. Pro detekci vozovky je použita pouze metoda statistického spojování regionů. Další metody nebyly použity proto, že tato metoda poskytuje v celku přesné výsledky, také je oproti jiným dosti rychlá. Samozřejmě je možné další metody přidat. Metoda pracuje s parametrem  $q = 8$ . Tento parametr byl zvolen na základě testování, při této hodnotě byla metoda ještě dost robustní a přitom selhávala v minimálním počtu případů.

Detekce překážek je řešena postupem popsaným v kapitole 8.3.

Takto sestavený program dokáže na počítači s procesorem Intel Core i5 s pracovní frekvencí 2.66 GHz v prostředí MATLAB zpracovat cca 1.6 snímku/s. Tato hodnota je přímo odstrašující. Největší díl na takto pomalém algoritmu je samo prostředí programu MATLAB, případným přepsáním do kódu C/C++ bychom dosáhli řádově lepších výsledků. S ohledem na tuto rychlost také byli ve finálním programu vynechány složité metody detekce vozovky.

Některé výsledky algoritmu jsou na následujícím obrázku. Je na nich videt že program generuje dost falešných překážek, tato vlastnost je dána jednoduchým klasifikátorem, temn je možné nahradit složitějšími (neuronová síť, rozhodovací stromy ...)





(a)

(b)

(c)



(d)

(e)

(f)



(g)

(h)

Obr. 9.1: Ukázka výsledků algoritmu

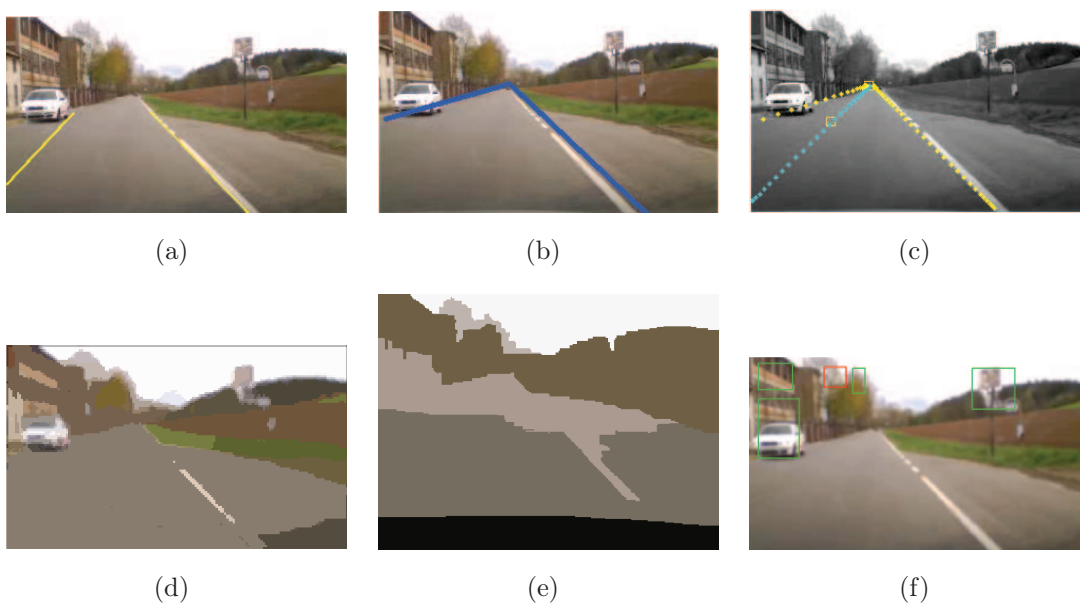
## 10 ZÁVĚR

V této práci jsme realizovali různé metody a algoritmy pro hledání vozovky respektive jízdnic pruhů. Vyzkoušeli jsme různé přístupy, ať již se jedná o metody vycházející z hranového obrazu nebo barevné informace.

### 10.1 Porovnání metod

Porovnáním metod zjistíme, že jsou metody dávající lepší výsledky a metody s horšími výsledky. Dalším aspektem je též výpočetní náročnost a s tím spojené doby výpočtů.

Z hlediska kvality výsledků vede metoda mean-shift a metoda založená na hranách s modelem b-splajn křivky. Výpočty mean-shift metody však trvají neúnosně dlouho. Naproti tomu, metoda statistického spojování regionů dosahuje ve většině případů výsledků srovnatelných s mean-shif. Je však nepoměrně rychlejší.



Obr. 10.1: Ukázky výsledků jednotlivých metod

### 10.2 Budoucí práce

Budoucí práce bude směřovat k použití složitějšího klasifikátoru pro ověřování hypotéz překážek. Také bude kladen větší důraz na rychlost výpočtů s ohledem na realtime provoz. Ten v současném stupni zpracování není myslitelný. Po vyřešení

problémů s rychlostí bude také možné skloubit více metod pro kvalitnější určení jízdních pruhů.

## LITERATURA

- [1] Aarts, E.; Korst, J.; Michiels, W.: Simulated Annealing. In *Search Methodologies*, editace E. K. Burke; G. Kendall, Springer US, 2005, ISBN 978-0-387-28356-2, s. 187–210, 10.1007/0-387-28356-0\_7.  
URL [http://dx.doi.org/10.1007/0-387-28356-0\\_7](http://dx.doi.org/10.1007/0-387-28356-0_7)
- [2] Comaniciu, D.; Meer, P.: Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník 24, č. 5, may 2002: s. 603–619, ISSN 0162-8828, doi:10.1109/34.1000236.
- [3] Handmann, U.; Kalinke, T.: Fusion of texture and contour based methods for object recognition. In *Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on*, nov 1997, s. 876–881, doi:10.1109/ITSC.1997.660589.
- [4] Hlaváč, V.; Šonka, M.: *Počítačové vidění*. Praha, Czech Republic: Grada, 1992, ISBN 80-85424-67-3, 252 s.
- [5] Kalinke, T.; Seelen, W. v.: Entropie als Maß des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeitssteuerung. In *Mustererkennung 1996, 18. DAGM-Symposium*, London, UK: Springer-Verlag, 1996, ISBN 3-540-61585-7, s. 627–634.  
URL <http://portal.acm.org/citation.cfm?id=648281.754828>
- [6] Kluge, K.; Lakshmanan, S.: A deformable-template approach to lane detection. In *Intelligent Vehicles '95 Symposium., Proceedings of the*, sep 1995, s. 54–59, doi:10.1109/IVS.1995.528257.
- [7] Matthews, N. D.; An, P. E.; Charnley, D.; aj.: Vehicle detection and recognition in greyscale imagery. *Control Engineering Practice*, ročník 4, č. 4, 1996: s. 473–479, ISSN 0967-0661, doi:DOI:10.1016/0967-0661(96)00028-7.  
URL <http://www.sciencedirect.com/science/article/B6V2H-3VVTWFC-5/2/8660c974ee4dd25b55d7b5547af52f92>
- [8] Mori, H.; Charkari, N.: Shadow and rhythm as sign patterns of obstacle detection. In *Industrial Electronics, 1993. Conference Proceedings, ISIE'93 - Budapest., IEEE International Symposium on*, 1993, s. 271–277, doi:10.1109/ISIE.1993.268796.
- [9] Nock, R.; Nielsen, F.: Statistical region merging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník 26, č. 11, nov. 2004: s. 1452–1458, ISSN 0162-8828, doi:10.1109/TPAMI.2004.110.

- [10] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Brno, Czech Republic: Computer Press, druhé vydání, 2005, ISBN 80-251-0454-0, 608 s.
- [11] Sonka, M.; Hlavac, V.; Boyle, R.: *Image Processing, Analysis, and Machine Vision*. CL-Engineering, 2007, ISBN 9780495082521.  
URL <http://www.amazon.com/Image-Processing-Analysis-Machine-Vision/dp/049508252X%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D049508252X>
- [12] Suman, B.; Hoda, N.; Jha, S.: Orthogonal simulated annealing for multiobjective optimization. *Computers and Chemical Engineering*, ročník 34, č. 10, 2010: s. 1618 – 1631, ISSN 0098-1354, doi: DOI:10.1016/j.compchemeng.2009.11.015.  
URL <http://www.sciencedirect.com/science/article/B6TFT-4XSJVPF-2/2/02fc5ae8b7761e69ef145e525eb27438>
- [13] Sun, Z.; Bebis, G.; Miller, R.: Monocular precrash vehicle detection: features and classifiers. *Image Processing, IEEE Transactions on*, ročník 15, č. 7, july 2006: s. 2019 –2034, ISSN 1057-7149, doi:10.1109/TIP.2006.877062.
- [14] Tzomakas, C.; von Seelen, W.: Vehicle Detection in Traffic Scenes Using Shadows. Technická zpráva, IR-INI, INSTITUT FUR NUEROINFORMATIK, RUHR-UNIVERSITAT, 1998.
- [15] Wang, Y.; Teoh, E. K.; Shen, D.: Lane detection using B-snake. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, 1999, s. 438 –443, doi:10.1109/ICIIS.1999.810313.
- [16] Xu, C.; Prince, J.: Gradient vector flow: a new external force for snakes. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, jun 1997, s. 66 –71, doi:10.1109/CVPR.1997.609299.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

RGB	barevný prostor se složkami: červená, zelená, modrá – red, green, blue
$L^*a^*b^*$	barevný prostor se složkami: L – světelnost, a – osa zelená - červená, b – osa modrá - žlutá
HSV	barevný prostor se složkami: H – barevný tón, S – sytost barvy, V – hodnota jasu
HSL	barevný prostor se složkami: H – barevný tón, S – sytost barvy, V – hodnota jasu
$u, v$	souřadnice $x, y$ v obrazové rovině
$x, y, z$	souřadnice $x, y, z$ v 3D prostoru
$f$	ohnisková vzdálenost optické soustavy
$O_x, O_y, O_z$	osy 3D prostoru
$P(x, y, z)$	bod 3D prostoru
$P(u, v)$	bod v rovině kamery
$O(u, v)$	Obraz z kamery
$m, n$	velikost obrazu z kamery $m \times n$
$\theta$	úhel mezi optickou osou kamery a osou $x$
$\alpha$	zorný úhel kamery
$\alpha_h, \alpha_v$	horizontální a vertikální zorný úhel kamery
$h$	vzdálenost kamery od země
$l$	vzdálenost kamery od počátku souřadného systému po ose $x$
$d$	vzdálenost kamery od počátku souřadného systému po ose $y$
$K(h, l, d)$	pozice kamery v prostoru
GVF	gradientní vektorové pole – gradient vector flow
$f(x, y)$	gradient obrazu

$\mu$	parametr udávající míru vlivu integrální složky a složky parciální derivace vektorového pole
L,R	levá a pravá krajnice
$r$	řádek obrazu – row
$c$	sloupec obrazu – column
$k$	křivost oblouku hyperboly
$hz$	řádek obrazu odpovídající horizontu
$b$	odsazení oblouku hyperboly od horizontu
$vp$	vertikální posun hyperboly
$\alpha_{width}$	parametr udávající požadovanou minimální šířku pruhu hyperbolického modelu
$s$	soubor parametrů hyperbolického modelu
$gm(r, c)$	velikost hrany v bodě obrazu (r,c) – gradient magnitude
$gd(r, c)$	směr hrany v bodě obrazu (r,c) – gradient direction
$\alpha_{prox}$	konstanta určující požadovanou míru schody mezi modelem a hranovým obrazem
$\alpha_{orient}$	konstanta určující požadovanou míru schody mezi modelem a orientací hrany
$L(sN)$	věrohodnost modelu hyperbol pro parametry $s$ v kroku $N$
$\Delta$	velikost změny
$T_{init}$	počáteční teplota v algoritmu simulovaného žíhání
$T_{final}$	koncová teplota
$maxiter$	maximální počet kroků simulovaného žíhání
$\rho()$	funkce na jejíž hodnotě se rozhoduje o novém parametru $s$
$Q_n$	n-tý kontrolní bod křivky b-splajn
$O^*(u, v)$	teoretický obraz

$g$	nejvyšší hodnota barevné složky
$\psi$	tolerance pravděpodobnosti výskytu v jednom regionu
$R_l$	region s $l$ pixely
$\lambda$	práh pro segmentaci stínu
$\mu$	průměrná hodnota
$\sigma$	rozptyl



## SEZNAM PŘÍLOH

- A DVD s elektronickou verzí této práce, zdrojovými texty a testovacími soubory 66

A DVD S ELEKTRONICKOU VERZÍ TÉTO PRÁCE,  
ZDROJOVÝMI TEXTY A TESTOVACÍMI SOU-  
BORY