



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DETEKTORY A DESKRIPTORY OBLASTÍ V OBRAZU

REGION DETECTORS AND DESCRIPTORS IN IMAGE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Žilka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Karel Horák, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Filip Žilka

ID: 147010

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Detektory a deskriptory oblastí v obrazu

POKYNY PRO VYPRACOVÁNÍ:

1. Nastudujte problematiku detektorů významných oblastí obrazu a sestavte přehled s příklady.
2. Nastudujte problematiku deskriptorů významných oblastí obrazu a sestavte přehled s příklady.
3. Podle pokynů vedoucího práce proveďte experiment nad zadanými obrazy, ve kterém aplikujete detektory a deskriptory oblastí.
4. Vyhodnoťte provedený experiment z hlediska opakovatelnosti a stability jednotlivých detektorů a deskriptorů.
5. Na zadané úloze sledování popř. rozpoznávání objektů implementujte mechanismus využívající významných bodů v obrazu a srovnajte výsledky všech implementovaných algoritmů.

DOPORUČENÁ LITERATURA:

1. GONZALES, R.C., WOODS, R.R.: Digital image processing (3rd edition). Prentice-Hall, Inc., 2008. 954 pages. ISBN 978-0131687288.
2. YOUNG, I.T., GERBRANDS, J.J., VLIET, L.J.: Fundamentals of Image Processing. TU Delft, 1998. 113 pages. ISBN 9075691017.
3. SZELINSKI, R.: Computer Vision: Algorithms and Applications. Springer, 2010. ISBN 978-1848829343.

Termín zadání: 8.2.2016

Termín odevzdání: 16.5.2016

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

Konzultant diplomové práce:

doc. Ing. Václav Jirsík, 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

Táto diplomová práca sa zaoberá významnou oblasťou počítačového videnia a to konkrétne detektormi a deskriptormi oblastí v obraze. Budú predstavené najjednoduchšie návrhy na detekciu významných bodov počnúc Moravcovým detektorom až po komplexné detektory ako napríklad MSER alebo FAST. Popis týchto významných bodov je cieľom deskriptorov oblastí. Rovnako tu bude uvedený postupný prehľad od počiatkov, za ktoré možno považovať SIFT až po najnovšie a najstabilnejšie deskriptory ako napríklad FREAK alebo ORB. Jadrom práce bude následné testovanie a porovnanie vybraných metód na zvolenej úlohe lokalizácie ŠPZ v obraze.

Klíčová slova

Detektory oblastí v obraze, deskriptory oblastí v obraze, Harrisov operátor, MSER, FAST, SIFT, SURF, BRIEF, BRISK, ORB, FREAK, Lokalizácia ŠPZ

Abstract

This master's thesis deals with an important part of computer vision field. Main focus of this thesis is on feature detectors and descriptors in an image. Throughout the thesis the simplest feature detectors like Moravec detector will be presented, building up to more complex detectors like MSER or FAST. The purpose of feature descriptors is in a mathematical description of these points. We begin with the oldest ones like SIFT and move on to newest and best performing descriptors like FREAK or ORB. The major objective of the thesis is comparison of presented methods on licence plate localization task.

Keywords

Feature detectors, feature descriptors, Harris operator, MSER, FAST, SIFT, SURF, BRIEF, BRISK, ORB, FREAK, Licence plate localization

Bibliografická citace:

ŽILKA, F. Detektory a deskriptory oblastí v obrazu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 86 s. Vedoucí diplomové práce byl Ing. Karel Horák, Ph.D..

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Detektory a deskriptory oblastí v obraze jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: **16. května 2016**

.....
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: **16. května 2016**

.....

podpis autora

OBSAH

Úvod.....	11
1 Detektory významných bodov a oblastí v obraze	14
1.1 Významná oblasť	14
1.2 Možnosti detekcie	15
1.3 Moravcov detektor (operátor)	15
1.4 Harrisov detektor (operátor).....	18
1.5 MSER.....	22
1.6 FAST	25
2 Deskripty oblasti v obraze	27
2.1 HoG	27
2.2 SIFT.....	29
2.3 SURF	34
2.4 BRIEF	37
2.5 BRISK	38
2.6 ORB.....	42
2.7 FREAK.....	43
3 Experimentálna časť	46
3.1 Testovacia galéria.....	47
3.2 Špecifikácia ŠPZ	48
3.3 Lokalizácia ŠPZ použitím detektora MSER	49
3.4 Lokalizácia ŠPZ použitím Harrisovho detektora	60
3.5 Lokalizácia ŠPZ použitím Shi-Tomasi detektora	67
3.6 Lokalizácia ŠPZ použitím detektora FAST	68
3.7 Lokalizácia ŠPZ použitím deskriptora BRISK	69
3.8 Lokalizácia ŠPZ použitím deskriptoru SURF	72
3.9 Použitie ostatných algoritmov k lokalizácii	76
3.10 Sumarizácia dosiahnutých výsledkov.....	78
Záver	80

ZOZNAM OBRÁZKOV

Obrázok 1: Reťazec spracovania obrazu [21].....	12
Obrázok 2: a) oblasť bez textúry, b) detekcia hrany, c) detekcia rohu [5]	14
Obrázok 3: Pohyb vyšetrovacieho okna pri Moravcovom operátore [6].....	16
Obrázok 4: Ukážka výsledku algoritmu Moravcovho detektora na testovacie obrázky	18
Obrázok 5: Klasifikácia významných oblastí podľa Harrisu [5]	20
Obrázok 6: Ukážka výsledku algoritmu Harrisovho detektora na testovacom obrázku.	22
Obrázok 7: Ukážka výsledku algoritmu MSER detektora na testovacom obrázku.....	24
Obrázok 8: K vysvetleniu princípu FAST [12]	25
Obrázok 9: Ukážka výsledku algoritmu FAST detektora na testovacie obrázky	26
Obrázok 10: Ukážka výsledku algoritmu HoG detektora na testovacie obrázky	28
Obrázok 11: Princíp tvorby scale-space pomocou DoG výpočtu [10]	30
Obrázok 12: Princíp vyhľadávania lokálneho extrému medzi priestormi vytvorenými pomocou DoG [10]	30
Obrázok 13: SIFT - princíp tvorby deskriptora [10]	32
Obrázok 14: Ukážka výsledku algoritmu SIFT deskriptora na testovacie obrázky	34
Obrázok 15: Druhé derivácie Gaussovej funkcie (podľa y a xy) a ich diskrétna aproximácia [11].....	35
Obrázok 16: Ukážka výsledku algoritmu SURF deskriptora na testovacie obrázky.....	36
Obrázok 17: Rozloženie dvojíc porovnávaných bodov [14]	38
Obrázok 18: BRISK scale-space a detekcia významných bodov [16]	39
Obrázok 19: Vzorkovací obrazec BRISK [16]	40
Obrázok 20: Ukážka výsledku algoritmu BRISK deskriptora na testovacie obrázky	41
Obrázok 21: Vzorkovacia mriežka pre FREAK [18]	44
Obrázok 22: Ukážka výsledku algoritmu FREAK deskriptora na testovacie obrázky...	45
Obrázok 23: Príklad snímkov, ktoré boli z galérie vynechané. LH: príliš malá ŠPZ, PH - časť ŠPZ neviditeľná, LD - fotené z perspektívy + starý typ ŠPZ, PD - starý typ ŠPZ (len MSER).....	48
Obrázok 24: Príklad ŠPZ pre účely DP	49
Obrázok 25: Príklad detegovaných oblastí v rámci ŠPZ po eliminácii	50
Obrázok 26: ilustrácia vyhľadávania dvojíc a trojíc elíps u algoritmu lokalizácie ŠPZ pomocou MSER.....	51
Obrázok 27: Ukážka nepomeru vzdialeností medzi znakmi	51
Obrázok 28: Názorné zobrazenie geometrie algoritmu lokalizácie ŠPZ v obraze pomocou oblastí MSER	52
Obrázok 29: Vývojový diagram algoritmu vyhľadávania kandidátov na ŠPZ pomocou detektora MSER.....	53
Obrázok 30: Amplitúdová projekcia ŠPZ do osy x	54
Obrázok 31: Príklady ŠPZ s problémovým testovaním.....	55

Obrázok 32: Vývojový diagram testovania kandidátov na potenciálnu ŠPZ	56
Obrázok 33: Vľavo ŠPZ z originálneho grayscale obrázku, vpravo kandidát spracovaný prahovaním, rozdelený na popredie a pozadie.....	57
Obrázok 34: Príklady ŠPZ kde algoritmus lokalizácie pomocou MSER zlyháva.....	58
Obrázok 35: Príklad neštandardne osvetlenej ŠPZ kde nie je možné detektorom MSER detegovať všetky znaky	59
Obrázok 36: Požadovaný výsledok algoritmu lokalizácie ŠPZ pomocou 4 rohových bodov	60
Obrázok 37: Ukážka kandidátov pre jednotlivé rohy ŠPZ: červené body reprezentujú detegované významné body. V prvom rade sú zobrazení kandidáti na ľavý horný a pravý horný roh, v spodnom rade kandidáti na ľavý dolný a pravý dolný roh ŠPZ....	63
Obrázok 38: Názorné zobrazenie geometrie algoritmu lokalizácie ŠPZ v obraze pomocou 4 rohových bodov	64
Obrázok 39: Príklady snímok kde zlyháva lokalizácia ŠPZ pomocou Harrisovho detektora.....	65
Obrázok 40: Príklady snímok, kde je detegované množstvo falošných kandidátov, ktorí úspešne prejdú testom na vlastnosti ŠPZ.....	66
Obrázok 41: Požadovaný výsledok pri lokalizácii ŠPZ pomocou deskriptora BRISK..	69
Obrázok 42: Príklad snímku kde nie je detegovaný jeden z rohov ŠPZ.....	70
Obrázok 43: Ukážka vysokej odozvy deskriptora BRISK na trávnatú plochu. Detektor pracuje na princípe detektora FAST a má vysokú odozvu v značne štruktúrovaných oblastiach kde sa nachádza množstvo ostrých rohov.....	71
Obrázok 44: Príklad nepomeru počtu získaných BRISK bodov pre dve snímky z rovnakej galérie získaných za rovnakých podmienok. Vo vrchom rade snímky bez predspracovania, v rade spodnom snímky filtrované Gaussovským filtrom (veľkosť 5x5 a smerodajná odchýlka $\sigma = 1$).....	72
Obrázok 45: Názorné zobrazenie geometrie algoritmu lokalizácie ŠPZ v obraze pomocou SURF oblastí.....	73
Obrázok 46: Ukážka požadovaného výsledku pri lokalizácii ŠPZ pomocou deskriptora SUFR (zelené oblasti).....	74
Obrázok 47: Príklady nepravidelného rozloženia SURF významných bodov v okolí znakov 1 a 4	74
Obrázok 48: Príklad snímku ŠPZ s chýbajúcim SURF bodom v pravom dolnom rohu. Chýbajúci bod je najčastejšou príčinou zlyhania algoritmu	75
Obrázok 49: Odozva HoG na ľavý horný roh ŠPZ a kandidáti na rohový bod.....	77
Obrázok 50: Graf zobrazujúci porovnanie úspešnosti TP lokalizácie ŠPZ v obraze na jednotlivých galériách.....	79

ZOZNAM TABULIEK

Tabuľka 1: Úspešnosť a parametre algoritmu využívajúceho MSER	59
Tabuľka 2: Úspešnosť a parametre algoritmu využívajúceho Harrisov detektor	66
Tabuľka 3: Úspešnosť a parametre algoritmu využívajúceho detektor Shi-Tomasi	68
Tabuľka 4: Úspešnosť a parametre algoritmu využívajúceho FAST detektor	69
Tabuľka 5: Úspešnosť a parametre algoritmu využívajúceho BRISK deskriptor	71
Tabuľka 6: Úspešnosť a parametre algoritmu využívajúceho SURF deskriptor.....	75
Tabuľka 7: Tabuľka zobrazujúca získané výsledky pre dané galérie použitím rôznych detektorov a deskriptorov. Výsledky sú rozdelené na „true positive“ teda úspešné lokalizovanie ŠPZ a „false positive“ teda falošné lokalizovanie ŠPZ. Výsledky označené zeleným písmom indikujú najlepší výsledok na danej galérii, naopak výsledky označené červeným písmom indikujú najhoršie výsledky. Je možné pozorovať najlepšie výsledky pre detektor MSER naopak najhoršie pre SURF a BRISK.....	78

ÚVOD

Spracovanie obrazu je veľmi dôležitou aplikačnou oblasťou techniky ako takej a s rozvojom výpočtovej techniky a zvyšujúcou rýchlosťou spracovania informácií nachádza pravidelné uplatnenie v komplexnejších úlohách. Aplikácie počítačového videnia sa čím ďalej tým viac využívajú v zdravotníctve, priemysle, doprave a podobne. Napríklad predstava plne autonómneho vozidla by bez počítačového videnia bola absolútne nereálna. Tieto aplikácie sa postupne snažia priblížiť k ultimátnemu cieľu, ktorým je takmer dokonalé napodobenie ľudského zraku. I keď s pokrokom vedy a techniky sa tomu cieľu aplikácie počítačového videnia blížia, stále sú veľmi ďaleko od jeho dosiahnutia jednak čo sa týka rýchlosti aj spoľahlivosti.

V ľudskom organizme slúži ľudské oko k získaniu informácie z prostredia, ktorá následne putuje do mozgu, kde je spracovaná a môže ovplyvňovať správanie a činnosti človeka. V počítačovom videní nahrádza funkciu oka kamera, konkrétne senzor, ktorý z okolia získava informáciu o jase. O spracovanie tejto informácie sa následne stará procesor počítača s perifériami, poprípade procesor v telefóne alebo podobnom zariadení. Tento proces pozostáva z viacerých krokov, ktoré názorne ilustruje schéma na Obrázok 1.

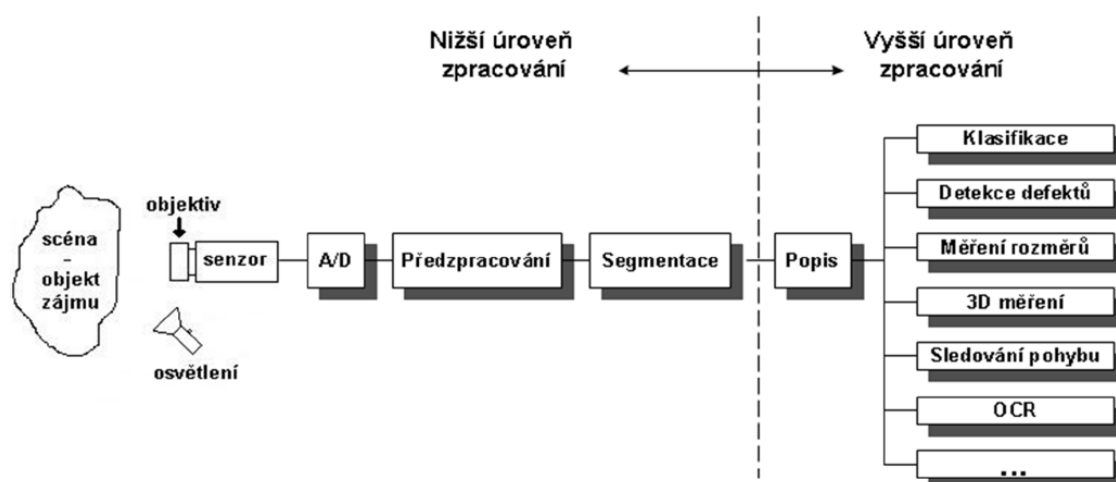
Predstavme si prípad, že máme fotoaparát, ktorý nepodporuje tvorbu panoramatických snímok, ale napriek tomu sme sa rozhodli, že si daný výhľad vyfotíme na viacero fotiek a zjednodušene povedané si ich následnej poskladáme dohromady podľa prelínajúcich sa častí a upravíme (napríklad orežeme) do výslednej podoby. V prípade, že by sme daný problém riešili s vytlačenými fotografiami, spolupráca medzi ľudským zrakom a mozgom nám umožní nájsť body, ktoré spolu korešpondujú a panorámu by sme dokázali vyskladať. Iným prípadom, kde je potrebné hľadať rovnaké body v dvoch (alebo viacerých) obrazoch je skladanie 3D modelu objektu, napríklad budovy. Predstavme si, že sa pokúsime vyfotiť budovu spredu a z uhla 45° oproti pôvodnej polohe za účelom vyfotenia bočnej strany budovy. Rovnako ako v prvom prípade potrebujeme nájsť body, ktoré sú zhodné na obidvoch snímkach. V prípade človeka tieto body ľahko nájdeme, napríklad roh okna odfoteného spredu a ten istý roh pri fotení budovy pod uhlom a podobne. Ako ale v týchto prípadoch funguje výpočtová technika? Počítače v posledných rokoch významne napredujú, no stále nie sú schopné v oblasti počítačového videnia nahradiť ľudský zrak a spracovanie obrazu v mozgu.

V prípade počítačového spracovania úloh popísaných vyššie je potrebné prejsť niekoľkými krokmi, ktoré v zjednodušenom modeli môžu predstavovať nájdenie významných bodov, ich následný matematický popis a v neposlednom rade spájanie (matching) obrazov na základe zhodných významných bodov. A práve detekcia a popis významných bodov sú cieľom tejto práce.

Práca je, čo sa štruktúry týka, rozdelená do troch veľkých tematických celkov. Prvá kapitola predstavuje časť zaoberajúcu sa detektormi oblastí v obraze. Cieľom je

naštudovať problematiku detektorov oblastí v obraze a zostaviť prehľad s príkladmi. V rámci tejto kapitoly budú predstavené nasledujúce detektory: Moravcov detektor, Harrisov detektor, MSER a FAST. Poradie v ktorom budú detektory predstavené odpovedá historickému vývoju. U každého z nich budú uvedené obrázky na ktorých bude prezentované ako detektor funguje na reálnom snímku týkajúcom sa následnej experimentálnej úlohy. Z hľadiska modelu spracovania a analýzy obrazu zobrazenej na Obrázok 1 možno detektory zahrnúť približne na koniec nižšej úrovne spracovania do oblasti segmentácie.

Druhá kapitola sa venuje deskriptorom oblastí v obraze a predstavuje vyššiu úroveň spracovania. V Obrázok 1 tejto oblasti odpovedá časť s názvom „popis“. O ten sa v skutočnosti pri deskriptoroch jedná. Úloha spočíva v matematickom popise nájdených významných bodov tak, aby bol tento popis dostatočne robustný, stabilný a opakovateľný v kompromise s náročnosťou výpočtu. Cieľom tejto kapitoly je naštudovať problematiku deskriptorov významných oblastí v obraze a opäť zostaviť prehľad s príkladmi. U väčšiny deskriptorov bude opäť prezentovaný ich výstup na snímke následnej experimentálnej úlohy aby si čitateľ vedel predstaviť ako daný deskriptor funguje na celom obraze a nie len na oblasti, ktorá bude zaujímavá pre experimentálnu časť. Vo vývoji problematiky deskriptorov je možné zaznamenať, že jednotlivé metódy na seba postupne nadväzujú. V hierarchickom slede budú postupne predstavené nasledujúce deskriptory: HoG, SIFT, SURF, BRIEF, BRISK, ORB a FREAK. Niektoré z týchto algoritmov v sebe zahŕňajú aj vlastný detektor, ale sú známe hlavne ako deskriptory a preto sú uvedené v tejto kapitole. Na túto skutočnosť bude u daných zástupcov upozornené v príslušných kapitolách.



Obrázok 1: Reťazec spracovania obrazu [21]

Finálnou a jadrovou časťou práce bude následne tretia kapitola, ktorej cieľom bude otestovať vybrané z predstavených detektorov a deskriptorov na demonštračnej úlohe. Danou úlohou bola zvolená úloha lokalizácie ŠPZ v obraze, ktorá je stále pomerne

živou témou a poskytuje potenciál pre kvalitatívne zhodnotenie daných detektorov a deskriptorov. Budú vytvorené a popísané algoritmy, ktoré budú využívať výstup jednotlivých metód a využívať tento potenciál k nájdeniu ŠPZ v obraze. Výsledkom tejto časti bude kvalitatívne zhodnotenie úspešnosti jednotlivých algoritmov na danej úlohe. Analyzovaná bude ich robustnosť a vyzdvihnuté budú ich silné stránky pre riešenie danej úlohy, ako aj situácie kde tieto metódy zlyhávajú. Dôležité je v tomto bode poznamenať, že cieľom práce nie je dosiahnuť 100% úspešnosť lokalizácie, ale vyskúšať dostupné prostriedky na náročnejších snímkach a odhaliť ich silné a slabé stránky. Cieľom rovnako nie je získanie veľmi rýchlych aplikácií, poprípade aplikácií blížiacich sa spracovávaníu k reálnom čase. Výsledky jednotlivých detektorov a deskriptorov budú uvedené v tabuľkovej forme a sprevádzané príkladmi ilustrujúcimi vzniknuté situácie.

1 DETEKTORY VÝZNAMNÝCH BODOV A OBLASTÍ V OBRAZE

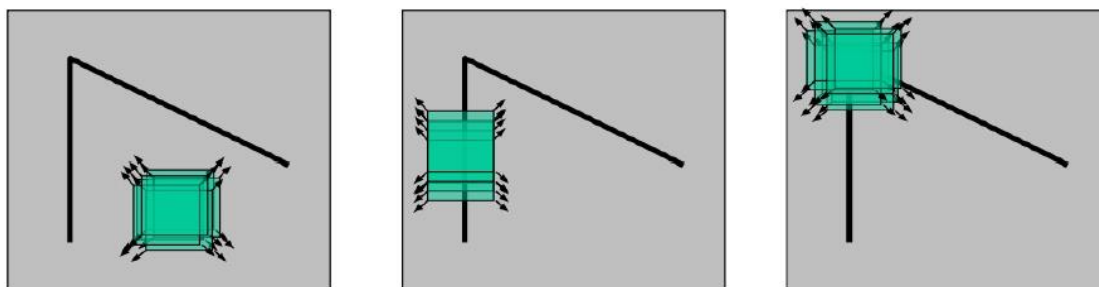
V 1. kapitole bude pojednávané o možnostiach detekcie významných bodov v obraze. Nachádza sa v nej stručný úvod do problematiky významných oblastí, pričom bude takáto významná oblasť charakterizovaná a ukázaný základný teoretický princíp vyhľadávania takej oblasti. Následne budú predstavení najznámejší predstavitelia detektorov oblastí v obraze. Medzi nich patrí napríklad Moravcov detektor alebo Harrisov detektor. Hlbšie sa dostaneme do problematiky komplexnejších a výkonnejších detektorov ako MSER a FAST. Práve posledné dva algoritmy sú v dnešnej dobe pre svoje vlastnosti a kvalitu výstupu relatívne často používané. Čo sa štruktúry týka, bude každý detektor popísaný v štyroch podkapitolách, v ktorých bude uvedený úvod do danej problematiky, funkčný a matematický popis danej metódy a budú zhrnuté výhody a nevýhody daného prístupu. V poslednej podkapitole bude u väčšiny detektorov prezentovaná ukážka týchto detektorov získaná v programovom prostredí MATLAB.

1.1 Významná oblasť

Táto kapitola čerpá informácie zo zdrojov [3] a [4].

Ako teda ale významná oblasť, ktorú hľadáme vyzerá? Tieto oblasti by mali byť voči ostatným oblastiam nejakým spôsobom extraodínárne, čo v konečnom dôsledku vďaka ich matematickému popisu uľahčí ich detekciu v inom obraze. Pre tieto oblasti je nevyhnutná ich stabilita, vďaka ktorej získavame dobrú lokalizovateľnosť v obraze a opakovateľnosť aj pri pôsobení geometrických zmien.

Medzi dve najdôležitejšie významné oblasti v obraze patria hrany a rohy. Detekciu hrán a rohov názorne ilustruje nasledujúci obrázok.



Obrázok 2: a) oblasť bez textúry, b) detekcia hrany, c) detekcia rohu [5]

V ľavom obrázku je možné vidieť, že pri pohybe s vyhľadávacím oknom v akejkoľvek smere nedochádza k zmene intenzity obrazovej funkcie. To je typické pre oblasti bez textúry, akou je napríklad obloha, kde v podstate nie je možné nájsť dobre

opakovateľnú významnú oblasť. Akési zlepšenie poskytuje hrana, ktorej detekcia je zobrazená na strednom obrázku. V prípade pohybu vyhľadávacieho okna dochádza k zmene intenzity v jednom smere. Na základe toho je možné detegovať hranu, problémom je však nájdenie polohy v rámci celej hrany v inom obraze. Existuje ale mnoho účinných detektorov hrán, ktoré sa využívajú k oddeleniu objektu od pozadia, teda segmentácii v obraze. Ako tretia možnosť je v pravej časti uvedená detekcia bodu, konkrétne rohu, spojnice dvoch hrán. Charakteristické pre roh je, že pri pohybe vyhľadávacím oknom dochádza k zmene intenzity obrazovej funkcie vo všetkých smeroch a pri vhodnom matematickom popise oblasti okolo daného bodu je zaistená vysoká opakovateľnosť detekcie daného bodu v inom obraze. Takýto bod budeme ďalej nazývať významný rohový bod.

Zhrňme si teda základné charakteristiky významných bodov:

- Majú jasnú a matematicky dobre podloženú definíciu
- Majú jasne definovanú pozíciu v obrazovom priestore
- Lokálna štruktúra okolo významného bodu je bohatá na informácie vhodné pre ďalšie spracovanie
- Sú stabilné z hľadiska pôsobenia lokálnych a globálnych deformácií a teda zaisťujú vysoký stupeň opakovateľnosti pri hľadaní v inom obraze

1.2 Možnosti detekcie

Logicky najjednoduchšou možnosťou ako nájsť istý bod v inom obraze je zobrať daný bod, popísať ho pomocou okolitých bodov a vyhľadať v obraze, v ktorom daný bod potrebujeme nájsť. Tento postup je ale maximálne neefektívny a vyžaduje enormné nároky na spracovanie. Navyše neposkytuje absolútne žiadnu toleranciu voči akýmkoľvek deformáciám. Ďalšou z jednoduchších variant je aplikácia konvolučného jadra o veľkosti 3 x 3 a následné získanie bodov prahovaním. Tvar konvolučného jadra je uvedený nižšie.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (1)$$

V ďalších kapitolách budú uvedené sofistikovanejšie detektory oblastí v obraze.

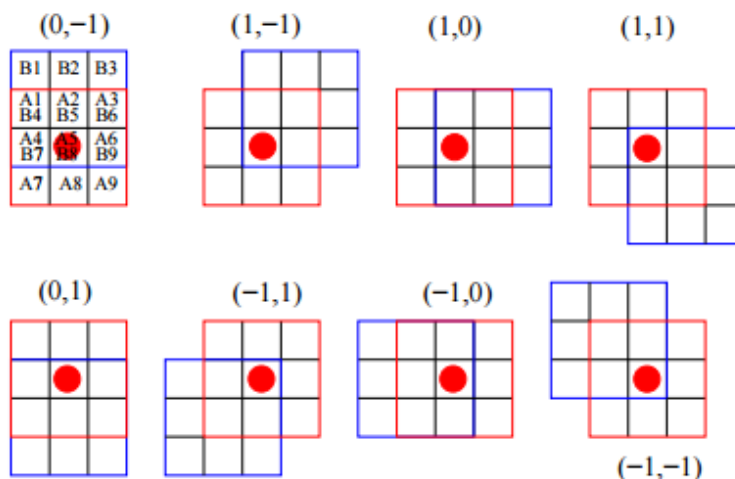
1.3 Moravcov detektor (operátor)

Kapitola o Moravcovom detektore bola vytvorená s pomocou zdrojov [4],[6],[7],[8].

1.3.1 Úvod

Jedná sa o snád' najstarší detektor významných oblastí v obraze na vyššej úrovni. Bol vyvinutý rakúskym vedcom Hansom Moravcom v roku 1977 keď potreboval detegovať objekty v okolí vozidla Stanford Cart. Moravcov detektor je považovaný za rohový

detektor a funguje na podobnom princípe aký bol popísaný v kapitole 1.1. Moravec, rovnako ako v našom výklade používa vyhľadávacie okno malých rozmerov, najčastejšie 3 x 3, 5 x 5 alebo 7 x 7 pixelov okolo vyšetřovaného bodu. Následne sa daným oknom pohybuje vo všetkých ôsmich smeroch o jeden pixel ako zobrazuje nasledujúci obrázok, kde červené okno ukazuje pôvodné okno okolo vyšetřovaného bodu a okno modré zobrazuje posunuté okno v jednom z ôsmich smerov. Červený kruh reprezentuje nami vyšetřovaný bod.



Obrázok 3: Pohyb vyhľadávacieho okna pri Moravcovom operátore [6]

Operátor vo svojom princípe počíta rozptyl intenzity jasu pri posúvaní okna v rôznych smeroch a z toho vyplývajú tri prípady.

- Pri minimálnej zmene sa jedná o plochú oblasť bez veľkých zmien intenzity
- Veľká zmena v jednom smere, ale žiadna alebo takmer nulová zmena v smere kolmom na daný smer značí hranu
- Všetky smery vykazujú veľkú zmenu – detegujeme roh

1.3.2 Funkčný a matematický popis

Rozptyl intenzity jasu pre posun v danom smere sa vypočíta ako suma štvorcov rozdielu odpovedajúcich si pixelov dvoch okien, okna pôvodného s centrom v nami vyšetřovanom bode a okna posunutého o jeden pixel v jednom z 8 smerov. Tento vzťah je vyjadrený rovnicou nižšie. Rozptyl intenzity v nami vyšetřovanom bode je braný ako minimum z ôsmich vyšetřovaných smerov a následne algoritmom prahovania určíme detegované body.

$$E(u, v) = \sum_{(x,y) \in O} (I(x, y) - I(x + u, y + v))^2 \quad (2)$$

kde O je nami vyšetřovaný bod a funkcia $I(x, y)$ predstavuje hodnoty jasu pixelov v obrázku.

Existuje ale aj druhý variant Moravcovho operátora. Táto verzia je rovnako jednoduchá a počíta rozdiely hodnôt jasu medzi nami vyšetrovaným bodom a v najbližšom 8-okolí. Absolútne hodnoty sa následne sčítajú a výsledná hodnota sa vydolí 8. Výpočet reprezentuje nasledujúca rovnica:

$$g(i,j) = \frac{1}{8} \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} |f(i,j) - f(m,n)| \quad (3)$$

Kde funkcia $f(i,j)$ predstavuje hodnoty jasu pixelov v obrázku.

1.3.3 Výhody / Nevýhody

Moravcov operátor ma kvôli svojej jednoduchosti množstvo nevýhod. Keďže počítame zmenu intenzity v ôsmich diskretných smeroch nie je tento operátor rotačne invariantný. Praktickým dopadom je nízka opakovateľnosť pri pootočení o iné hodnoty, napríklad o 30°. Ďalšou nevýhodou je použitie štvorcového popr. obdĺžnikového okna. Vylepšením by bolo použitie „diskrétného“ kruhového okna, ktorí znižuje euklidovskú vzdialenosť uvažovaných okolitých pixelov a teda do výpočtu zahrňuje relevantnejšie údaje. Nevýhodou je rovnako kladenie rovnakej váhy na všetky pixely. To v prípade okna 3x3 neovplyvňuje výpočet, ale pri použití väčšieho okna je vhodné klásť bližším susedom väčšiu váhu. Jeho ďalšími nevhodnými vlastnosťami sú vysoká odozva na hrany a šum. Na druhej strane vďaka relatívnej jednoduchosti sa tento operátor vyznačuje jednoduchou implementáciou a rýchlosťou spracovania.

1.3.4 Ukážka aplikácie

Pre testovacie účely Moravcovho operátora bola vytvorená jednoduchá aplikácia v MATLABe, ktorá detekuje významné body.



Obrázok 4: Ukážka výsledku algoritmu Moravcovho detektora na testovacie obrázky

1.4 Harrisov detektor (operátor)

Pri tvorbe kapitoly o Harrisovom detektore bolo vychádzané zo zdrojov [8],[9],[22].

1.4.1 Úvod

Harrisov detektor, rovnako označovaný aj ako „Plessey operátor“ bol vyvinutý Chrisom Harrisom a Mikom Stephensonom v roku 1988. Autori sa zaoberali úlohou nájdenia a priradenia rovnakých bodov v dvoch rozdielnych obrazoch získaných jednou pohybujúcou sa kamerou. Autori naviazali na už publikovaný a používaný Moravcov operátor a pokúsili sa o jeho vylepšenie, s cieľom vytvoriť detektor schopný detegovať hrany aj rohy. Hlavným nedostatkom Moravcovho operátora bola závislosť na rotácii a nízka odolnosť voči šumu. To bolo spôsobené použitím obdĺžnikového okna, ktoré je v tomto prípade nahradené oknom kruhovým s Gaussovským rozložením vnútorných hodnôt prikladajúcim najväčšiu váhu hodnotám v blízkosti vyšetřovaného bodu. K odstránení neizotropných odoziev je použitá auto-korelačná funkcia (rovnic 4) schopná merať zmenu intenzity vo všetkých smeroch. Tento operátor napriek svojim obmedzeniam, ktoré budú opísané o dve kapitoly nižšie, je aj v dnešnej dobe pomerne hodne používaný.

1.4.2 Funkčný a matematický popis

Základom odvodenia operátoru je použitie auto-korelačnej funkcie:

$$c(N_0) = \min \sum_{(u,v) \in W} w(u,v) * [I(u,v) - I(u + \Delta x, v + \Delta y)]^2 \quad (4)$$

kde:

W je vyšetrovacie okienko okolo nami vyšetrovaného bodu,

(x,y) sú súradnice v rámci okienka,

$w(x,y)$ je váhové funkcia,

$I(x,y)$ je jasová funkcia a

$(\Delta x, \Delta y)$ označujú lokálny posun okienka.

Váhová funkcia môže byť napríklad Gaussovská, alebo v jednoduchšom prípade konštantná, s hodnotou rovnou 1 v prípade, že súradnice sú v rámci nášho vyšetrovaného okna a 0 v prípade, že nie sú.

V ďalšom kroku sa pokúsime aproximovať výraz $I(u+\Delta x, v+\Delta y)$, ktorý značí intenzitu jasovej funkcie na posunutých súradniciach. Pomocou prvých dvoch členov Taylorovho rozvoja približne dostávame:

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + \nabla I(x, y)^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (5)$$

A po následnom dosadení aproximácie do rovnice 4 dostávame:

$$\begin{aligned} c(N_0) &\approx \min \sum_{(u,v) \in W} [\Delta x, \Delta y] \nabla I(x, y) \nabla I(x, y)^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \min [\Delta x, \Delta y] \left(\sum_{(x,y) \in N_0} \nabla I(x, y) \nabla I(x, y)^T \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \min [\Delta x, \Delta y] \mathbf{A} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned} \quad (6)$$

kde sa nám objavuje matica $\mathbf{A}$. Matica $\mathbf{A}$ je pozitívne semidefinitná o veľkosti 2×2 a platí pre ňu:

$$\begin{aligned} \mathbf{A} &= \sum_{(x,y) \in N_0} \nabla I(x, y) \nabla I(x, y)^T \\ &= \sum_{(x,y) \in N_0} \left[\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right] \begin{bmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{bmatrix} \end{aligned} \quad (7)$$

Nakoľko je matica $\mathbf{A}$ symetrická, ďalej platí

$$\min [\Delta x, \Delta y] \mathbf{A} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \|\Delta x, \Delta y\|^2 \lambda_{\min} \quad (8)$$

Kde $\lambda_{\min}$ je najmenšie vlastné číslo matice $\mathbf{A}$.

Výpočet gradientu (derivácií) je v diskretnom obraze možno nahradiť výpočtom symetrických diferencií podľa:

$$\nabla I(x, y) = \begin{bmatrix} I(x+1, y) - I(x-1, y) \\ I(x, y+1) - I(x, y-1) \end{bmatrix} \quad (9)$$

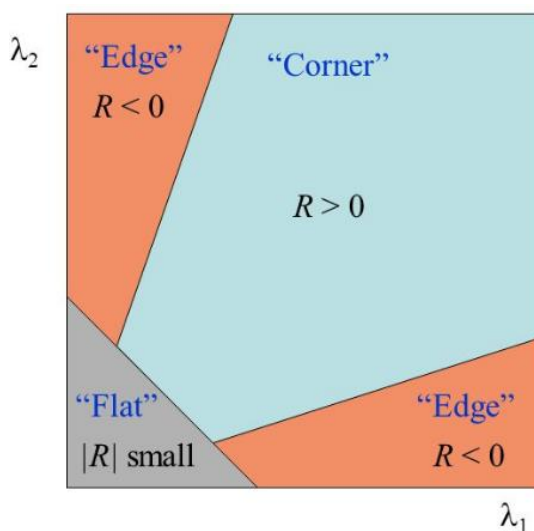
Po vypočítaní symetrickej matice A nasleduje analýza. Existujú tri možnosti vyhodnotenia veľkosti gradientu a matice A .

- V prípade, že má jasová funkcia konštantný charakter v okolí N_0 , potom gradient je blízky nulovému vektoru a matica A je blízka nulovej matici. Tým pádom sa obidve vlastné čísla matice A blížia k nule.
- V prípade, že je jedno vlastné číslo matice malé a druhé veľké, indikuje to detekciu hrany. Malé vlastné číslo indikuje posun po hrane, väčšie vlastné číslo posun kolmo na hranu.
- V prípade, že je gradient rôzny vo všetkých smeroch, majú obe vlastné čísla vysokú hodnotu čo indikuje detekciu rohu

K získaniu významných bodov sa ale v praxi nepoužíva výpočet vlastných čísel matice A . Namiesto toho Harris a Stephens navrhli výpočet Harrisovej funkcie ktorá má nasledujúci tvar:

$$R = \det(A) - k * tr^2(A) \quad (10)$$

Operácia tr (trace) je nazývaná stopa matice a vypočítame ju ako súčet prvkov na hlavnej diagonále matice A . Konštanta k je empirická konštanta stanovená experimentálne a podľa [8] dostávame najlepšie výsledky pre hodnoty k medzi 0,04 a 0,06. Po výpočte Harrisovej funkcie pre všetky body obrázku vyberieme lokálne maximá, ktoré prehlásime za významné body. Jednou z možností je použitie tzv. algoritmu non-maxima suppression. Klasifikáciu bodov podľa vlastných čísel matice a hodnoty Harrisovej funkcie ilustračne zobrazuje nasledujúci obrázok.



Obrázok 5: Klasifikácia významných oblastí podľa Harrisa [5]

1.4.3 Výhody / Nevýhody

Harrisov operátor odstránil niektoré nevýhody a limitácie Moravcovho operátora. Asi najväčšou daňou za to je zvýšená náročnosť na výpočet a teda dlhší čas spracovania. Zo všeobecnosti ja ďalej známe, že derivácia a jej diskretný ekvivalent – diferenciacia, zvyšujú šum. V Harrisovom operátore sa využíva gradient, ktorý opäť využíva deriváciu (diferenciaciu) čo z neho opäť činí detektor citlivý na šum. To je možné čiastočne odstrániť prvotnou aplikáciou filtra dolnej priepuste, ktorý obraz rozostří a zníži tým šum, ale na druhej strane znižuje hodnoty gradientu. Pri použití väčšieho Gaussovského okna sa síce výsledky zlepšia, pretože sa zníži veľkosť vplyvu zašumených pixelov za cenu zvýšenia výpočtovej náročnosti.

Veľkou nevýhodou Harrisovho detektora je jeho závislosť na mierke alebo teda úrovni zoomu s akým sa na daný obrázok pozeráme. Keď si predstavíme roh, ktorý dokáže celé vyšetrovacie okienko pokryť bude tento roh správne detegovaný ako roh. V prípade, že sa ale na daný roh pozeráme pri väčšom zoome, môže sa do vyšetrovacieho okienka rovnakej veľkosti ako v predchádzajúcom úsudku dostať len časť daného objektu, ktorá bude vyhodnotená ako rovná čiara, teda hrana. Touto problematikou sa v [22] zaoberal Krystian Mikolajczyk a navrhol úpravu, pomocou ktorej je možné získať detektor nezávislý na zmene mierky. Tento upravený detektor nesie meno Harris-Laplace detektor a spočíva vo vyhľadávaní maxima vo vytvorenom scale-space. Koncept scale-space a jeho tvorby je vysvetlený napríklad v kapitole 2.2 o deskriptore SIFT. Jedná sa o vytvorenie akéhosi priestoru, v ktorom sa daný obrázok nachádza s rôznou mierkou. Princípom je vyhľadávanie významných bodov (rohov) skrz tento priestor a následne je vybratý bod, ktorý odpovedá maximu daného významného bodu v rámci úrovni scale-space. S pozíciou tohto bodu je uchovaná aj miera obrázku, v ktorom bol ako maximum detegovaný. Ako nedostatok sa môže javiť fakt, že pri nevhodne zvolenom rozdieli mierok medzi danými vrstvami scale-space nemusí byť skutočné maximum detegované. S ohľadom na danú skutočnosť je potrebné vhodne zvoliť krok medzi danými úrovňami mierky, poprípade zvoliť prah v porovnaní s ktorým sú dané body označené ako významné alebo sa s nimi v opačnom prípade nepočíta.

1.4.4 Ukážka aplikácie

Za účelom prezentácie výsledkov Harrisovho detektora bola použitá funkcia MATLABu *detectHarrisFeatures*. Ukážka funkčného kódu:

```
harris_real = detectHarrisFeatures(I_real_gray);  
figure(1)  
imshow(I_real);  
hold on  
plot(harris_real);
```



Obrázok 6: Ukážka výsledku algoritmu Harrisovho detektora na testovacom obrázku

1.5 MSER

1.5.1 Úvod

Tento detektor bol publikovaný v práci z ČVUT v Prahe a University of Guildford v Británii s názvom Robust Wide Baseline Stereo from Maximally Stable Extremal Regions [15], z ktorej boli pre túto kapitolu čerpané informácie. Skratka MSER pochádza z anglického Maximally Stable Extremal Regions, čo by mohlo byť voľne preložené ako maximálne stabilné extrémne oblasti.

Metódy, o ktorých sme sa bavili doteraz boli založené na detekcii významných bodov. MSER zavádza do problematiky iný prístup a hľadá v obraze významné oblasti (v angličtine blobs). Tieto oblasti by ale mali vykazovať rovnaké vlastnosti ako významné body, konkrétne hlavne opakovateľnosť pri zmene jasů, natočenia a optimálne aj mierky obrazu. Algoritmus, ktorý si popíšeme deteguje takéto oblasti a vyberá z nich práve tie maximálne stabilné ako napovedá jeho názov.

1.5.2 Funkčný a matematický popis

V tejto časti uvedieme niekoľko definícií spojených s MSER a následne si slovne popíšeme algoritmus detekcie.

Obraz I je mapovaním $I: D \subset \mathbb{Z}^2 \rightarrow S$. Extrémne regióny sú dobre definovateľné ak:

- 1) S je plne usporiadané tzn. reflexívny, nesymetrický a tranzitívne binárny vzťah $\leq$ existuje. Množina S môže byť uvažovaná ako $S=\{0,1,...,255\}$ pre čierno-biely obraz, ale metóda môže byť definovaná aj na reálnych číslach.
- 2) Susedstvo je v obraze definované ako $A \subset D \times D$. Môže byť definované napríklad 4-susedstvo kde $p,q \in D$ sú susedné (pAq) práve vtedy keď

$$\sum_{i=1}^d |p_i - q_i| \leq 1 \quad (11)$$

Oblasť Q je súvislá podmnožina D , tzn. že pre každú kombináciu $p,q \in D$ existuje postupnosť $p, a_1, a_2, \dots, a_n, q$ a $pAa_1, a_1Aa_2, \dots, a_nAq$.

(Vonkajšia) hranica oblasti: $\partial Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$, tzn. že hranica ∂Q oblasti Q je množina bodov, ktoré sú susedné s aspoň jedným pixelom z Q , ale nepatria do oblasti Q .

Extrémna oblasť: $Q \subset D$ je oblasť taká, že pre všetky $p \in Q, q \in \partial Q : I(p) > I(q)$ alebo $I(p) < I(q)$. V prvom prípade sa jedná o oblasť s maximálnou intenzitou jasu, v druhom prípade o oblasť s minimálnou intenzitou jasu.

Maximálne stabilná extrémna oblasť (MSER): Nech $Q_1, \dots, Q_{i-1}, Q_i, \dots$ je postupnosť vnorených extrémnych oblastí tak, že $Q_i \subset Q_{i+1}$. Extrémna oblasť Q_i je maximálne stabilná práve vtedy keď $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}|/|Q_i|$ má lokálne minimum v i . $\Delta \in S$ je parametrom metódy MSER.

Ako teda ale prakticky metóda funguje? Tento princíp je založený na prahovaní rôznymi prahmi. Na úvod si predstavme všetky možné hodnoty prahu pre šedo tónový obraz. Hodnoty intenzity, ktoré sú pod danou hodnotou prahu označíme čiernou a hodnoty nad prahovou hodnotou bielou farbou. Ak budeme postupne obraz týmito hodnotami od najnižšej prahovať dostaneme na začiatku čisto biely obraz nakoľko všetky hodnoty intenzity v obraze sú väčšie ako nami zvolený absolútne minimálny prah. Postupne sa začnú objavovať čierne body a zhluky čiernych bodov až sa dostaneme k plne čiernemu obrazu. To je prípad pre maximálnu intenzitu jasu, minimálne oblasti môžu byť získané invertovaním intenzity obrazu I . Následne vyšetříme zhluky čiernych (bielych) bodov ktoré sú pre nás zaujímavé naprieč viacerými hodnotami prahu. Tie z nich ktoré ukazujú stabilitu pre viaceré hodnoty prahu prehlásime za oblasti MSER.

1.5.3 Výhody / nevýhody

Takto získané oblasti v obraze vykazujú nasledujúce vlastnosti:

- sú invariantné voči čiastočným jasovým transformáciám,
- susedstvo bodov zostáva zachované aj pri transformácii obrazu,
- sú stabilné naprieč viacerými hodnotami prahu,
- sú detegované oblasti rôznych veľkostí.

Medzi výhody tohto algoritmu patrí jeho rýchlosť v prípade, že je vhodne implementovaný. Algoritmus dobre funguje v prípade, že obraz obsahuje homogénne regióny s dobre čitateľnými hranicami. Na druhú stranu ale algoritmus nevykazuje dobré výsledky v prípade, že je prítomný šum spôsobený pohybom. Dôležité je spomenúť, že existuje nadstavba Multi-resolution MSER, ktorá poskytuje väčšiu robustnosť pre zmeny mierky a je založená na vyhľadani významných oblastí naprieč celým scale-space priestorom o ktorom sa hovorí napríklad v kapitole 2.2.1 o algoritme SIFT.

1.5.4 Ukážka aplikácie

Ukážka výsledku aplikácie algoritmu MSER bola získaná funkciou *detectMSERFeatures*, ktorá je priamo implementovaná v MATLABe. Výsledky boli získané na základe nasledujúceho kódu:

```
mser_regions_real = detectMSERFeatures(I_real_gray);
figure(1)
imshow(I_real)
hold on
plot(mser_regions_real)
```



Obrázok 7: Ukážka výsledku algoritmu MSER detektora na testovacom obrázku

1.6 FAST

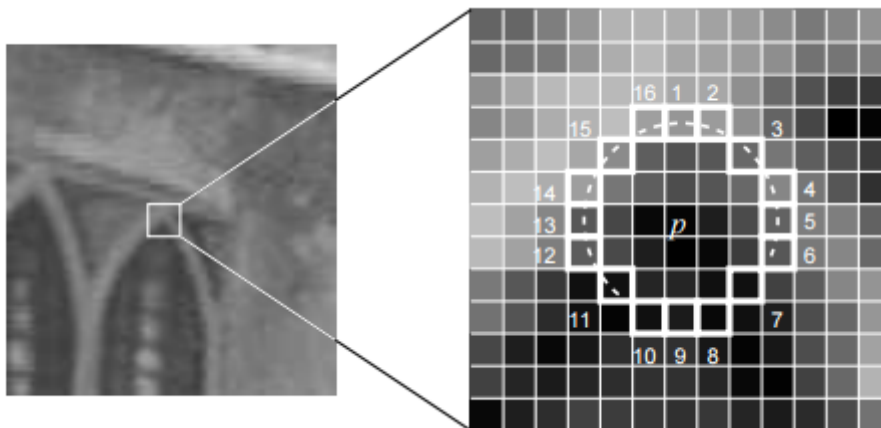
Kapitola 1.6 sa opiera o zdroje [12] a [13].

1.6.1 Úvod

Jedná sa o rohový detektor v obraze vyvinutý Edwardom Rostenom a Tomom Drummondom v roku 2006. Skratka FAST je z angličtiny a znamená Features From Accelerated Segment Test, čo vo voľnom preklade môžeme preložiť ako vlastnosti zo zrýchleného testu segmentu.

1.6.2 Funkčný a matematický popis

Metóda je založená na testovaní hodnôt jasu zvolených pixelov oproti pixelu vyšetrovanému. Táto metóda nevyužíva priamo testovanie oproti bodom susediacim s nami vyšetrovaným pixelom ale porovnanie prebieha oproti bodom ležiacim na kružnici o polomere 3 pixelov so stredom v nami vyšetrovanom bode. Týchto bodov je celkovo 16 a ich číslovanie a polohu zobrazuje nižšie uvedený obrázok. Metódou zisťujeme, či aspoň 12 susediacich pixelov na kružnici má vyššiu hodnotu intenzity ako nami vyšetrovaný bod p (alebo hodnotu vyššiu o nejakú úroveň prahu t) popr. či aspoň 12 susediacich bodov na kružnici má hodnotu intenzity menšiu ako nami vyšetrovaný bod (alebo hodnotu nižšiu o hodnotu prahu t).



Obrázok 8: K vysvetleniu princípu FAST [12]

K urýchleniu metódy sa netestujú pixely v poradí v akom ležia veľa seba ale postupne otestujeme pixely 1,9,5 a 13, nakoľko aby bola splnená podmienka minimálne 12 susedných pixelov pod alebo nad úrovňou jasu bodu p , musia aspoň 3 z týchto 4 testovaných bodov byť nad alebo pod úrovňou tohto jasu. Autor vo svojom článku [13] uvádza, že štatisticky je v priemere potrebné testovať pre daný bod len 3,8 pixelov aby sme dokázali určiť či je daný bod rohom.

1.6.3 Výhody / Nevýhody

FAST detektor je relatívne jednoduchý detektor významných bodov – rohov. Jeho jednoduchosť významne hrá v prospech jeho rýchlosti oproti ostatným detektorom. Na druhú stranu aj pri svojej jednoduchosti vykazuje dosť dobré výsledky čo sa týka opakovateľnosti detekcie významného bodu. Jeho nevýhodou je, že nie je dostatočne robustný voči vysokej úrovni šumu a rovnako, že môže detegovať 1 pixel hrubé hrany pri nevhodnom natočení diskrétného kruhu voči čiare. Ďalšou nevýhodou je závislosť kvality výsledkov na nastavení hodnoty prahu t .

1.6.4 Ukážka aplikácie

Algoritmus FAST je priamo implementovaný v MATLABe a preto bolo možné získať ukážku na testovacom obrázku pomocou funkcie `detectFASTFeatures` pomocou nasledujúceho kódu:

```
fast_points_real = detectFASTFeatures(I_real_gray,  
'MinQuality', 0.001, 'MinContrast', 0.12);  
figure(1);  
imshow(I_real);  
hold on  
plot(fast_points_real)
```



Obrázok 9: Ukážka výsledku algoritmu FAST detektora na testovacie obrázky

2 DESKRIPTORY OBLASTÍ V OBRAZE

Kapitola č.2 je ďalším stupňom v procese spracovania obrazu. Po extrakcii významných bodov a oblastí z obrazu, ktorej sme sa venovali v predchádzajúcej kapitole sa budeme následne sústrediť na matematický popis týchto oblastí, aby získali potrebné vlastnosti ako sú napríklad opakovateľnosť, robustnosť a invariantnosť voči rotácii alebo zmene svetelných podmienok. Postupne budú predstavené deskriptory HoG, SIFT, SURF, BRIEF, BRISK, ORB a FREAK. Väčšina z nich patrí medzi pomerne moderné metódy, často prezentované na konferenciách o počítačovom videní a jeho aplikáciách. Ako bude z popisu jasne vidieť, tieto deskriptory postupne budujú na predchádzajúcich poznatkoch a vylepšujú svojich predchodcov. Čo sa týka štruktúry kapitoly, je analogická so štruktúrou kapitoly o detektoroch a rovnako tak v tejto časti budú prezentované ukážky funkčnosti týchto algoritmov.

2.1 HoG

2.1.1 Úvod [20]

Prvý z prezentovaných deskriptorov je označovaný skratkou HoG, ktorá znamená Histogram of Oriented Gradients, teda voľne preložené ako histogram orientovaných gradientov. Myšlienka vychádza zo skúseností, že vzhľad a tvar objektu môže byť často charakterizovaný lokálnym rozložením gradientov intenzity. Tento deskriptor je dielom Navneeta Dalala a Balla Triggsa, ktorí sa zaoberali jeho použitím v oblasti detekcie ľudí. Mimo klasického princípu HoG predstavili aj jeho vylepšené modifikácie s rôznym priestorovým usporiadaním blokov a buniek, rozdielnym spôsobom výpočtu gradientu alebo normalizácie. Medzi tieto modifikácie patrí napríklad R-HoG, C-HoG alebo EC-HoG.

2.1.2 Funkčný a matematický popis [20]

Základnou myšlienkou algoritmu je rozdelenie obrázku na pomerne malé, navzájom spojené oblasti (nazývané bunky) a pre každú takto vytvorenú bunku je následné vytvorený histogram orientovaných gradientov. Množina týchto histogramov následne tvorí deskriptor obrazu. Za účelom zlepšenia výkonnosti daného deskriptoru sa u HoG vykonáva normalizácia kontrastu. Bunky spájame do väčších oblastí, nazývaných bloky a v rámci týchto blokov počítame mieru intenzity, ktorá následne slúži na normalizáciu v rámci všetkých buniek v danom bloku. V tomto bode je dobré poznamenať, že na rozdiel od deskriptorov, ktoré budú popísané nižšie, deskriptor HoG popisuje celú oblasť, nie len významné body extrahované niektorým z detektorov.

Najznámejšou a najčastejšie používanou metódou výpočtu gradientu je aplikácia derivačnej masky v horizontálnom a vertikálnom smere. To v praxi znamená filtrácie obrazu s filtrami:

$$D_x = [-1 \ 0 \ 1], \quad D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (12)$$

Hodnoty derivácií (diferencií) v horizontálnom a vertikálnom smere získame konvolúciou následnými vzťahmi:

$$I_x = I * D_x, \quad I_y = I * D_y \quad (13)$$

Následne už len spočítame amplitúdu a orientáciu gradientu:

$$|G| = \sqrt{I_x^2 + I_y^2}, \quad \theta = \arctan\left(\frac{I_y}{I_x}\right) \quad (14)$$

Zo získaných údajov môžeme pristúpiť k vytvoreniu histogramu pre každú z buniek. V prípade že sa jedná o absolútnu hodnotu gradientu uvažujeme orientácie v rozmedzí 0 – 180° v prípade, že sledujeme znamienko gradientu je potrebné histogram uvažovať od 0 do 360°. Histogram predstavuje pravidelne rozdelený daný interval. V práci [20] autori získali uspokojujúce výsledky použitím neznamienkového gradientu a rozdelením intervalu do 9 oblastí po 20-tich stupňoch.

2.1.3 Výhody / nevýhody

Histogram of oriented gradients v sebe zahŕňa niekoľko významných výhod oproti iným deskriptorom. Jednou z nich je jeho invariantnosť voči geometrickým a svetelným transformáciám čo vyplýva z jeho činnosti na lokálnych bunkách. Nie je ale odolný voči rotácií objektu čo je značné obmedzenie.

2.1.4 Ukážka aplikácie

Algoritmus HoG je priamo implementovaný v MATLABe a preto bolo možné získať ukážku na testovacích obrázkoch pomocou funkcie `extractHOGFeature` pomocou nasledujúceho kódu:

```
[H_real, visual_real] =  
extractHOGFeatures(I_real, 'CellSize', [5 5]);  
figure(1)  
imshow(I_real);  
hold on;  
plot(visual_real, 'Color', 'red')
```



Obrázok 10: Ukážka výsledku algoritmu HoG detektora na testovacie obrázky

2.2 SIFT

Informácie o deskriptore SIFT boli získané zo zdrojov [7] a [10].

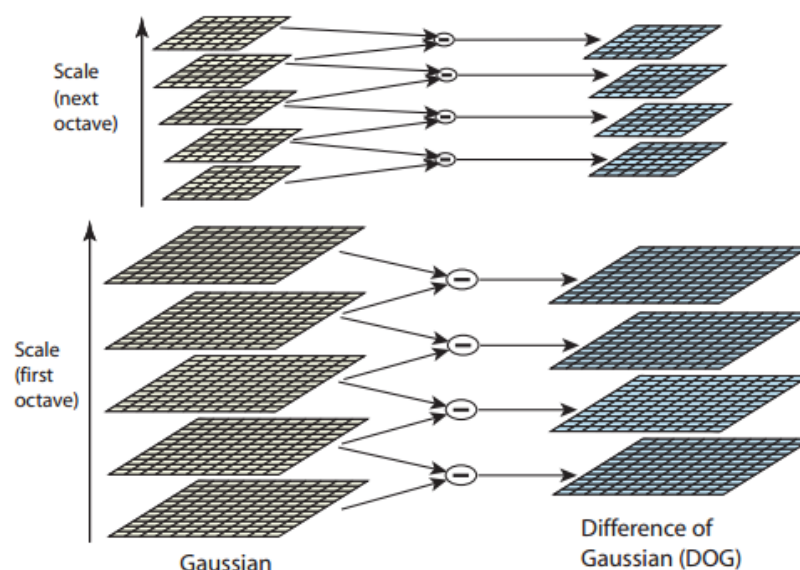
2.2.1 Úvod

Skratka SIFT je odvodená z anglického Scale Invariant Feature Transform a v skratke popisuje princíp danej metódy – transformácia obrázkových dát do súradníc nezávislých na mierke. Tento algoritmus bol vyvinutý Davidom Lowem na University of British Columbia v roku 2004 a patrí medzi jedny z najčastejšie používaných algoritmov v oblasti počítačového videnia. Prakticky sa jedná o akúsi kombináciu detektoru a deskriptoru, kde cieľom je opäť extrakcia významných bodov z obrazu a následný popis rysov daného bodu. SIFT je hlavne známy vďaka svojej kvalitnej implementácii deskriptora a preto je zaradený do tejto kapitoly. Metóda štandardne vychádza z poznatkov predchádzajúcich metód a snaží sa opäť vytvoriť detektor/deskriptor nezávislý na mierke. Ako uvádza sám autor v [10], je SIFT skutočne nezávislý na mierke. Nezávislosť na mierke sa prejavila na komplexnosti riešenia a teda v konečnom dôsledku na náročnosti výpočtu, činiaceho túto aplikáciu ako nevhodnú pre použitie v reálnom čase. Podľa popisu metódy autorom ju v nasledujúcom priblížení môžeme rozdeliť do 4 základných krokov.

2.2.2 Funkčný a matematický popis

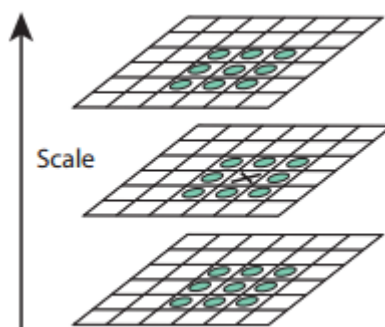
- Scale-space a hľadanie lokálnych extrémov

Nositeľom údajov o mierke pri konverzii obrázku z 3D priestoru do 2D priestoru je frekvenčné spektrum obrazu. Aby sme teda dosiahli požadovanú nezávislosť na mierke musíme detegovať cez viacero mierok. Pre získanie priestorovej frekvencie je potrebné obraz vyhladiť. To sa vykoná viacnásobným aplikovaním Gaussovskej funkcie typu low-pass filter, čím postupne dostávame obrazy o menšej mierke. Následným zostavením týchto vytvorených obrazov o rôznej mierke získame jednu oktávu priestoru. Následne vezmeme dva posledné obrazy z danej oktávy a polovičným vzorkovaním získame základ pre ďalšiu oktávu. Obrazy v daných oktávach sú následne využité k výpočtu rozdielu Gaussianových funkcií (Difference of Gaussian). Daný postup názorne ilustruje nasledujúci obrázok:



Obrázok 11: Princíp tvorby scale-space pomocou DoG výpočtu [10]

Ďalším krokom algoritmu je nájdenie maxima v rámci oktávy. Ten získame postupným porovnaním danej úrovne rozdielu Gaussianových funkcií a priľahlých dvoch úrovní. Dochádza teda k porovnaniu daného pixelu s jeho ôsmimi susedmi v rámci jednej vrstvy a následne s 2 x 9 susedmi z vrstvy nad a pod nami vyšetrovanou vrstvou. To opakujeme pre všetky úrovne okrem prvej a poslednej, pretože pre tieto úrovne neexistuje jeden zo susedov. V prípade, že bol bod detegovaný ako extrém, dôjde k jeho zapamätaniu pre neskoršie využitie. Situácia je opäť znázornená na obrázku:



Obrázok 12: Princíp vyhľadávania lokálneho extrémumu medzi priestormi vytvorenými pomocou DoG [10]

- Lokalizácia významných bodov

Výstupom z predchádzajúceho kroku sú hodnoty získaných lokálnych maxím, ktoré budeme musieť ďalej spracovať aby sme boli schopní lokalizovať významné body a zaistiť stabilitu vybraných bodov. K tomu sa používa metóda vyvinutá M. Brownom, kde sa presná poloha bodu z dôvodov nízkej stability aproximuje pomocou funkcií popisujúcich okolie týchto bodov. Tieto funkcie využívajú Taylorov rozvoj a majú tvar:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (15)$$

kde $x=(x,y,s)^T$ sú súradnice bodu určeného ako lokálne maximum v scale-space v prvom kroku a $D(x)$ je aproximácia rozdielu Gausiánových funkcií. Následne rovnicu položíme rovnú nule a vyjadríme sub-pixelovú a sub-mierkovú pozíciu ako extrém:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (16)$$

Ďalej vyšetrujeme hodnotu x oproti hodnote 0,5. V prípade, že je väčšia v ktorejkoľvek dimenzii, značí to polohu extrému bližšie k susednému bodu. V tomto prípade vyšetovaný bod vymeníme za bod susedný a výpočet opakujeme. V prípade, že je hodnota menšia ako 0,5 je hodnota x pripočítaná k súradniciam skúmaného bodu.

V rámci tohto kroku sa môžeme postarať rovnako o odstránenie bodov lokalizovaných v okolí hrán. Ako je všeobecne známe o DoG, má vysokú odozvu v susedstve hrán z čoho plynie mnoho detegovaných maxím v týchto oblastiach. Tieto body je možné odfiltrovať takzvanou Hessovou maticou. Tá je podobná matici **A** uvedenej v kapitole 1.4.2 o Harrisovom detektore s tým rozdielom, že využívame druhé derivácie. Maticu označíme **H** a bude mať nasledujúci tvar:

$$\mathbf{H} = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \quad (17)$$

Tu sa podobne ako u Harrisovho detektoru snažíme obísť výpočet vlastných čísel matice a pomocou operácií determinantu a stopy matice dostávame:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \quad (18)$$

Kde r je pomer medzi vlastnými číslami matice a pri splnení nerovnice daný bod zachováame. Podľa [10] využíval pri svojich experimentoch David Lowe hodnotu $r=10$.

- Určenie orientácie

Určením orientácie pre nami nájdené významné body zabezpečíme invariantnosť danej metódy voči rotácii. Pre tieto body sa vyberie Gaussovská vyfiltrovaná plocha o najbližšej mierke. Bod v tejto ploche nesie označenie $L(x,y)$ a následne pre daný bod spočítame veľkosť gradientu $m(x,y)$ a orientácie $\theta(x,y)$ podľa nasledujúcich vzťahov:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (19)$$

$$\theta(x,y) = \tan^{-1}((L(x+1,y) - L(x-1,y)) / (L(x,y+1) - L(x,y-1))) \quad (20)$$

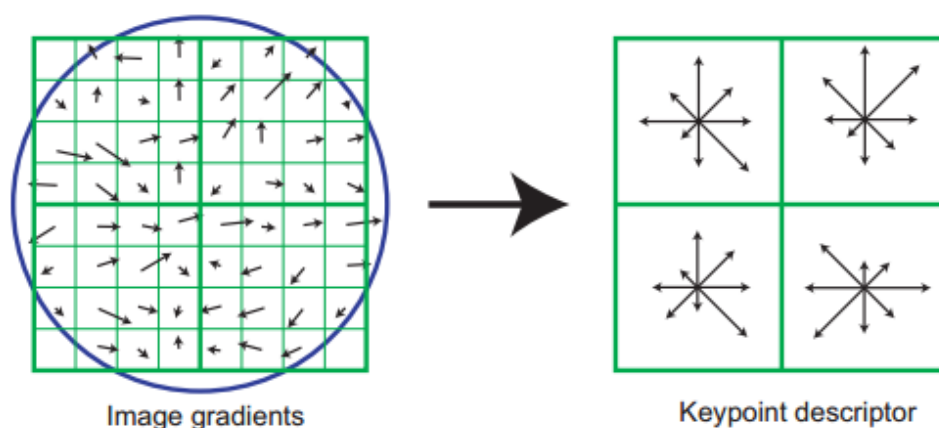
Na základe získaných hodnôt následne vytvoríme histogram o 36 binoch, kde každý bin odpovedá uhlu 10° a pokrývajúcich celých 360° . Veľkosť gradientu je vážená

Gaussovským oknom s rozložením σ 1,5 krát väčším ako mierka vyšetřovaného významného bodu. Výberom maximálnej hodnoty z daného histogramu určíme orientáciu daného významného bodu. Následne vyšetříme ešte ostatné hodnoty histogramu a pokiaľ niektorá z nich dosahuje aspoň 80% hodnoty maxima v danom histograme, vytvoríme pre daný bod extra významný bod, ktorý bude mať inú orientáciu. Ako píše David Lowe vo svojej publikácii, týchto bodov je menej ako 15% a ich hlavnou výhodou je, že prispievajú k stabilite nachádzania párov v neskoršom procese.

- Popis pomocou deskriptorov

V doterajšej časti metódy SIFT sme sa zaoberali viac-menej detekciou významných bodov. Ďalším krokom algoritmu je ich následný popis teda vytvorenie deskriptora. Vo vyššie uvedených krokoch sme analyzovali a brali v úvahu polohu, mierku a orientáciu týchto významných bodov, čím sme v konečnom dôsledku dosiahli nezávislosť na týchto parametroch.

Pri tvorbe deskriptoru metódy SIFT následne využijeme už vypočítané hodnoty amplitúdy gradientov a ich orientácie pre dané mierky. Princíp tvorby deskriptoru bude vysvetlený pre oblasť 8x8 bodov, ako zobrazuje nasledujúci obrázok:



Obrázok 13: SIFT - princíp tvorby deskriptoru [10]

V danej oblasti sme pre každý bod určili amplitúdu a orientáciu gradientu, čo je znázornené čiernymi šípkami v ľavej časti obrázku. Gradienty sú vážené Gaussovskou funkciou zobrazenou modrou kružnicou vďaka čomu predídeme nečakaným zmenám v deskriptore spôsobenými malými zmenami v našom poli. Následne sú z podoblastí o veľkosti 4x4 vytvorené 4 samostatné oblasti s histogramami o 8 intervaloch (angl. bins) odpovedajúcich 45°. Deskriptor je následne vytvorený z týchto hodnôt usporiadaných do vektora a v tomto prípade by mal veľkosť $2 \times 2 \times 8 = 32$ prvkov. David Lowe ale vo svojej práci počíta s pôvodnou oblasťou o veľkosti 16x16 bodov zosumovanou do oblasti 4x4 a teda jeho deskriptor pozostáva z $4 \times 4 \times 8 = 128$ binov.

Aby sa predišlo ovplyvneniu deskriptora zmenou kontrastu, je deskriptor po zostrojení normalizovaný. Zmena kontrastu pri ktorej je hodnota každého bodu násobená

konštantou sa týmto spôsobom vyruší. Zmena kontrastu pri pričítaní konštantnej hodnoty ku každému pixelu sa neprejaví, pretože hodnoty gradientov sú počítané z rozdielu, ktorý sa nezmení. V praxi sa môže objaviť nelineárna zmena kontrastu v rámci obrazu. Tieto zmeny menia hlavne amplitúdu gradientov a neprejavujú sa v orientácii, preto ako elimináciu týchto javov Lowe používa metódy, ktoré znižujú vplyv vysokých hodnôt amplitúdy gradientov.

2.2.3 Výhody / nevýhody

Ako vidieť z celého popisu je SIFT kompaktným balíkom detektoru a deskriptoru. Jeho komplexita narastá analogicky s množstvom problémov, ktoré sa snažil autor vyriešiť a to hlavne s cieľom vyvinúť algortimus, ktorý je nezávislý na posune, rotácii, mierke, zmene jas a malých skresleniach obrazu. Výhodou metódy je množstvo významných bodov, ktoré detekuje a z ktorých je následne možné vylúčiť tie, ktoré s vysokou pravdepodobnosťou budú dávať zlé výsledky pri aplikácii matching algoritmu, čo prispieva k robustnosti celého algoritmu. Na druhú stranu ide s komplexitou a dokonalosťou riešenie ruka v ruke výpočtová náročnosť. Autor v [10] síce tvrdí, že algortimus dosahuje takmer real-time výkon, ale ako poznamenávajú autori v [6] a [7] metóda je pomerne pomalá pri obrázku už relatívne bežnej veľkosti. Každopádne SIFT priniesol mnoho nových poznatkov, postupov a možností na ďalšie vylepšenia, hlavne čo sa týka výpočetnej rýchlosti a ďalšiemu priblíženiu sa real-time výpočtu.

2.2.4 Ukážka aplikácie

K ukážke výsledkov deskriptora bola použitá implementácia od Andrea Vedaldiho pre MATLAB dostupná z [19]. Na nasledujúcich obrázkoch sú prezentované získané výsledky. Nastavenie detektora bolo také, aby bolo detegovaných približne 500 oblastí a boli nejakým rovnomerným spôsobom rozložené v celom obraze. Ako je možné vidieť z priblíženej oblasti ŠPZ, nie sú v tejto oblasti detegované žiadne distingvované oblasti na základe, ktorých by bolo možné lokalizovať ŠPZ.



Obrázok 14: Ukážka výsledku algoritmu SIFT deskriptora na testovacie obrázky

2.3 SURF

2.3.1 Úvod

Ďalším z veľmi významných prístupov k analýze obrazu je metóda SURF. Skratka SURF, teda Speeded Up Robust Features pochádza z angličtiny a naznačuje, že sa jedná o metódu, ktorej cieľom bude rýchlejšia extrakcia vlastností z obrazu. Rovnako ako v prípade SIFT-u sa jedná o metódu kombinujúcu detektor a deskriptor v jednom. Metóda je pomerne mladá a stále sa predpokladá jej vývoj, poprípade vývoj nových metód s použitím poznatkov získaných zo SURF-u, podobne ako tomu pristupoval autor Herbert Bay, ktorý študoval dostupné možnosti a už vyvinuté postupy a pokúsil sa ich vylepšiť. Autor sa opieral hlavne o metódu SIFT, ktorá dokazovala dostačujúce výsledky, ale stále bola pomerne náročná na výpočet. Informácie získané pre danú kapitolu sú zo zdroja [11].

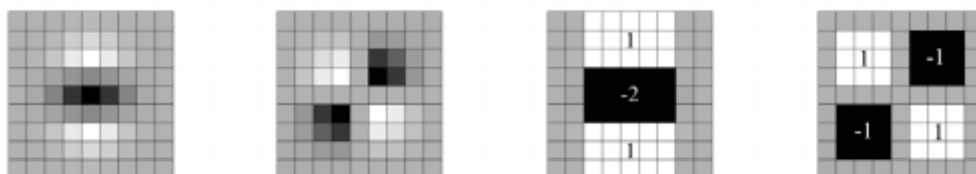
2.3.2 Funkčný a matematický popis

Ako bolo ukázané u metódy SIFT, Lowe využíva k vytvoreniu scale-space aproximáciu Laplaciánu rozdielom Gaussiánov. Nakoľko je tento spôsob ale náchylný k detekcií hrán a následne je potrebné tieto body odstrániť pomocou Hessian matice, Bay sa snažil o použitie detektora, ktorý hrany detegovať nebude a teda ušetrí sa čas, inak použitý na odstránenie týchto „falošných“ bodov. Iným spôsobom by bolo použiť Hessian maticu na začiatku a následne danému bodu priradiť mierku k dosiahnutiu nezávislosti na

mierke. Bay sa snažil obidva spôsoby sklbiť a vznikol tzv. Fast-Hessian detektor, popísaný Hessian maticou:

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (21)$$

kde $L_{xx}(\mathbf{x}, \sigma)$ je konvolúcia druhej derivácie Gaussovej funkcie v bode $\mathbf{x} = (x, y)$. Nasledujúci obrázok zobrazuje druhé derivácie Gaussovej funkcie a ich diskkrétne aproximácie:



Obrázok 15: Druhé derivácie Gaussovej funkcie (podľa x a xy) a ich diskkrétne aproximácie [11]

Použitie týchto aproximačných okien o veľkosti 9×9 bodov odpovedá najmenej mierke (konkrétne $s=1,2$). Vyššie mierky sú generované postupným zväčšovaním konvolučnej masky. Tieto rozmery sú 15×15 , 21×21 , 27×27 atď. Následne je potrebná extrakcia významných bodov. Tá je zhodná s metódou výberu maxima popísaného v kapitole o SIFT-e a zahŕňa porovnanie so susediacimi bodmi v priestore $3 \times 3 \times 3$.

Ďalším krokom postupu je priradenie orientácii zisteným významným bodom. To sa opäť uskutoční rovnakým spôsobom ako u SIFT-u, ale aj v tejto oblasti sa bral ohľad na zrýchlenie výpočtu a na rozdiel od SIFT-u je uvažovaný len dominantný smer gradientu. Pre výpočet orientácie sa používa kruhová oblasť v okolí detegovaného významného bodu o polomere $6s$. Pre dané body sa spočítajú odozvy na tzv. Haarove vlny v x a y smere. Hodnoty sú následne vážené Gaussovskou funkciou v rámci kruhu a dominantný smer je určený sumovaním cez všetky vypočítané smery a následným zaradením do jedného zo 6 smerov odpovedajúcich 60 stupňovému výseku.

Nasleduje samotný popis významného bodu, teda kalkulácia deskriptoru. Autori sa opäť snažili inšpirovať metódou SIFT, ktorá ale využíva deskriptor o 128 prvkoch a pri hľadaní korešpondencií medzi dvoma obrazmi, je potrebné počítat Euklidovské vzdialenosti medzi deskriptormi o dĺžke 128 prvkov čo je opäť pomerne výpočtovo náročné a v istom zmysle dáva priestor pre zlepšenie. K tomu si najprv okolo daného významného bodu vytvoríme okno o veľkosti $20s$, ktoré ma rovnakú rotáciu ako bola priradená danému významnému bodu. Tá sa následne rozdelí na 4×4 podoblastí každá o veľkosti 5×5 bodov. Pre každý bod podoblasti sa následne určí odozva na Haarovu funkciu d_x a d_y , teda v smere horizontálnom a vertikálnom s ohľadom na natočenie okna. Z dôvodu zvýšenia robustnosti sú tieto hodnoty opäť vážené Gaussovskou funkciou. Pre všetkých 16 podoblastí sú následne sumované hodnoty d_x , d_y a rovnako

tak $|d_x|$ a $|d_y|$. To v konečnom dôsledku vytvára deskriptor o $4 \times 4 \times 4 = 64$ hodnotách pre jeden významný bod. Následne prevedieme normalizáciu zaručujúcu nezávislosť na lineárnej zmene kontrastu.

2.3.3 Výhody / nevýhody

Metóda SURF splňuje to, čo o nej prezrádza jej názov. Autorovi sa podarilo nájsť ešte priestor na zjednodušenie metódy SIFT, poprípade nahradiť niektoré kroky metódy krkmi výpočtovo jednoduchšími pri minimálnom obmedzení na stabilitu a kvalite výsledku. Výhodou je množstvo dostupných implementácií a relatívna rýchlosť, ale ako autor sám v závere svojej práce naznačuje, v budúcnosti sa bude ďalej pracovať na optimalizácii kódu pre ďalší zrýchlenie metódy.

2.3.4 Ukážka aplikácie

V prípade deskriptora SURF bolo opäť možné použiť funkciu implementovanú priamo v MATLAB-e. Konkrétne sa jedná o funkciu *detectSURFFeatures*. Nasledujúcim kódom boli získané vzorové ukážky:

```
surf_points_real = detectSURFFeatures(I_real_gray);  
figure(1)  
imshow(I_real);  
hold on;  
plot(surf_points_real.selectStrongest(200))
```



Obrázok 16: Ukážka výsledku algoritmu SURF deskriptora na testovacie obrázky

2.4 BRIEF

Zdrojom informácií pre túto kapitolu je [14].

2.4.1 Úvod

Cieľom stanoveným pri vyvíjaní metódy SURF bolo zjednodušenie deskriptoru pri zachovaní vlastností a tým zníženie výpočtovej náročnosti. To je rovnako cieľom metódy BRIEF, kde sa ale autori navyše snažia o zníženie požiadaviek na pamäť, teda zmenšenie priestoru, ktorý deskriptor v pamäti zaberá. To je možné viacerými spôsobmi, napríklad použitím techník PCA a LDE, ktoré v princípe deskriptor redukujú na jeho veľkosti zmenou jeho dimenzií. Druhou možnosťou je kvantizácia floating-point hodnôt na celočíselné integer hodnoty s nižšou pamäťovou náročnosťou. Poslednou možnosťou je daný deskriptor zmenšiť čo sa dimenzie týka a previesť ho na jeho binárnu reprezentáciu. Celý tento proces môže ušetriť čas a pamäť pri následnom hľadaní korešpondencií, avšak tento postup nie je z hľadiska časovej optimalizácie najlepší. Preto sa autori rozhodli o priame vytvorenie robustného binárneho deskriptora – Binary Robust Independent Elementary Features.

2.4.2 Funkčný a matematický popis

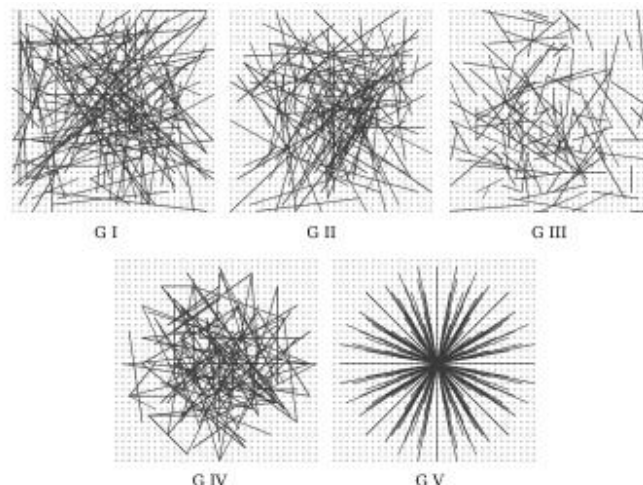
Východiskom pre daný deskriptor sú už známe významné body získané niektorým z detektorov spomenutých v kapitole 2. Následne pracuje s obrazom, na ktorý bol aplikovaný filter na rozostrenie a odstránenie šumu. Metóda je založená na porovnávaní intenzity skúmaného bodu voči bodom v okolí a popísaná nasledujúcou testovacou funkciou:

$$\tau(p, x, y) = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{inak} \end{cases} \quad (22)$$

kde $p(x)$ je hodnota intenzity bodu $x = (u, v)^T$ v obraze p . Následnej je vytvorený deskriptor o veľkosti $n_d = 126, 256$ alebo 512 prvkov. Deskriptor je bitový reťazec a môžeme ho definovať nasledovne:

$$f_{n_d}(p) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p, x, y) \quad (23)$$

Dvoma problémami celej metódy sú teda vhodná voľba filtrácie šumu v obraze a rozloženie bodov $p(y)$ slúžiacich k porovnaniu s bodom vyšetrovaným. K tomu autori použili 5 nasledujúcich rozložení, ktoré podrobili experimentu a zistili, že symetrické rozloženie č.5 vykazuje najhoršie výsledky, zatiaľ čo ostatné rozloženia vykazujú pomerne rovnaké výsledky s jemným náskokom rozloženia č.2 :



Obrázok 17: Rozloženie dvojíc porovnávaných bodov [14]

2.4.3 Výhody / nevýhody

Metóda je svojim princípom pomerne jednoduchá na tvorbu ako aj na výpočtovú náročnosť a je vhodná pre real-time aplikácie s menšími pamäťovými nárokmi. Nevýhodou môže byť závislosť metódy BRIEF na rotácii, aj keď podľa experimentov BRIEF dokáže pracovať s málo pootočenými obrazmi.

2.5 BRISK

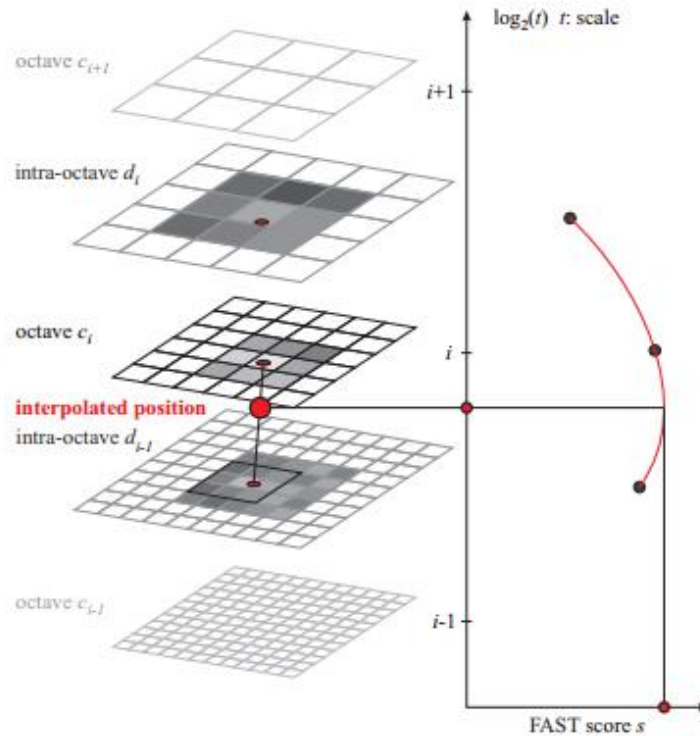
Pri tvorbe tejto kapitoly boli využité informácie zo zdroja [16].

2.5.1 Úvod

Za touto relatívne mladou prácou z roku 2011 stoja Stefan Leutenegger, Margarita Chli a Roland Y. Siegwart z univerzity ETH vo švajčiarskom Zürichu. Vo svojej práci zhrňajú aktuálny stav v oblasti počítačového videnia, hlavne teda oblastí detektorov a deskriptorov významných oblastí a ich následného spájania pomocou „matching“ algoritmov. Autori vyzdvihujú kvality algoritmu SIFT, ale na druhú stranu konštatujú jeho najväčšie nedostatky, konkrétne rýchlosť a náročnosť výpočtu. Oproti nemu vyzdvihujú algoritmus SURF, poprípade kombináciu detektora FAST a deskriptora BRIEF, ktoré ale poskytujú nižšiu opakovateľnosť, robustnosť a nízku mieru invariantnosti voči transformáciám obrazu za cenu vysokej rýchlosti výpočtu. Autori deklarujú, že nimi vytvorený BRISK (z anglického Binary Robust Invariant Scalable Keypoints) dosahuje kvalitatívne približne rovnaké výsledky ako SURF, avšak znižuje výpočtovú náročnosť. Autori teda vytvorili kombináciu detektora, deskriptora a rovnako tak matching algoritmu, ktorá by mala byť invariantná voči rotácii a zmene mierky, ako odhaľuje už názov. Vytvorený detektor a deskriptor by mali byť použiteľné aj samostatne s iným deskriptorom popr. detektorom. Ich prácu si predstavíme v nasledujúcej podkapitole.

2.5.2 Funkčný a matematický popis

Autori sa pri tvorbe nového detektora inšpirovali metódou AGAST, ktorá je akýmsi rozšírením už spomínaného detektora FAST. Z dôvodu invariantnosti voči zmene mierky bolo potrebné hľadať významné body okrem úrovne obrazu, aj v iných mierkach za použitia scale-space, ktoré bolo popísané napríklad u algoritmu SIFT. Princíp tvorby scale-space ukazuje Obrázok 11. Zmenou ale je, že BRISK detektor určuje skutočnú mierku v spojitom scale-space. Nasledujúci obrázok zobrazuje formu scale-space pre detektor algoritmu BRISK:



Obrázok 18: BRISK scale-space a detekcia významných bodov [16]

Ako je možné vidieť z obrázku, tento scale-space pozostáva z n oktáv označených c_i a n pod-oktáv označených d_i . Hodnota n je najčastejšie 4. Oktávy c_i vznikajú polovičným vzorkovaním z predchádzajúcej oktávy, začínajúc na oktáve c_0 odpovedajúcej pôvodnému obrazu. Pod-oktávy sa nachádzajú medzi c_i a c_{i+1} . Prvá z nich, d_0 , vzniká 1,5 násobným pod-vzorkovaním originálneho obrazu zatiaľ čo ostatné pod-oktávy vznikajú postupným polovičným vzorkovaním. V prípade, že označíme mierku t , potom

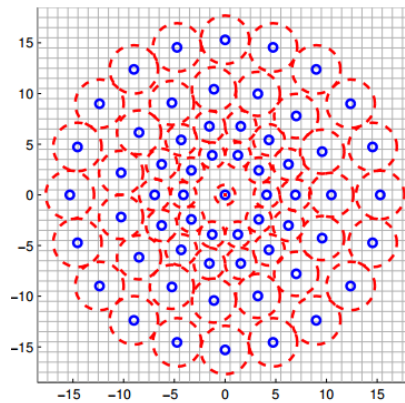
$$t(c_i) = 2^i \text{ a } t(d_i) = 2^i * 1,5 \quad (24)$$

Po vytvorení scale-space pristúpime priamo k hľadaniu významných bodov. To prebieha pomocou algoritmu FAST aplikovaného na každú oktávu a pod-oktávu s rovnakou hodnotou prahu T . Na nájdené body je následne aplikovaný non-maxima suppression tak, že vybraný bod musí spĺňať podmienku maxima v porovnaní so svojimi 8 susedmi v rámci oktávy v ktorej sa nachádza a následne oproti oktáve pod a nad oktávou v ktorej leží. Porovnáva sa hodnota skóre s , ktorá je definovaná ako

maximálny prah, pri ktorom je bod stále považovaný za roh. Aby sme následne získali skutočnú mierku významného bodu, definujeme v smere zmeny mierky parabolu, ako to zobrazuje pravá časť Obrázok 18. Skutočná mierka významného bodu je určená ako maximum tejto funkcie. Záverečným krokom detekcie je interpolácia súradníc medzi oktávami okolo získanej mierky.

Po získaní významných bodov pristúpime k ich popisu. Vyššie popísaným spôsobom detekcie sme získali významné body invariantné voči mierke. Cieľom deskriptora bude získanie invariantnosti voči rotácii tak, že identifikujeme charakteristickú orientáciu každého významného bodu. Následne prevedieme porovnanie jasú a vytvoríme 512 bitový deskriptor.

Na úvod sa vytvorí vzorkovací obrazec tak, že je vytvorených N rovnomerne rozložených kruhov ležiacich na sústredných kružniciach so stredom vo významnom bode. Na tieto oblasti je aplikovaný Gaussovský filter s hodnotou σ úmernou vzdialenosti medzi stredmi kruhov ležiacich na jednej kružnici. Nasledujúci obrázok ilustruje danú situáciu:



Obrázok 19: Vzorkovací obrazec BRISK [16]

Predpokladáme, že existuje $N * (N - 1)/2$ párov bodov. Vyberieme jeden z nich, pár (p_i, p_j) pre ktorý platí, že intenzity jeho zložiek sú $I(p_i, \sigma_i)$ a $I(p_j, \sigma_j)$. Pre tento pár spočítame hodnotu gradientu $g(p_i, p_j)$ ako:

$$g(p_i, p_j) = (p_j - p_i) * \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \quad (25)$$

Následne pre všetky existujúce páry vytvoríme dve množiny. Množinu blízkych a vzdialených párov, kde vzdialenosti sú menšie ako hodnota δ_{max} , poprípade väčšia ako hodnota δ_{min} . Štandardne sú tieto hodnoty volené ako $\delta_{max}=9,75*t$ a $\delta_{min}=13,67*t$, kde t je mierka významného bodu. Pre ďalší výpočet sa počíta so vzdialenými pármami pre ktoré určíme celkovú charakteristickú orientáciu obrazca ako:

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} * \sum_{(p_i, p_j) \in L} g(p_i, p_j) \quad (26)$$

Vzorkovací obrazec je následne rotovaný o uhol $\alpha = \arctg(g_y, g_x)$ okolo zvoleného významného bodu. Pre porovnanie sú následne využité už získané blízke páry a v prípade, že je intenzita bodu p_j vyššia ako intenzita bodu p_i je prislúchajúca hodnota v deskriptore 1, v opačnom prípade 0.

2.5.3 Výhody / nevýhody

Táto metóda patrí medzi hodne sofistikované metódy a zahŕňa v sebe mnoho užitočných vlastností. Cieľom bolo získať kvalitatívne približne rovnaké výsledky ako metódou SURF s podmienkou zrýchlenia a zjednodušenia výpočtu. Medzi výhody algoritmu patrí jeho nezávislosť (do istej miery) na rotácii a zmene mierky a hlavne rýchlosť, ktorá je vhodná pre hard real-time aplikácie. Metóda vykazuje oproti SURF-u zhoršené výsledky pri väčších zmenách mierky alebo otočenia.

2.5.4 Ukážka aplikácie

Za účelom prezentácie výsledkov BRISK deskriptora bola použitá funkcia MATLABu `detectBRISKFeatures`. Ukážka funkčného kódu:

```
brisk_points_real = detectBRISKFeatures(I_real_gray);  
figure(1)  
imshow(I_real);hold on;  
plot(brisk_points_real);
```



Obrázok 20: Ukážka výsledku algoritmu BRISK deskriptora na testovacie obrázky

2.6 ORB

Kapitola o deskriptore ORB vychádza zo zdroja [17].

2.6.1 Úvod

Opäť jedna z relatívne nových metód publikovaná v roku 2011 Ethanom Rublee a jeho kolegami v Kalifornii. Cieľom autorov je vytvorenie výpočtovo jednoduchšej náhrady algoritmu SIFT s možnosťou využitia v jednoduchších zariadeniach (napríklad chytrých telefónoch) za účelom rýchlej tvorby panoramatických fotografií, poprípade sledovanie oblasti v obraze v reálnom čase. Z výsledkov, ktoré prezentujú plynne, že ORB funguje približne rovnako ako SIFT, lepšie ako SURF a je o dva rády rýchlejší ako SIFT.

Skratka ORB vznikla na základe už známych algoritmov využitých pre vznik tejto novej metódy, konkrétne: Oriented FAST and Rotated BRIEF. Názov už teda implikuje, že metóda použije dobre známy FAST detektor, ktorý upraví o detekciu orientácie a ako deskriptor bude použitý rýchly a jednoduchý BRIEF s vylepšením za cieľom získania invariantnosti na rotácii. Opäť sa teda jedná o komplexné riešenie detektoru a deskriptor v jednom.

2.6.2 Funkčný a matematický popis

K detekcii sa teda využíva algoritmus FAST rozšírený o pyramídu pre rôzne mierky a Harrisov rohový filter, ktorý odstraňuje detegované rohy. Princíp je založený na meraní „rohovosti“ získaných významných bodov, ktoré následne zoradíme a vyberieme N bodov s najvyššou hodnotou „rohovosti“. Algoritmus bolo následne potrebné rozšíriť o operátor rotácie. Inšpirovať sa u SIFT a SURF nebolo možné, pretože tieto spôsoby vyžadujú vysokú výpočtovú náročnosť (SIFT) alebo poskytujú slabšie výsledky (SURF). Autori sa preto inšpirovali technikou ťažiska prezentovanou L.P Rosinom. Táto metóda poskytuje jednu presnú hodnotu orientácie pre zvolený významný bod. Táto technika predpokladá, že intenzita ťažiska je posunutá mimo svoj stred, a tento vektor slúži k priradeniu orientácie. Pomocou nasledujúcich výpočtov určíme ťažisko C a uhol orientácie θ :

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (27)$$

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (28)$$

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (29)$$

Princíp algoritmu BRIEF nie je potrebné vysvetľovať, je popísaný v kapitole BRIEF . Problémom je, že je veľmi citlivý na rotáciu v rámci obrazu. Pri ORB existujú dva požiadavky pre tvorbu kvalitného deskriptora:

1) Požadujeme nekorelované páry na porovnávanie, čo znamená, že každý nový pár prináša do deskriptora novú informáciu.

2) Vysoký rozptyl párov, ktorý zabezpečí väčšiu selektívnosť, pretože reaguje inak na vstupy.

Aby to bolo zabezpečené autor navrhuje učenie sa vzorkovacích párov. Vytvorí sa množina tréningových vzorov o počte asi 200 000 významných bodov. Tie vzory následne zoradíme podľa vzdialenosti od strednej hodnoty 0,5 čím vytvoríme vektor T . Prvý test zaradíme do množiny výsledkov R . Následne vyšetríme ďalší test, ktorý počítame do výsledku len v prípade, že absolútna korelácia voči všetkým hodnotám z množiny R je menšia ako nastavený prah. Tento proces opakujeme kým nezískame 256 testov. V prípade, že nezískame 256 testov, je potrebné znížiť hodnotu prahu a celý postup opakovať. Následne prebehne samotné porovnanie a vytvorenie binárneho deskriptora.

2.6.3 Výhody / nevýhody

Deskriptor metódy ORB oproti pôvodnému BRIEF navyše implementuje invariantnosť voči rotácii a vhodné rozloženie párov pre pozorovanie na základe učenia. Metóda je pomerne robustná, invariantná voči rotácii a zmene mierky, pomerne rýchla a odolná voči šumu a je vhodná pre aplikácie spomínané v úvode. Pri porovnaní s pôvodnou kombináciou FAST + BRIEF vykazuje pri malých odchýlkach mierky a otočenia horšie výsledky, ale so zvyšujúcou sa výchylkou od pôvodnej mierky a uhlu natočenia ORB víťazí.

2.7 FREAK

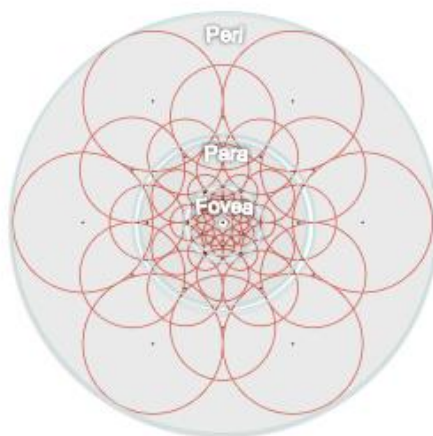
2.7.1 Úvod

Najnovšia práca [18] zo všetkých doteraz prezentovaných algoritmov. Táto práca bola použitá ako zdroj informácií pre túto kapitolu. Za jej vznikom stoja Alexandre Alahi, Raphael Ortiz a Pierre Vandergheynst z Ecole Polytechnique Federale de Lausanne vo Švajčiarsku. Ich cieľom bolo už štandardne vytvoriť rýchly, kompaktný a robustný deskriptor. Zamerali sa na deskriptor BRIEF, ktorý sa úspešne (s modifikáciami) používa aj v deskriptoroch BRISK a ORB. Problémom tohto princípu je výber správnych bodov pre porovnanie intenzity. Autori sa zameriavajú na spôsob akým daný problém efektívne rieši príroda, teda ľudské oko, konkrétne sieťnica. Z toho je odvodený aj názov: Fast Retina Keypoint.

2.7.2 Funkčný a matematický popis

Na úvod je potrebné nahliadnuť do toho ako funguje ľudské oko, konkrétne sieťnica. Predpokladá sa, že sieťnica získava detaily o obraze na základe aplikácie rozdielu gaussianov pre rôzne veľkosti. Dôležitou je pri tom topológia sieťnice. Vysvetľovať

princíp ľudského videnia do hĺbky je mimo rozsah tejto práce. Z týchto faktov ale plyní zvolené rozloženie vzorkovacej mriežky pre navrhovaný deskriptor. Táto plocha je kruhového tvaru a hustota bodov klesá exponenciálne od stredu. Aby sa odstránil vplyv šumu je potrebné aplikovať filter. V tomto prípade, je na každú oblasť použitý filter s inou veľkosťou jadra a plochy na ktoré je aplikovaný sa prekrývajú. Túto situáciu zobrazuje nasledujúci obrázok:



Obrázok 21: Vzorkovacia mriežka pre FREAK [18]

Reťazec deskriptora F je následne vytvorený podobným spôsobom ako u deskriptora BRIEF:

$$F = \sum_{0 \leq a < N} 2^a T(P_a) \quad (30)$$

Kde:

$$T(P_a) = \begin{cases} 1 & \text{ak } I(P_a^{r_1}) - I(P_a^{r_2}) > 0 \\ 0 & \text{inak} \end{cases} \quad (31)$$

a P_a je pár vybraných oblastí, N je požadovaná dĺžka deskriptoru a $I(P_a^r)$ je intenzita daného bodu po vyhladení. Tento spôsob by ale viedol na porovnávanie tisícok bodov a preto autori využívajú algoritmus prezentovaný u ORB, ktorý vyberá len žiadané množstvo párov, ktoré majú nízku vzájomnú koreláciu.

2.7.3 Ukážka aplikácie

V prípade deskriptoru FREAK MATLAB zatiaľ neposkytuje priamo funkciu, ktorá by z obrázku získala významné body a následne vypočítala deskriptory. Je však možné použiť kombináciu niektorého z vyššie spomenutých detektorov a deskriptor FREAK zadať ako parameter do funkcie `extractFeatures`. Za vzorovým kódom sú uvedené výsledky.

```
regions = detectMSERFeatures(I_real_gray,
'MaxAreaVariation', 0.1, 'ThresholdDelta', 3);
```

```
[features,valid_points]=extractFeatures(I_real_gray,regions
,'Method','FREAK');
figure(1); imshow(I_real); hold on;
plot(valid_points,'showOrientation',true);
```



Obrázok 22: Ukážka výsledku algoritmu FREAK deskriptora na testovacie obrázky

3 EXPERIMENTÁLNA ČASŤ

Porovnať detektory a deskriptory významných oblastí na základe teoretického spracovanie samozrejme nie je možné a preto bolo potrebné nájsť nejakú aktuálnu úlohu z oblasti počítačového videnia kde by bolo možné otestovať odozvu daných algoritmov pri riešení danej úlohy, odhaliť ich silné stránky a podmienky pri ktorých pôsobia robustne a spoľahlivo a rovnako tak nájsť ich slabé miesta a odhaliť problémy a podmienky pri ktorých sú dané riešenia nespoľahlivé. Výstupom by mohlo byť akési poradie testovaných metód a algoritmov na danej úlohe čo sa týka štatistickej úspešnosti.

Po konzultácii s vedúcim diplomovej práce a zvážení rôznych variant aplikácii, ktoré sa v rámci skupiny počítačového videnia na ústave automatizácie riešia sa do užšieho výberu dostali dve možné úlohy. Konkrétne lokalizácia ŠPZ alebo trojuholníkových dopravných značiek v obraze. Víťaznou úlohou je napokon lokalizácia ŠPZ v obraze a to z toho dôvodu, že k tejto úlohe bola dostupná pomerne široká testovacia galéria a na rozdiel od druhej spomínanej, kde by sa jednalo iba o časť celej lokalizačnej problematiky poskytuje lokalizácia ŠPZ kompletnú možnosť riešenia, nakoľko neexistujú viaceré rozličné tvary poznávacích značiek a záver, ktorý bude výstupom tejto práce bude štatisticky platný pre lokalizáciu všetkých ŠPZ, samozrejme za daných, neskôr popísaných podmienok.

V tomto úvodnom bode je ale dôležité poznamenať, že cieľom tejto diplomovej práce nie je vyriešiť problém lokalizácie ŠPZ ako takej. To čo je predmetom danej testovacej úlohy je zistiť, ako sa dané detektory a deskriptory na danej úlohe správajú, za akých podmienok zlyhávajú a za akých sa naopak správajú spoľahlivo. Rovnako tak bude dôležité porovnať, ako dobre je možné ich zamerať na riešenie danej úlohy. To napríklad znamená zhodnotiť, nakoľko detegujú oblasti, ktoré pre našu úlohu nie sú relevantné, voči oblastiam ktoré sú, a či je možná prípadná eliminácia tých, pre danú úlohu nevýznamných oblastí bez toho, aby sme eliminovali oblasti fundamentálne pre danú problematiku, teda riešenie lokalizácie ŠPZ v obraze.

Z detektorov a deskriptorov popísaných v prvej a druhej kapitole budú na testovacie účely použité tie, ktoré sú už priamo implementované v prostredí MATLAB 2013 (poprípade verzia 2014 pre BRISK) a ktoré poskytujú potenciál pre riešenie danej problematiky. Na základe zadania a dohody s vedúcim práce bolo rozhodnuté, že budú testované nasledujúce metódy: Harrisov detektor, Shi-Tomasi detektor, FAST, MSER, SURF a BRISK. Potenciál ďalej poskytujú HoG a SIFT, ktorých možnosti budú popísané, ale z príslušných dôvodov, ktoré budú uvedené v odpovedajúcich kapitolách nebudú štatisticky testované.

Z koncepcného hľadiska, bude tretia kapitola delená do podkapitol v rámci ktorých bude popísaný algoritmus lokalizácie ŠPZ použitím danej metódy, nastavenie

parametrov pre rôzne galérie a zhodnotené dosiahnuté výsledky. Prvá a posledná podkapitola budú do istej miery špecifické. V prvej podkapitole bude v krátkosti popísaná testovacia galéria a úpravy, ktoré na nej boli prevedené. V kapitole poslednej sa bude nachádzať zdôvodnenie netestovania niektorých metód s krátkou analýzou.

3.1 Testovacia galéria

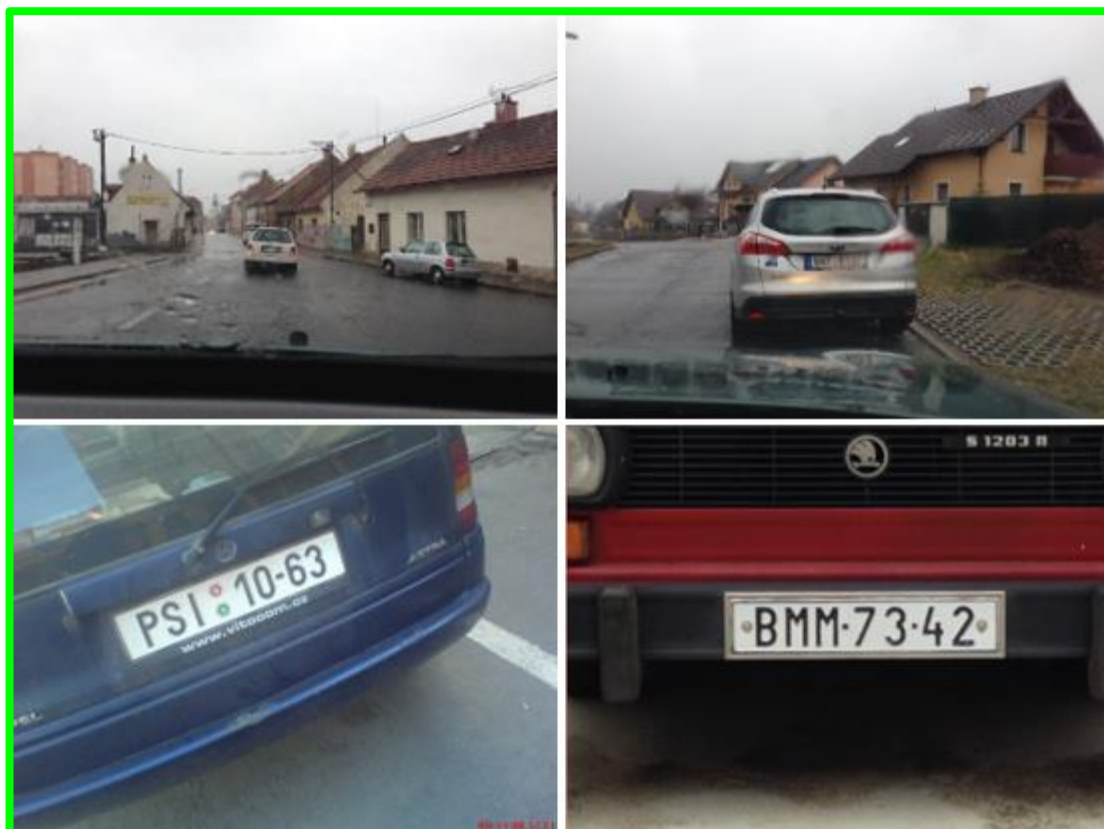
Pre účely testovania vyššie spomenutých metód v tejto diplomovej práci bola poskytnutá galéria vozidiel s ŠPZ, získaná v rámci výuky v predmetoch počítačového videnia ako aj pri spracovávaní inej bakalárskej a diplomovej práce. Galéria je rôznorodého charakteru a obsahuje snímky získané z fotoaparátov a mobilných telefónov, fotené ručne, bez statívu (okrem Real1 a Real2), čo do tohto súboru vnáša čiastočný prvok náhody. Fotky sú získané za rôznych vonkajších podmienok. Galéria má nasledujúcu štruktúru:

- Real1 – 531 snímok
- Real2 – 152 snímok
- MPOV_1 – 90 snímok
- MPOV_2 – 45 snímok
- MPOV_3 – 79 snímok
- db_parkoviste_slunecno – 258 snímok
- db_parkoviste_zatazeno – 121 snímok
- db_destivo – 88 snímok
- db_slunecno – 34 snímok

Metódy a algoritmy, ktoré budú popísané nižšie nie sú stavané k tomu, aby lokalizovali ŠPZ v obraze, ktorý je značne rotovaný, prípadne časť ŠPZ nie je vidieť. Pre princíp algoritmu MSER následne nie sú vhodné ŠPZ starého typu, ktoré boli vo formáte troch znakov a dvoch číselných dvojíc oddelených pomlčkou („PPP ČČ-ČČ“). Z tohto dôvodu bolo potrebné dané galériu kriticky zredukovať. V každom prierezníku galérie je uvedený súbor Annotation.csv, ktorý obsahuje zoznam všetkých snímok v galérii a súradnice 4 rohových bodov ŠPZ v obraze. Ďalej sa tu nachádzajú dva podobné .csv súbory z nich jeden má príponu „_MSER“ a druhý „_4P“. Tie obsahujú už redukovanú galériu, pre danú metódu, či už MSER, alebo metódu založenú na hľadaní 4 rohových bodov (neplatí pre galérie Real1 a Real2, ktoré sú pomerne nové a zatiaľ bez anotácií). Obrázok 23 názorne ukazuje, aké snímky boli z testovania vynechané.

Galérie Real1 a Real2 sú špecifické tým, že poskytujú snímky aké by boli spracovávané keby sa jednalo o reálny kamerový systém nainštalovaný nad vozovkou snímajúci prechádzajúce vozidlá. Snímky nie sú dokonalé v tom, že sa vozidlo sa nachádza v rôznych vzdialenostiach od kamery. V reálnom systéme by tento nedostatok mohol byť pomerne jednoducho eliminovaný a ŠPZ by sa nachádzala v približne rovnakej

vertikálnej oblasti v rámci snímku, čo by umožnilo vytvorenie záujmovej oblasti (ROI-region of interest), ktorá by bola postačujúca pre ďalšie vyšetrenie. Tieto dve galérie majú ale pre výsledky práce menší vplyv, pretože neposkytujú možnosť porovnať rôzne metódy za značne rozdielnych podmienok. Snímky sú síce získané za rôznych svetelných podmienok, ale čo sa scény týka sú homogénne. Tieto galérie slúžia ako dobrý odrazový mostík na overenie funkčnosti algoritmov a následne sú možné záťažové testy za rôznych podmienok v ostatných galériách.



Obrázok 23: Príklad snímok, ktoré boli z galérie vynechané. LH: príliš malá ŠPZ, PH - časť ŠPZ neviditeľná, LD - fotené z perspektívy + starý typ ŠPZ, PD - starý typ ŠPZ (len MSER)

3.2 Špecifikácia ŠPZ

V tejto podkapitole bude veľmi stručne popísaná štruktúra českej štátnej poznávacej značky (ŠPZ), ktorá bude pre ďalšie časti tejto práce esenciálnym prvkom. Tvarom, vlastnosťami a pravidlami týkajúcimi sa štátnych poznávacích značiek sa v Českej republike zaoberá zákon č. 243/201 Sb. ktorý prešiel novelizáciou v roku 2004. Pre účely tejto práce je najdôležitejšia aktuálna štandardná (a najrozšírenejšia) štátna poznávacia značka, ktorá na ľavej strane obsahuje modrý prúžok s logom európskej únie a kódom krajiny, v našom prípade CZ. Telo ŠPZ následne tvorí trojica znakov pozostávajúca z kombinácie čísla, písmena a čísla kde písmeno označuje kraj v ktorom bola ŠPZ vydaná. Tento typ ŠPZ je momentálne v Českej republike najrozšírenejší a všetky ďalej vytvorené algoritmy by si s ním mali poradiť. Niektoré z nich si navyše

poradia aj s ŠPZ staršieho typu, konkrétne s typom ktorý, je rozmerovo a obsahovo rovnaký, ale neobsahuje modrý pruh Európskej únie ako aj s ešte starším typom ktorý obsahoval trojicu písmen a dve dvojice čísel oddelené pomlčkou ako je možné vidieť napríklad v ľavej dolnej časti Obrázok 23. Príklad tejto štandardnej, pre účely tejto práce esenciálne štátnej poznávacej značky je uvedený na nasledujúcom obrázku:



Obrázok 24: Príklad ŠPZ pre účely DP

3.3 Lokalizácia ŠPZ použitím detektora MSER

3.3.1 Popis algoritmu

Ako bolo popísané v kapitole 1.5, MSER funguje na princípe vyhľadávania homogénnych oblastí, takzvaných „blobov“. Jedná sa teda o akési tmavé a svetlé spojené oblasti v obraze. Táto skutočnosť je dobre využiteľná pre riešenie úlohy lokalizácie ŠPZ, pretože samotné tmavé znaky na svetlom podklade takouto oblasťou nepochybne sú. Algoritmus je priamo implementovaný v MATLAB-e a MSER oblasti (regióny) je možné získať pomocou príkazu `detectMSERFeatures` s rôznymi parametrami. Pre naše využitie sú dôležité najmä:

- **ThresholdDelta** – je parameter, ktorý definuje krok medzi úrovňami prahu pre ktorý bude testovaná stabilita daného MSER regiónu.
- **MaxAreaVariation** – parameter, ktorý určuje maximálnu možnú zmenu detegovanej oblasti, medzi jednotlivými prahmi.

Príkaz nám vracia detegované MSER oblasti ako premennú typu `MSERRegions`, ktorá obsahuje stred elipsy ohraničujúcej danú oblasť, jej horizontálnu a vertikálnu osu, orientáciu a štruktúru obsahujúcu všetky pixely, ktoré patria do danej oblasti. Túto štruktúru je vhodné pred samotným činením algoritmu vyhľadávania ŠPZ kriticky zredukovať čo následne značne zrýchli jadro algoritmu. Eliminácia prebieha na základe nasledujúcich krokov:

- Odstránenie elíps, kde je uhol medzi horizontálnou rovinou a smerom dlhšej osy elipsy v intervale 0 až 45°, pretože na znakoch ŠPZ sú elipsy orientované dlhšou osou vertikálne.
- Odstránenie elíps, ktoré sú 9-krát dlhšie ako sú široké. To odstráni množstvo pre našu úlohu falošných elíps, ktoré sa objavujú na mriežke chladiča a pomer 9 zaručuje, že najužšia elipsa na znaku „I“ nebude eliminovaná.

- Eliminácia svetlých MSER oblastí, ktorá značne znižuje počet kandidátov nakoľko štatisticky eliminuje polovicu pôvodných oblastí.
- Eliminácia oblastí obsahujúcich menší počet pixelov ako 2×10^5 -násobok veľkosti obrazu odstráni veľmi malé MSER oblastí, ktoré by v ďalšom kroku nebolo možné použiť na rozpoznávanie znakov.
- Eliminácia oblastí, ktoré majú stred elipsy v blízkosti inej oblasti. Táto vzdialenosť je definovaná parametrom a ponecháva elipsu s najväčším počtom pixelov patriacich do daného regiónu. Pred použitím danej eliminácie boli na väčšine znakov v rámci ŠPZ detegované 2-4 približne rovnaké MSER regióny, líšiace sa len o jednotky pixelov.

Princíp samotného algoritmu ilustruje nasledujúci obrázok. Algoritmus hľadá 7 elíps, špecifickej orientácie, špecifických geometrických vlastností v špecifickom rozmiestnení v priestore. Z tohto dôvodu, bolo potrebné zredukovať galériu o snímky automobilov s ŠPZ starého typu ako bolo už spomínané v kapitole, kde bola analyzovaná galéria. Vytvoriť algoritmus, ktorý by detegoval aj ŠPZ s týmto usporiadaním znakov by samozrejme nebol problém, každopádne nie je cieľom tejto práce vytvoriť algoritmus detegujúci každý typ ŠPZ, ale overiť ako na danom type ŠPZ MSER funguje.

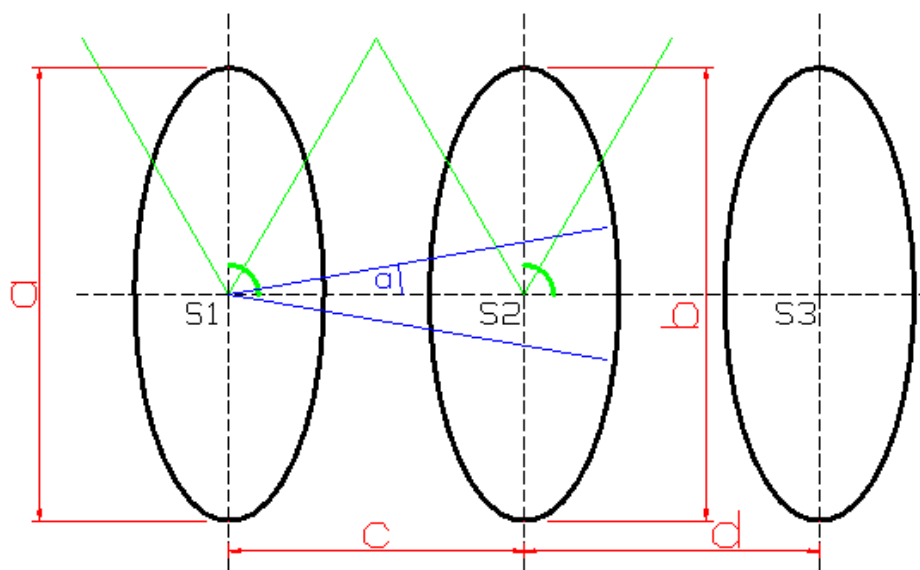


Obrázok 25: Príklad detegovaných oblastí v rámci ŠPZ po eliminácii

Vytvorený algoritmus obsahuje veľké množstvo pevných číselných konštánt, ktoré definujú vôľu pri hľadaní kandidátskych kombinácií. Tieto parametre boli získané experimentálnym procesom na snímkach, ktoré boli považované na hranici, kde by ŠPZ mala byť ešte detegovaná. Tieto konštanty sú považované za interné, principiálne hodnoty a preto sú pre celú galériu nemenné a rovnako nie sú uvádzané v tabuľke parametrov s ktorými boli dané výsledky získané.

Systematický princíp algoritmu je nasledovný a zobrazený na Obrázok 26 a Obrázok 28: Nakoľko sa jedná o ŠPZ, ktorá obsahuje jednu trojicu a jednu štvoricu znakov, na počiatku je potrebné nájsť dvojice elíps (oblastí), ktoré majú uhol, ktorý zvierá dlhšia osa elipsy s vodorovnou rovinou v intervale 60° až 120° a ktorých veľkostí sa líšia o maximálne $\pm 50\%$ (pomer veľkostí $a:b$). Táto konštanta sa môže zdať veľká, každopádne zahŕňa najnáročnejšie prípady, keď cez časť ŠPZ prechádza tieň, a oblasť na znaku detegovaná ako oblasť MSER môže byť značne menšia. Výchylka v ose y

medzi stredmi daných elíps môže byť maximálne $\pm 10^\circ$ (uhol α). Tento postup, ako aj postup následného nájdenia tretej elipsy zobrazuje nasledujúci obrázok.



Obrázok 26: ilustrácia vyhľadávania dvojíc a trojíc elíps u algoritmu lokalizácie ŠPZ pomocou MSER

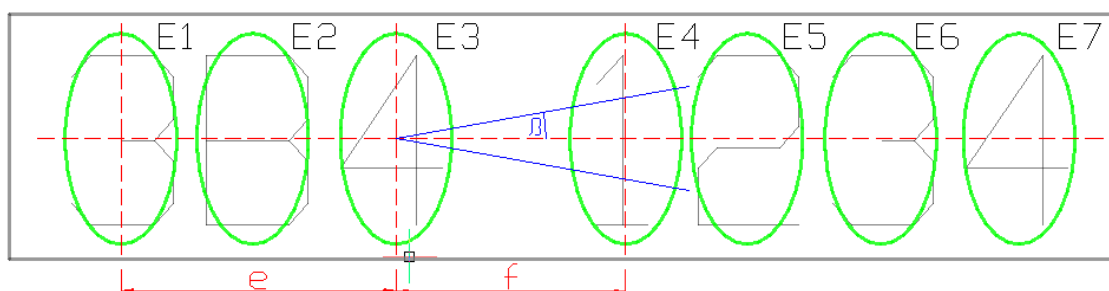
Takto získané dvojice elíps následne podrobíme testu, v ktorom k nim pripájame elipsu tretiu a tvoríme tak trojice. Spájanie prebieha na základe toho, že prvá elipsa z druhej dvojice musí byť rovnaká ako druhá elipsa z prvej dvojice (spojenie dvojíc S1-S2 a S2-S3). Geometrickým faktorom je v tomto prípade vzdialenosť medzi elipsami, ktorá musí byť $\pm 48\%$ rovnaká (pomer veľkostí $c:d$). Táto konštanta opäť pôsobí benevolentne, ale zahŕňa kritické prípady, kedy sú dve elipsy prvej dvojice blízko u seba a tretí znak vzdialený, ako je tomu napríklad u trojice znakov „1L7“ alebo „5P1“. V tomto bode máme vytvorených kandidátov na ľavú časť ŠPZ. Následne si vytvoríme ešte potenciálne štvorce, čo je výpočtovo pomerne nenáročne nakoľko sa vlastne jedná o spojenie dvoch trojíc ktoré majú dve zhodné elipsy (trojice E4-E5-E6 a E5-E6-E7). Týchto kombinácií je skutočné málo.



Obrázok 27: Ukážka nepomeru vzdialeností medzi znakmi

Posledným krokom k nájdeniu potenciálnych kandidátov je spojenie trojice a štvorice. Testovacou podmienkou je aby vzdialenosť medzi 1 a 2 elipsou z trojice a štvorice bola $\pm 28\%$ rovnaká (vzdialenosť E1-E2 E4-E5) a na základe geometrie aby vzdialenosť medzi poslednou elipsou trojice a prvou elipsou štvorice bola menšia ako 1,5 násobok

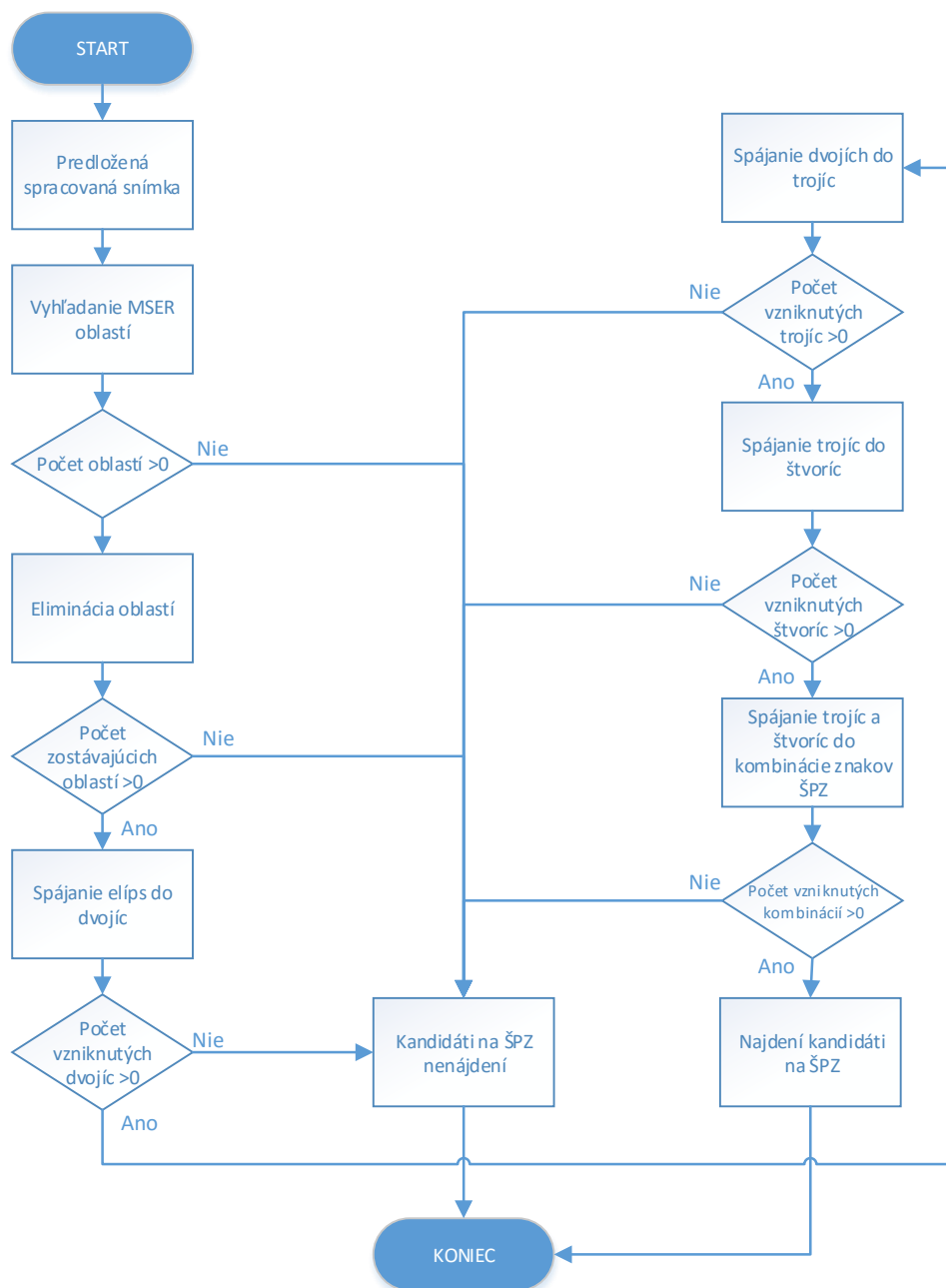
vzdialenosti medzi elipsami na krajoch trojice ($f < 1,5 * e$). Samozrejmosťou je, že prvá elipsa štvorice musí mať väčšiu x-ovú súradnicu ako posledná elipsa trojice.



Obrázok 28: Názorné zobrazenie geometrie algoritmu lokalizácie ŠPZ v obraze pomocou oblastí MSER

Na základe vyššie spomínaných geometrických vlastností a podmienok sú následne získaní kandidáti, ktorí by mohli vlastnosťami odpovedať ŠPZ. Aj keď sa eliminačnými krokmi spomínanými vyššie môže odstrániť až 90% nepotrebných oblastí u ktorých je 100% pravdepodobnosť, že sa nebudú nachádzať v oblasti znaku na ŠPZ, stále zostávajú aj oblasti, ktoré môžu vytvoriť falošné signály na ŠPZ. Jedná sa hlavne o snímky kde sa v pozadí nachádzajú budovy s homogénnym rozložením okien, poprípade vozovka, ak sa skladá z pravidelných dlaždíc, napríklad tzv. mačacích hláv. Je teda potrebné týchto falošných kandidátov eliminovať.

Táto eliminácia pozostáva z viacerých krokov. Z obrazového pohľadu na vec má ŠPZ charakteristické vlastnosti čo sa týka jas. ŠPZ je tvorená pomerom svetlých a tmavých bodov približne rovnakej jasovej úrovne. Na základe daného kritéria bude možné falošných kandidátov do istej miery odstrániť. Problémom je určiť hranicu medzi svetlými a tmavými pixelmi. Do istej miery to je možné v rámci danej galérie a daných svetelných podmienok, každopádne aj v rámci jednotlivých galérií sa nachádzajú tak nejednotné snímky, že nie je možné určiť hranicu tmavých a svetlých pixelov a následne vyšetrovať pomer medzi nimi.

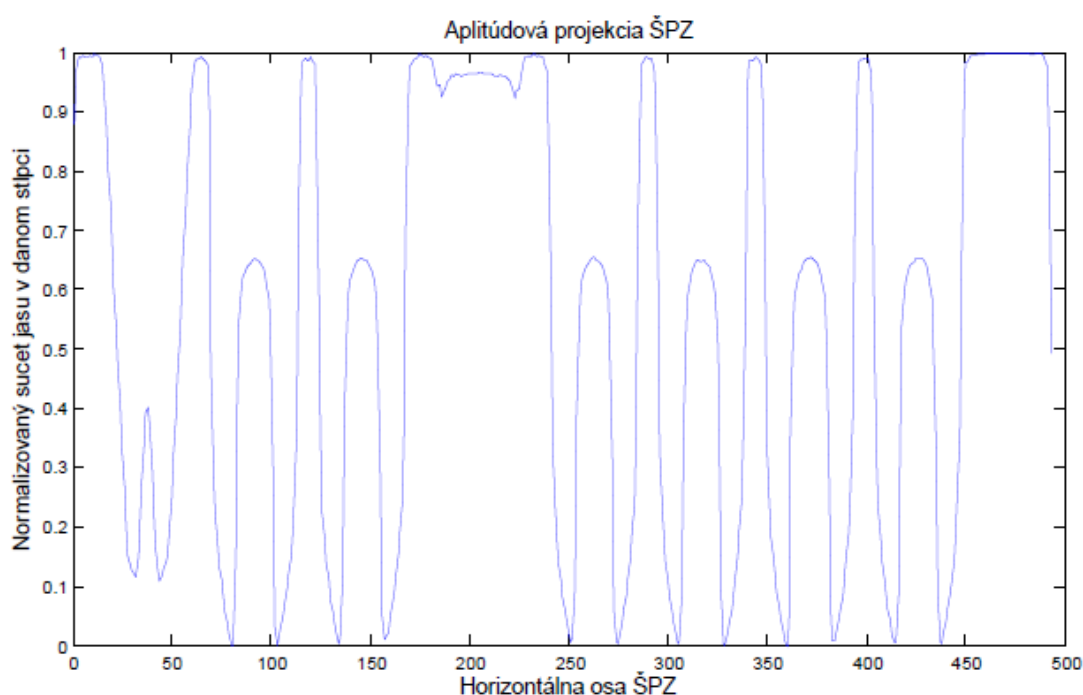


Obrázok 29: Vývojový diagram algoritmu vyhľadávania kandidátov na ŠPZ pomocou detektora MSER

Vytvorený algoritmus pracujúci so šedo tónovou snímkou si definíciu svetlého pixelu určuje pre každého kandidáta na ŠPZ sám tak, že si v istej vzdialenosti (3% celkovej horizontálnej veľkosti) od pravého okraja kandidáta na ŠPZ vyjme okno 5x5 pixelov z ktorého si spočíta priemernú hodnotu a túto prehlási za hodnotu bieleho pozadia ŠPZ. Táto hodnota sa nachádza niekde medzi 0 čo odpovedá čiernemu pixelu a hodnotou 255 zodpovedajúcou úplne bielemu pixelu. Táto hodnota je následne ponížená o 47 jasových úrovní a následne je vyhodnocovaný počet pixelov s jasom vyšším ako daná úroveň. Zníženie úrovne o konkrétnu konštantu vnáša do algoritmu viacej voľnosti a možnosť falošného rozpoznania ŠPZ. Navrhnutá bola na základe experimentov

s galériou, kde sa nachádzali snímky, kde časť ŠPZ bola v tieni a teda časť pixelov mala nižšiu jasovú úroveň. Následné vyhodnotenie je na základe pomeru svetlých pixelov k celkovému počtu pixelov a malo by byť menšie ako 80% ale väčšie ako 55%. Tieto konštanty boli opäť určené experimentálne u ŠPZ, kde sa nachádzalo množstvo znakov s veľkou ako sú napríklad B, 8, 6 alebo 9. Ďalším kritériom v rámci prvého stupňa testovania je kontrola hodnoty bieleho pixelu pozadia ŠPZ, ktorá musí byť väčšia ako 60, teda byť približne v horných $\frac{3}{4}$ rozsahu.

Druhým stupňom testovania je testovanie na priebeh ŠPZ. Jedná sa o svetlé pozadie, na ktorom sa nachádza 7 tmavých znakov. Nasledujúci obrázok ukazuje amplitúdovú projekciu ŠPZ zobrazenú na Obrázok 24 pri zanedbaní modrého prúžku na ľavej strane.



Obrázok 30: Amplitúdová projekcia ŠPZ do osy x

Jedná sa o techniku, ktorá sčítava hodnoty jasov pre dané pixely v danom stĺpci obrazu. Ako vidno z obrázku, bolo by náročné testovať na prítomnosť minim, nakoľko každý znak má iný priebeh. Z tohto dôvodu bol zvolený opačný prístup a to konkrétne vyšetrenie maxim, alebo inými slovami svetlých oblastí medzi znakmi. Na úvod je teda vytvorený priemet ŠPZ do osy x a ten je následne normalizovaný na hodnoty od 0 do 1. Experimentálne bola určená hodnota nad ktorou by sa mali nachádzať svetlé oblasti ako hodnota 0,735. Následne sú počítané prechody zo svetlých oblastí do oblastí tmavých čo odpovedá počtu znakov. Nasledujúca rovnica matematicky popisuje horizontálnu amplitúdovú projekciu.

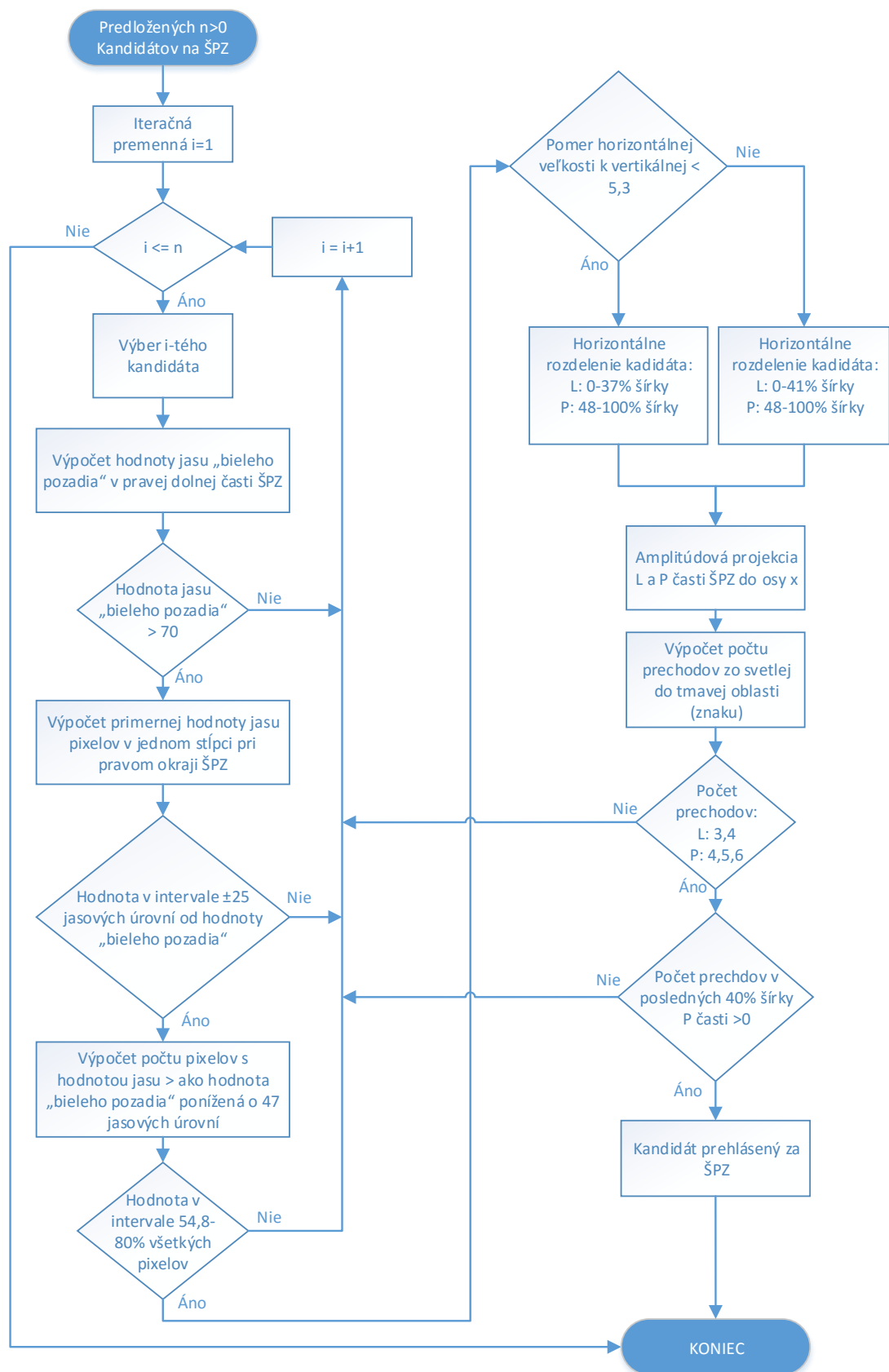
$$H(x) = \sum_{y=1}^Y f(x, y) \quad (32)$$

V danom algoritme nastáva problém pri vozidlách, ktoré sú fotené zozadu nakoľko sa tu nachádzajú nálepky o platnosti technickej a emisnej kontroly, ktoré sa v jasovej oblasti a princípe vytvoreného testu správajú ako znak. Preto bola ŠPZ rozdelená na dve časti a to konkrétne 0 – 37% horizontálnej vzdialenosti, čo odpovedá časti ŠPZ od ľavého okraja po nálepky a 48% - 100% horizontálnej vzdialenosti čo odpovedá ŠPZ od nálepky po pravý koniec. V ľavej časti sú povolené 3 alebo 4 prechody, v pravej časti 4 alebo 5 prechodov. Rozdelením ŠPZ sa ľavú a pravú časť sa v skutočnosti získalo ešte prísnejšie kritériu nakoľko už nie je postačujúci daný počet prechodov, ale aj ich čiastočné geometrické rozdelenie charakteristické pre oblasť ŠPZ.

Tretím stupňom testovania na falošných kandidátov je jednoduchá doplňujúca podmienka na svetlú oblasť medzi pravým horným a dolným bodom. Jedna sa o pomerne jednoduchý test s nie veľkou výpovednou hodnotou, ale na základe experimentov po vykonaní prvého stupňa sa ukázalo, že existujú náhodné prípady, kde sa v oblasti v ktorej sa vyhodnocuje hodnota svetlého pozadia nachádzala skutočne svetlá oblasť a aj pomer svetlých pixelov k celkovému počtu splnil podmienku. Tento prípad nastával napríklad na bielych nálepkách s textom na vozidlách. Vytvorený test spočíva vo vytvorení spojnice medzi dvoma definovanými bodmi a určenie priemernej hodnoty svetlosti všetkých bodov na danej spojnici. Tie body sa nachádzajú pod uhlom 45% od horizontálnej roviny a vzdialenosťou 2 pixelov od pravého okraja ŠPZ smerom do vnútra ŠPZ. Priemerná hodnota jasu na spojnici týchto bodov musí ležať v určitom intervale (± 25 jasových úrovní) okolo jasovej hodnoty určenej ako hodnota pozadia z prvého stupňa testovania. Vývojový diagram, ktorý stručne popisuje testovanie kandidátov je zobrazený na Obrázok 32.



Obrázok 31: Príklady ŠPZ s problémovým testovaním



Obrázok 32: Vývojový diagram testovania kandidátov na potenciálnu ŠPZ

Nakoľko sa daný algoritmus testovania pravosti kandidátov javil ako najslabší článok riešenia boli hľadané prostriedky ako ho vylepšiť. Bol využitý iteračný algoritmus, ktorý v každom kroku počíta jasovú úroveň popredia a pozadia a ako prah do ďalšej iterácie berie priemer týchto prahov. Cyklus pokračuje pokiaľ v dvoch cykloch za sebou nedôjde k zmene prahu. Algoritmus produkuje nasledujúce výsledky:



Obrázok 33: Vľavo ŠPZ z originálneho grayscale obrázku, vpravo kandidát spracovaný prahovaním, rozdelený na popredie a pozadie

Výsledkom daného algoritmu je snímok obsahujúci tri jasové úrovne:

- 0 pre body ležiace mimo obdĺžnika ŠPZ využívané len z dôvodu zobrazenia celej ŠPZ pomocou príkazu *roipoly*,
- 1 pre body popredia (znaky),
- 255 pre body pozadia.

Daný prvok bol k testovaniu bohužiaľ vytvorený až v záverečnej fáze tvorby diplomovej práce a tak nebol z časových dôvodov testovaný na celej galérii pre všetky implementované algoritmy. Na základe krátkych testov ale bolo možné pozorovať mierne zlepšenie hlavne v oblasti označenia falošného kandidáta vnútri ŠPZ, ktorý neobsahoval všetky znaky. Je na mieste ale poznamenať, že vzniknutá situácia neovplyvňuje výsledky práce. Avšak zdokonalenie poskytuje potenciál k získaniu všeobecne lepších výsledkov.

3.3.2 Vyhodnotenie

Názornú ukážku toho aké výsledky dosahuje MSER na daných snímkach ukazuje obrázok v kapitole 1.5, ktorá sa zaoberá detektorom MSER. Ako je vidieť, algoritmus pri daných parametroch, ktoré sú experimentálne nastavené tak, aby pokryli znaky ŠPZ na pomerne vzdialených vozidlách od objektívu fotoaparátu deteguje v obrázku o rozlíšení 800x600 pixelov veľmi veľa takýchto MSER oblastí. Na Obrázok 7 ich bolo konkrétne nájdených 1281. Hľadať vhodnú kombináciu algoritmom vyššie popísaným by okrem časovej náročnosti znamenalo aj množstvo falošných kandidátov. Tento problém je čiastočne odstránený popísanou elimináciou a v testovacom obrázku následne zostáva len 67 MSER oblastí, čo odpovedá len približne 5,5% všetkých pôvodných oblastí a predstavuje obrovské zlepšenie v oblasti výpočtovej náročnosti. Z týchto oblastí je v konečnom dôsledku na základe geometrických vlastností získaný jediný kandidát, ktorý je následne podrobený testom, či odpovedá ŠPZ a vyhodnotený ako ŠPZ.

Za daného nastavenia parametrov a využitím množstva for-cyklov v prostredí MATLAB nie je tento algoritmus vôbec optimálny a rýchly. Na danom testovacom

obrázku trvá samotné získanie MSER regiónov príkazom detectMSERFeatures približne 2,17 sekundy, následná úprava a dopočet potrebných vlastností pre všetky MSER regióny (prekopírovanie hodnôt z premennej typu MSERRegions do štandardnej matice typu float, zistenie jasovej úrovne pre daný región z dôvodu eliminácie svetlých oblastí) a eliminácia na základe geometrických vlastností trvá približne rovnako ako získanie daných oblastí. Po tejto eliminácii zostáva už len 80 oblastí a následné odstránenie oblastí, ktoré majú počiatok v oblasti rovnakej ako iný región, ale obsahujú menší počet pixelov odstráni už len 13 regiónov a má trvanie približne 0,49 sekúnd. Následný jadrový algoritmus hľadania kandidátov a ich overenie trvá 0,28 sekundy. Tieto časové údaje sú neprípustne vysoké, pre systém podobného zaradenia v praxi. Faktom ale ostáva, že v prípade jednotnej galérie snímok, povedzme z kamerového systému na cestách poprípadne závary na parkovisku je možné dosť presne predpokladať veľkosť ŠPZ v danom obraze. To upresní parametre samotného príkazu na získanie MSER regiónov a už v prvom kole budeme schopný pracovať s počtom regiónom medzi 100 a 200, po následnej eliminácii sa môžeme pohybovať na úrovni 20-30 oblastí, ktoré by mohli vyhovovať znaku na ŠPZ, individuálne od úlohy akú by sme riešili. Pri snímaní automobilov na ceste z mosta prípadne návesti nad vozovkou bude pozadie obrázku (vozovka) homogénna oblasť, kde nebudú detegované žiadne oblasti MSER. V tomto prípade budú tie oblasti skutočne detegované na samom aute a nebude ich veľký počet.

Veľkou výhodou daného algoritmu je fakt, že jeho surovým výstupom nie je obdĺžnik, ktorý reprezentuje ŠPZ, ale 7 samotných znakov, ktoré môžu byť následne veľmi rýchlo podrobené algoritmu rozpoznávania znakov. Medzi jeho nevýhody patrí to, že nefunguje univerzálne na všetky typy ŠPZ. Algoritmus bol navrhnutý len na ŠPZ skladajúce sa z trojice a štvorice znakov. Nie je teda funkčný na ŠPZ zahraničných účastníkov ako ani na české ŠPZ starého typu obsahujúce trojicu písmen a následne dve dvojice znakov oddelené pomlčkou. To je bohužiaľ veľká limitácia, ktorá znemožňuje jeho univerzálne použitie napríklad na detekciu ŠPZ na vozidlách na vozovke. Systém by musel byť následne vybavený nejakým ďalším stupňom, ktorý by spracovával neštandardné snímky, kde nebola ŠPZ detegovaná.



Obrázok 34: Príklady ŠPZ kde algoritmus lokalizácie pomocou MSER zlyháva

Obrázok 34 zobrazuje niekoľko príkladov ŠPZ, kde algoritmus zlyhal a ŠPZ nebola lokalizovaná. Algoritmus principiálne zlyháva na poškodených ŠPZ, na ŠPZ kde je časť oblasti prekrytá (napríklad ťažné zariadenie) a ŠPZ s neštandardnou distribúciou znakov. V galérii sa nachádzalo aj niekoľko snímok kde je ŠPZ neštandardne prichytená a znaky nie sú detegované ako MSER oblasti pretože nie sú jednotnou tmavou oblasťou na svetlom pozadí. Príkladom je ŠPZ v pravom dolnom rohu ilustračného obrázka. Štatistické výsledky na jednotlivých galériách sú uvedené v nasledujúcej tabuľke.

MSER	Vyhodnotenie:		Parametre:	
	FP	TP	MaxArea	ThresholdDelta
Real1	0,0%	97,3%	0,050	1,90
Real2	0,0%	97,4%	0,050	1,90
MPOV_1	0,0%	93,4%	0,180	1,90
MPOV_2	0,0%	73,0%	0,180	1,90
MPOV_3	0,0%	94,8%	0,180	1,90
db_destivo	0,0%	92,4%	0,030	1,17
db_park_slunecno	0,0%	79,5%	0,018	1,50
db_park_zatazeno	0,9%	97,2%	0,100	1,80
db_slunecno	0,0%	78,6%	0,180	1,40

Tabuľka 1: Úspešnosť a parametre algoritmu využívajúceho MSER

Problémom, pre daný detektor sú snímky, kde dopadá slnečné žiarenie priamo na ŠPZ a znaky sú úplne neviditeľné z dôvodu odrazu od lesklého povrchu ŠPZ popri prípade kontrast medzi znakom a svetlou oblasťou je tak malý, že detektor by musel byť nastavený extrémne citlivo a aj pri danom nastavení existujú prípady, kedy daný znak detegovaný nebude. Toto nastavenie by samozrejme zvyšovalo už teraz vysoký počet detegovaných oblastí. Z tohto dôvodu je úspešnosť algoritmu na najslnečnejšej galérii nižšia, približne 79% a aj za daných okolností trvá beh algoritmu na niektorých snímkach viac ako 1 minútu. Naopak najvyšší výsledok vykazuje metóda na galérii so zaťaženým počasím ale dostatkom svetla. Na týchto snímkach je väčšinou pomerne dobrý kontrast medzi tmavým znakom a pozadím, ktoré je síce tmavšie ako za štandardných svetelných podmienok, každopádne stále poskytuje dostatočný prechod kontrastu.



Obrázok 35: Príklad neštandardne osvetlenej ŠPZ kde nie je možné detektorom MSER detegovať všetky znaky

Záverom je vhodné zhodnotiť, že pri predom známych podmienkach, je možné algoritmus nastaviť tak, aby fungoval veľmi spoľahlivo a rovnako rýchlo. Princíp algoritmu samozrejme nie je stavaný na real-time aplikácie a neoptimalizovaný MATLAB kód opäť uberá na efektivite. Ako aj pri ostatných implementovaných algoritmoch aj tu platí, že výsledky by mohli byť zlepšené vhodnou úpravou snímok pred samotným vyhľadávaním oblastí.

3.4 Lokalizácia ŠPZ použitím Harrisovho detektora

3.4.1 Popis algoritmu

Algoritmus, ktorý bol popísaný v kapitole 3.3.1 bol postavený pre detektor MSER, ktorý v obraze vyhľadáva významné oblasti (v angličtine blobs). Geometrické vlastnosti, na základe, ktorých je možné získať kandidáta na ŠPZ sú úzko špecifikované k tomu, čo daná metóda v obraze deteguje. Druhým z principiálnych riešení, ktoré budú v rámci tejto práce testované bude vyhľadávanie ŠPZ na základe detekcie 4 rohových bodov. Tento algoritmus bude mať najpočetnejšie využitie čo sa týka množstva detektorov, ktoré bude na jeho základe možné testovať. Algoritmus je teda postavený na testovanie rohových detektorov (corner detector) a jeho vstupom sú pre všetky testované metódy súradnice detegovaných významných bodov. Je dôležité poznamenať, že nižšie popísaný princíp hľadania ŠPZ na základe 4 rohových bodov je spoločný pre Harrisov detektor, FAST detektor, Shi-Tomasi detektor a v geometrickom princípe aj pre deskriptor BRISK, ktorý má implementovaný vlastný detektor a je bližšie popísaný v príslušnej kapitole.

V tejto časti bude popísaný implementovaný algoritmus vyhľadávania tejto špecifickej kombinácie 4 rohových bodov. Pre registračnú tabuľku popísanú pár kapitol vyššie platia isté pravidlá čo sa rozmerov týka a tie sa úspešne využívajú k nájdeniu tej správnej kombinácie významných bodov. Pre túto registračnú tabuľku platí, že jej rozmery sú 520x110 milimetrov, takže pomer horizontálnej veľkosti, k veľkosti vertikálnej je približne 4,73. Daný algoritmus je ale vhodný aj pre detekciu staršieho typu poznávacej značky, konkrétne tej s rovnakými rozmermi, ale bez modrého prúžku Európskej únie ako aj historického typu s trojicou znakov a dvoma dvojicami oddelenými pomlčkou o ktorom bola zmienka v kapitole vyššie.



Obrázok 36: Požadovaný výsledok algoritmu lokalizácie ŠPZ pomocou 4 rohových bodov

Problémom ale je, že u ŠPZ s modrým prúžkom väčšina rohových detektorov deteguje rohy na prechode z modrého prúžku do svetlého pozadia, nakoľko sa v tých bodoch jedná o väčšiu jasovú zmenu ako v bodoch skutočných ľavých rohov ŠPZ, kde dochádza z prechodu z čierneho rámčeka na modré pozadie. Druhým poznatkom je, že historický typ ŠPZ je rozmerovo o niečo menší, než aktuálne typy ŠPZ. To vedie k záveru, že je potrebné nastaviť parametre pomeru horizontálnej veľkosti k veľkosti vertikálnej mierne benevolentnejšie. Ďalším faktom ku ktorému je potrebné pri tvorbe algoritmu prihliadať je ten, že väčšina fotografií je vyfotená z výšky väčšej ako je výška v ktorej sa nachádza samotná ŠPZ a teda tu dochádza k efektu perspektívy a časti ŠPZ si nemusia plne rozmerovo odpovedať.

Samotný algoritmus si berie ako vstup súradnice významných bodov detegovaných ktoroukoľvek z vyššie spomínaných metód rohovej detekcie. V tomto odstavci sa čo sa výsledkov týka budeme zameriavať na Harrisov detektor.

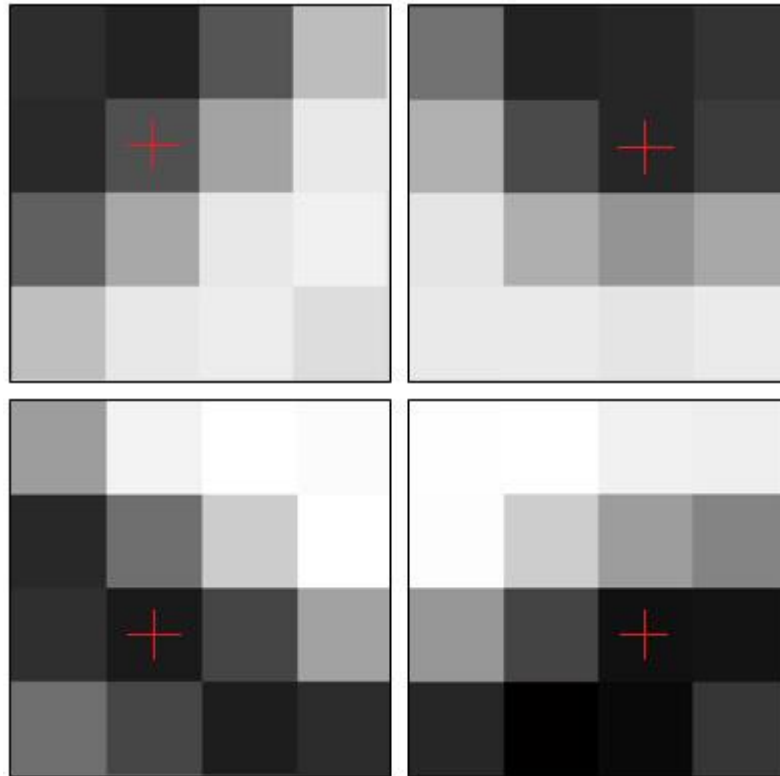
Prvým krokom riešenia, ktorý následne mnohonásobne zjednoduší výpočet hľadania príhodnej kombinácie 4 bodov je algoritmus nazývaný Non-Maxima Suppresion, pomocou ktorého je možné potlačiť významné body nachádzajúce sa blízko lokálneho maxima. Tento algoritmus je možné použiť u Harrisovho detektora ako aj u detektora Shi-Tomasi pre ktoré je možné v prostredí MATLAB získať maticu „rohovatosti“ (z anglického *cornerness*) pre každý bod v obraze. Obrovskou nevýhodou je, že výpočet „rohovatosti“ prebieha v podstate dva krát, prvý krát v rámci príkazu na získanie významných bodov (*detectHarrisFeatures*, *detectMinEigenValues*) a druhý krát umelo pomocou príkazu *cornermetric* s parametrom metódy. Do budúcnosti by zrejme bolo vhodné, keby jedným z parametrov príkazu na získanie významných bodov bolo nastavenie oblasti v ktorej je možné detegovať maximálne jeden významný bod teda lokálne maximum v danej oblasti. Je možné, že týmto postupom by boli stratené niektoré body, ktoré vykazujú vyššiu mieru „rohovatosti“ ako body, ktoré by stále ostali detegované. Faktom ale ostáva, že pre niektoré aplikácie by to našlo uplatnenie a zjednodušilo a zrýchlilo by to nasledujúci výpočet. V prípade tejto diplomovej práce bol algoritmus non-maxima suppresion implementovaný samostatne a to tak, že po výpočte matice „rohovatosti“ pre všetky body, sú pre jednotlivé body priradené sledované hodnoty ich „rohovatosti“. Na základe zadného parametru *border* sú následne vo štvorcovej oblasti o veľkosti *border* na všetky smery od vyšetrovaného významného hľadané iné významné body, z ktorých sa zachováva práve ten, ktorý má najvyššiu hodnotu „rohovatosti“. Tento krok síce znižuje počet významných bodov a môže byť úspešne využitý v prípade, že vieme explicitne aspoň približnú veľkosť ŠPZ. Pri testovaní sa totiž veľmi často stávalo, že pri snímkach s nižším rozlíšením boli v rámci okna 7x7 pixelov v ľavom dolnom rohu detegované dva body: bod, ktorý je na prechode z modrej oblasti na svetlé pozadie a odpovedá skutočnému požadovanému rohu a bod, ktorý na nachádzal na pravej časti písmena „Z“ v kóde krajiny v modrom prúžku a vykazoval väčšiu hodnotu „rohovatosti“ ako skutočný rohový bod ŠPZ. Z tohto dôvodu,

je parameter *border* nastavený na maximálne hodnotu 2 a dochádza k zníženiu efektívnosti daného algoritmu.

Pre zrýchlenie algoritmu sú ďalej odstránené body nachádzajúce sa na okrajoch snímku. Pri charaktere testovacej galérie, nie je možná dostatočná eliminácia, pretože ŠPZ sa v obraze nachádza v ľubovoľných častiach. V prípade riešenia konkrétnej úlohy by bolo možné tento krok realizovať už v počiatku, kde by bola vybraná časť obrazu, región záujmu, v ktorom by sme následne hľadali významné body a nebola by potrebná ďalšia eliminácia.

Následne sú zostávajúce významné body zoradené tak, aby v matici v ktorej sú uložené postupovali postupne z ľavého horného rohu nadol a následne o jeden stĺpec pixelov doprava a opäť nadol a podobne až kým nedorazíme do pravého dolného rohu. Tento krok nemá žiadny funkčný význam, jedná sa len o zjednodušenie následného výpočtu horizontálnej popríklad vertikálnej vzdialenosti dvoch bodov, kedy na základe ich poradového čísla v matici všetkých detegovaných bodov vieme určiť ktorý z bodov sa nachádza viac vľavo, popríklad vyššie v obraze. Ďalším krokom, by malo byť nájdenie kandidátov na dvojicu bodov zodpovedajúcich horným rohom ŠPZ. Vyhľadávať tieto dvojice medzi všetkými kombináciami bodov by bolo opäť výpočtovo náročné a preto bolo potrebné prísť so spôsobom ako eliminovať body o ktorých je zrejmé, že nemôžu danými bodmi v skutočnosti byť. Ako efektívne sa javí akési maskovanie alebo testovanie najbližšieho okolia daného bodu. Pre každý významný bod sú kontrolované hodnoty jas pixelov na obidvoch diagonálach. Hodnoty sú porovnávané s istou prahovou hodnotou, ktorú definuje parameter *prah_maskovanie*. V prípade, že hodnoty vo všetkých 4 smeroch menšie ako daná prahová hodnota je vyšetřovaný bod úplne vylúčený. Zvyšné body sú následne rozdelené do minimálne jednej zo štyroch kategórií:

- Kandidát na ľavý horný roh (LH): pixel vpravo dole od vyšetřovaného má hodnotu jas vyššiu ako prah
- Kandidát na ľavý dolný roh (LD): pixel vpravo hore od vyšetřovaného má hodnotu jas vyššiu ako prah
- Kandidát na pravý horný roh (PH): pixel vľavo dole od vyšetřovaného má hodnotu jas vyššiu ako prah
- Kandidát na pravý dolný roh (PD): pixel vľavo hore od vyšetřovaného má hodnotu jas vyššiu ako prah

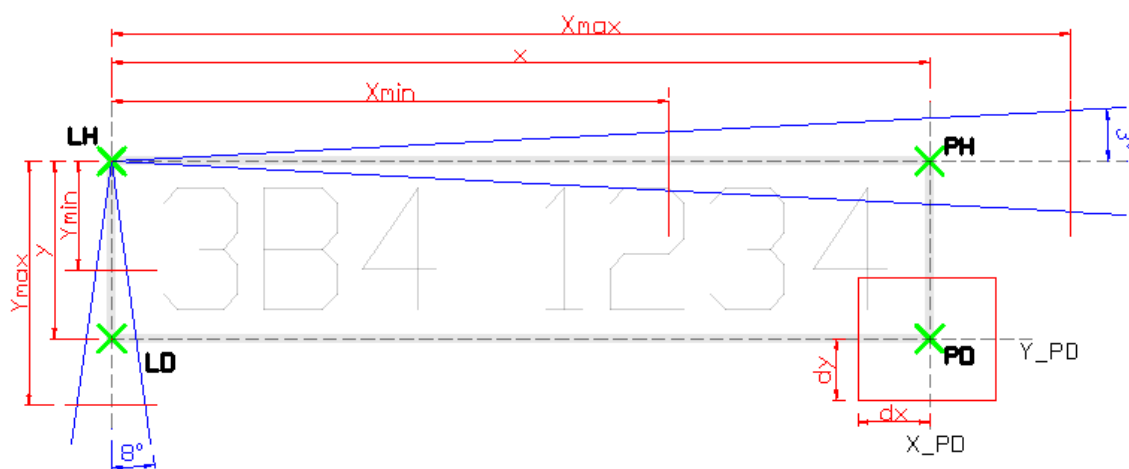


Obrázok 37: Ukážka kandidátov pre jednotlivé rohy ŠPZ: červené body reprezentujú detegované významné body. V prvom rade sú zobrazení kandidáti na ľavý horný a pravý horný roh, v spodnom rade kandidáti na ľavý dolný a pravý dolný roh ŠPZ

Nasledujúci popis algoritmu je vyobrazený na Obrázok 38. Prvým krokom je vyhľadávanie dvojíc bodov, ktoré sú potenciálnymi kandidátmi na horné rohy poznávacej značky (1. a 3. kategória). Pre tieto dva body by malo platiť, že uhol ktorý zvierajú spojnice týchto dvoch bodov a horizontálna rovina je v absolútnej hodnote menší ako 3 stupne, čo je postačujúce pre snímky, na ktorých sa má nachádzať ŠPZ vo vodorovnej rovine. Doplňujúcimi podmienkami sú podmienky na vzdialenosť medzi bodmi. Tá je počítaná pomocou Pytagorovej vety ako prepona trojuholníka. Vzdialenosť musí byť väčšia ako 50 pixelov (X_{min}), pretože to by odpovedalo horizontálnej veľkosti približne 10 pixelov, čo je považované za minimum potrebné pre následné rozpoznávanie znakov. Rovnako je tu požiadavka na to, aby táto vzdialenosť nebola väčšia ako jedna tretina celkovej horizontálnej veľkosti obrázku (X_{max}), nakoľko sa predpokladá, že ŠPZ nebude na šírku zaberat' viac ako 1/3 obrázku. Kandidáti, ktorí vyhovujú týmto podmienkam sú uložení a pristupuje sa k vyhľadávaniu tretieho bodu.

Tretí bod je vyhľadávaný tak, že pre všetky body, ktoré by mohli tvoriť ľavý horný a ľavý dolný roh (1. a 2. kategória) sú spočítané vzdialenosti a uhly, ktoré dané body zvierajú. Pre všetky kombinácie LH-PH sú následne hľadané všetky kombinácie LH-LD, za predpokladu, že sa ľavý horný roh zhoduje. Ďalšími podmienkami sú, že uhol ktorý zvierajú spojnice bodov LH a LD s vertikálnou rovinou musí byť menší ako 8° a zároveň musí byť $\pm 4^\circ$ rovnaký, ako uhol ktorý prislúcha k bodom LH a PH.

Poslednou podmienkou je, že horizontálna vzdialenosť musí byť $\pm 23\%$ 5-násobkom vertikálnej vzdialenosti ($x:Y_{min}$, $x:Y_{max}$). V tomto bode je opäť potrebné pripomenúť, že táto vysoká miera tolerancie je na základe pokrytie celého portfólia typov ŠPZ ako bolo už vyššie v rámci tejto kapitoly spomenuté. Kritickým bodom tohto kroku je práve výpočet uhlu prislúchajúcemu k LH a LD bodu. V prípade, že je v snímku potenciálna ŠPZ u ktorej je vertikálna veľkosť na hranici použiteľnosti (tj. 10 pixelov), znamená „ustrelenie“ významného bodu o 1 pixel v horizontálnom smere veľkosť uhla $5,7^\circ$ a daná kombinácia nemusí vyhovieť podmienkam. Posledným krokom je nájdenie 4. bodu, konkrétne PD. Pre tento bod je možné nastaviť pomerne kritické podmienky a slúži ako verifikácia, či sa skutočne jedna o oblasť geometricky podobnú ŠPZ. Lokality, kde by mal tento bod ležať určíme na základe x-ovej súradnice potenciálneho PH rohu a y-ovej súradnice LD rohu (X_{PD} , Y_{PD}). Pre potenciálnu oblasť navyše zvažujeme veľkosť posunutia (v pixeloch) v x-smere medzi LH a LD a y-smere medzi LH a PH. Takto určenému bodu následne priradíme tolerancie o veľkosti 5% vertikálnej vzdialenosti medzi bodmi LH a LD (dy) a 5% horizontálnej vzdialenosti medzi LH a PH (dx). Tieto tolerancie sú zaokrúhlené na celý pixel smerom nahor, čo je esenciálne hlavne pre vertikálnu vzdialenosť u najmenších ŠPZ. Celý geometrický význam algoritmu je ilustračne zobrazený na nasledujúcom obrázku.



Obrázok 38: Názorné zobrazenie geometrie algoritmu lokalizácie ŠPZ v obraze pomocou 4 rohových bodov

Postup nastavovania tolerancií vychádza z množstva vykonaných experimentov a analyzovaných snímok. Parametre bohužiaľ nemôžu byť nastavené príliš prísne, nakoľko je celá galéria snímok získaná fotením ručne, bez statívu a je do nej zanesený prvok náhody. To spôsobuje fakt, že v snímku je na základe daných geometrických vlastností nájdených viacero oblastí, ktoré odpovedajú rozmerovo ŠPZ a je potrebné ich eliminovať. Množstvo týchto oblastí vzniká priamo vnútri danej ŠPZ, kde sú významné body detegované na ostrých rohoch znakov. Ako prvý krok eliminácie sa ukázalo ako vhodné odstrániť kandidátov, ktorí v rámci plochy ktorú ohraničujú obsahujú menej ako

20 významných bodov. Experimenty ukázali, že pri daných nastavenia detektoroch je tento počet významných bodov na znakoch vnútri ŠPZ s prehľadom splnený a tieto body vždy odpovedajú niektorej zo 4 kategórií ako kandidátne body na niektorý z rohov a teda nie sú na začiatku algoritmu eliminované. Ďalší stupeň testovania je už ale distinguishingovanejší. Jedná sa o tri stupne testovania, ktoré boli už popísané v rámci kapitoly 3.3.1 a ostávajú pre metódu hľadania 4 rohových bodov v platnosti.

3.4.2 Vyhodnotenie

Harrisov detektor napriek svojej pomernej jednoduchosti funguje na danej lokalizačnej úlohe relatívne spoľahlivo. Pre galériu Real1 a Real2 bolo možné nájsť nastavenie parametrov, ktoré zaručilo, že významné body (rohy) budú nájdené. Nakoľko sú v galériách snímky s rôzne veľkými ŠPZ, sú tieto parametre nastavené značne voľne. Na niektorých špecifických snímkach (biele vozidlo s množstvom tmavých prvkov, biele vozidlo s reklamnými nápismi) to znamená detegovanie viac ako 1000 významných bodov, čo značne spomaľuje výpočet a zvyšuje šance na false-positive výsledok. Na nasledujúcom obrázku sú uvedené príklady ŠPZ, kde algoritmus zlyháva.



Obrázok 39: Príklady snímok kde zlyháva lokalizácia ŠPZ pomocou Harrisovho detektora

Ako je možné vidieť z názornej ukážky, algoritmus principiálne zlyháva tam, kde neexistujú dostatočné ostré rohy ŠPZ, nakoľko je spodná časť rámčeka ŠPZ v bielom prevedení. V týchto prípadoch dochádza k zlyhaniu, nakoľko 4 potrebné body v obraze detegované nie sú. Najnáchylnejší na zlyhanie je ľavý dolný bod, kde sa nachádza prechod z bieleho pozadia na modrý pruh Európskej únie.

Harris	Vyhodnotenie		Parametre:	
	FP	TP	FilterSize	MinQuality
Real1	1,8%	91,1%	7	0,0018
Real2	1,3%	96,1%	7	0,0018
MPOV_1	8,4%	57,7%	5	0,0400
MPOV_2	13,5%	40,5%	5	0,0080
MPOV_3	5,1%	79,5%	7	0,0060
db_destivo	6,0%	17,9%	5	0,1300
db_park_slunecno	13,3%	48,1%	5	0,0150
db_park_zatazeno	8,0%	57,1%	7	0,0080
db_slunecno	11,1%	7,4%	7	0,0015

Tabuľka 2: Úspešnosť a parametre algoritmu využívajúceho Harrisov detektor

Ďalšími prípadmi kde algoritmus zlyháva z dôvodu neprítomnosti všetkých 4 významných rohov sú ŠPZ, ktoré vykazujú istú formu deformácie napríklad ohnuté rohy ŠPZ. Tieto snímky sú ale považované za neštandardné, no daný detektor voči týmto obmedzeniam nie je robustný.

Druhým typom zlyhania je zlyhanie false-positive. Po analýze snímok kde algoritmus lokalizoval ŠPZ na inom mieste ako v skutočnosti bola bolo zistené, že v drvivej väčšine týchto snímok boli rohové body detegované, ale testom na ŠPZ prešiel iný kandidát. Nejedná sa teda o principiálne zlyhanie detektora, ale nedokonalosť algoritmu testovania potenciálnych kandidátov. Špecifickým prípadom daného zlyhania sú snímky svetlých vozidiel s tmavými reklamnými nápismi. V týchto oblastiach je detegované veľmi veľké množstvo významných bodov a teda veľmi jednoducho je tu nájdený kandidát na základe geometrického rozdelenia týchto bodov. Títo kandidáti nakoniec vyhovujú všetkým podmienkam na ŠPZ, pretože sa jedná o tmavý text na svetlom pozadí, značne podobný priebehu ŠPZ. Za tieto prípady prikladám dvoch zástupcov.



Obrázok 40: Príklady snímok, kde je detegované množstvo falošných kandidátov, ktorí úspešne prejdú testom na vlastnosti ŠPZ

3.5 Lokalizácia ŠPZ použitím Shi-Tomasi detektora

3.5.1 Popis algoritmu

Ďalším z testovaných detektorov je detektor vlastných hodnôt, teda detektor Shi-Tomasi. Detektor funguje na rovnakom princípe ako Harrisov detektor, ktorý bol popísaný v kapitole 1.4. Podstatnou zmenou je to, že tento detektor využíva iný systém vyhodnotenia významných bodov. Pri Harrisovom detektore sme sa stretli s tým, že neboli priamo počítané vlastné hodnoty, ale významné body boli vyhodnocované pomocou výpočtu determinantu a stopy matice. Jianbo Shi a Carlo Tomasi z amerických univerzít v Cornell a Stanford navrhli iný spôsob vyhodnocovania, konkrétne priamym výpočtom vlastných hodnôt. Výpočet matice R (pri Harrisovom detektore nazývanej Harrisova matica) je nasledovný:

$$R = \min(\lambda_1, \lambda_2) \quad (33)$$

Algoritmus je opäť implementovaný v MATLAB-e a ako zlepšenie od Harrisovho detektora je pre účely práce zaujímavé detektor otestovať na úspešnosť riešenia úlohy lokalizácie ŠPZ. Samotný algoritmus vyhľadávania ŠPZ je totožný s algoritmom popísaným v kapitole 3.4.1 a ilustrovanom na Obrázok 38.

3.5.2 Vyhodnotenie

Zo svojho princípu je detektor Shi-Tomasi komplexnejšou a prepracovanejšou verziou Harrisovho detektora. Čo sa týka výsledkov, je úspešnosť týchto algoritmov na približne rovnakej úrovni a obidva zlyhávajú v rovnakých situáciách, ktoré popisujú Obrázok 39 a Obrázok 40 v kapitole 3.4.2. Najčastejším zlyhaním je zlyhanie z dôvodu absencie rohových bodov. Druhým z pozorovaných faktov je ten, že u detektora Shi-Tomasi je vyšší podiel „false positive“ výsledkov. To môže byť spôsobené nesprávnym nastavením parametrov kedy je detegované množstvo bodov, ktoré by na základe kvality mohli byť odstránené ešte pred odstránením stabilnejších bodov v oblasti ŠPZ. Druhou možnosťou je nedostatočné predspracovanie obrazu a nekvalitné testovanie na pravosť kandidátov.

Shi-Tomasi	Vyhodnotenie:		Parametre:	
	FP	TP	FilterSize	MinQuality
Real1	4,1%	90,4%	7	0,04
Real2	0,7%	95,4%	7	0,04
MPOV_1	11,3%	50,7%	5	0,12
MPOV_2	21,6%	27,1%	9	0,06
MPOV_3	7,7%	74,9%	7	0,10
db_destivo	12,1%	10,6%	5	0,06
db_park_slunecno	10,7%	58,0%	7	0,08
db_park_zatazeno	12,5%	68,8%	7	0,12
db_slunecno	29,6%	7,4%	7	0,11

Tabuľka 3: Úspešnosť a parametre algoritmu využívajúceho detektor Shi-Tomasi

3.6 Lokalizácia ŠPZ použitím detektora FAST

3.6.1 Popis algoritmu

Posledným z testovaných rohových detektorov je detektor FAST bližšie popísaný v kapitole 1.6. Jedná sa o pomerne jednoduchý ale zato účinný detektor, ktorý nachádza uplatnenie v mnohých aplikáciách a preto je ďalším kandidátom na testovanie úlohy lokalizácie ŠPZ. Nakoľko sa opäť jedná o detektor rohov, bude testovací algoritmus zhodný s tým prezentovaným u Harrisovho detektora a použitého aj pre Shi-Tomasi detektor. Vstupom do algoritmu sú teda detegované významné body, ktoré sú podrobené geometrickej analýze, na základe ktorej sú vyhľadani kandidáti na ŠPZ.

3.6.2 Vyhodnotenie

Nasledujúca tabuľka zhrňuje úspešnosť detektoru FAST. Algoritmus všeobecne vykazuje nižšiu úspešnosť ako vyššie testované algoritmy. Tento výsledok vychádza z princípu daného algoritmu, ktorý pri úlohe lokalizácie ŠPZ pracuje na hrane svojich detekčných možností a má veľmi blízko k zlyhaniu. Algoritmus totiž za rohový bod považuje iba taký bod, kedy na kružnici vo vzdialenosti 3 pixelov od daného vyšetřovaného bodu, leží vedľa seba aspoň 12 zo 16 pixelov s nižšou (vyššou) jasovou úrovňou ako daný vyšetřovaný pixel. To odpovedá presne 270° a preto by algoritmus pri dokonalej ŠPZ, to znamená bez pootočenia a bez perspektívy fungoval presne na hrane, t.j. rohy ŠPZ by boli detegované. Na reálnych snímkach pomerne často nastáva situácia, kedy sa vnútorný uhol ŠPZ javí väčší ako 90° a teda významný bod v danom mieste už nie je detegovaný. Riešením pre danú úlohu by bola implementácia podobného algoritmu, kde by bolo sledovaných len 10 susediacich bodov na kružnici, čo by odpovedalo uhlu 225° . Táto úprava by znamenala väčší počet detegovaných rohov s nižšou kvalitou, ale pokryla by rohy ŠPZ a navyše ponúkla priestor na vyhľadávanie mierne pootočených ŠPZ. Znamenala by však vyššiu výpočtovú náročnosť, pretože pôvodný algoritmus nevyžadoval testovanie všetkých bodov na kružnici k vylúčeniu kandidáta, ale postačovala analýza konkrétnych bodov ako bolo popísané v príslušnej kapitole.

FAST	Vyhodnotenie:		Parametre:	
	FP	TP	MinQuality	MinContrast
Real1	7,3%	81,7%	0,001	0,09
Real2	1,9%	82,9%	0,001	0,09
MPOV_1	16,9%	42,6%	0,010	0,11
MPOV_2	13,5%	24,3%	0,010	0,11
MPOV_3	6,4%	73,1%	0,010	0,12
db_destivo	15,0%	46,3%	0,010	0,15
db_park_slunecno	5,8%	47,4%	0,010	0,14
db_park_zatazeno	10,7%	57,1%	0,010	0,20
db_slunecno	22,2%	14,8%	0,010	0,13

Tabuľka 4: Úspešnosť a parametre algoritmu využívajúceho FAST detektor

3.7 Lokalizácia ŠPZ použitím deskriptora BRISK

3.7.1 Popis algoritmu

Poslednou z metód, ktorá prakticky pre lokalizáciu ŠPZ používa 4 rohové body je lokalizácia pomocou deskriptora BRISK. Výsledky boli opäť získané pomocou MATLAB-u, funkcia *detectBRISKFeatures* je ale zahrnutá až od verzie 2014a vyššie. Jadro algoritmu opäť spočíva vo vyhľadávaní kombinácia 4 rohových bodov v špecifickom geometrickom rozložení ako to už bolo popísané v kapitole 3.4.1. Nakoľko sa ale nejedná o samostatný detektor, ale deskriptor, poskytuje daný algoritmus istú pridanú hodnotu. Pre účely lokalizácie sa údaj o mierke (scale) javí ako pomerne použiteľný, nakoľko je pre 4 rohové body viac-menej zhodný.



Obrázok 41: Požadovaný výsledok pri lokalizácii ŠPZ pomocou deskriptora BRISK

Obrázok 41 ukazuje požadovaný výsledok pre daný algoritmus. Oproti algoritmu, ktorý bol implementovaný pre Harrisov detektor, je navyše využitý spomínaný údaj o mierke. Pri výpočte horizontálnych a vertikálnych kombinácií sú uvažované iba kombinácie pre ktoré platí, že veľkosti mierok sa líšia o maximálne $\pm 20\%$. BRISK je ale oveľa viac náchylnejší k posunutiu deklarovaného významného bodu od skutočného rohu ŠPZ a preto bolo znížené z nárokov na geometrické vlastnosti. Experimenty ukázali, že najprísnejšie kritéria boli zvolené pre 4 bod, ktorého sa teda dotklo mierne uvoľnenie kritérií.

3.7.2 Vyhodnotenie

Algoritmus bol opäť podrobený testovaniu na priložených galériách. Daný deskriptor na danej úlohe vykazuje všeobecne nižšiu úspešnosť ako rohové detektory, ktoré boli na danej úlohe princípom 4 rohových bodov testované. Na nasledujúcom obrázku je napríklad možné vidieť oblasť ŠPZ, kde chýba jeden z rohových bodov. Na danom snímku s rozlíšením 800x600 pixelov bolo detegovaných 497 oblastí. Možnosťou by bolo zníženie hodnoty parametrov *MinQuality* alebo *MinContrast*, ktoré by ale viedlo k značnému zvýšeniu detegovaných bodov. Algoritmus už pri danom nastavení v rôznorodých galériách nachádzal snímky, kde jeho beh trval vyše minúty.



Obrázok 42: Príklad snímku kde nie je detegovaný jeden z rohov ŠPZ

Z experimentov vyplynulo, že deskriptor BRISK nie je najvhodnejším kandidátom na riešenie úlohy lokalizácie ŠPZ v obraze. Prvou z príčin je fakt, že metóda k vyhľadávaniu bodov v rámci rôznych mierok využíva detektor FAST, ktorý už bol analyzovaný a jeho hlavný nedostatok bol pomenovaný v kapitole 3.6.2.

Príkaz *detectBRISKFeatures* poskytuje ako jeden z parametrov parameter *NumOctaves*, ktorý definuje cez koľko oktáv budú vyhľadávané dané významné body. Experimenty ukázali, že použitie daného parametru na daných galériách nemalo význam a najlepšie výsledky boli nájdené pri nastavení daného parametru na hodnotu 0, tj. vyhľadávanie oblastí iba v mierke obrázku. S týmto nastavením boli získané všetky hodnoty v tabuľke úspešnosti na jednotlivých galériách okrem dvoch reálnych galérii kde bol tento parameter nastavený na hodnotu 1. Problémom u tohto parametru je, že sa ŠPZ v galériách nachádzajú v rôznych veľkostiach a pri ponechaní ostatných parametrov bez zmeny, môže dochádzať k detekcii bodov s rovnakou kvalitou a kontrastom cez dve rôzne mierky a veľmi často nastávala situácia, že niektorý z rohov ŠPZ bol detegovaný v inej mierke a nevyhovel podmienke na mierku uvedenej vyššie. Vynechaním využitia tejto pridanej hodnoty by sme sa vrátili k detektoru FAST a nemalo by význam BRISK analyzovať.

BRISK	Vyhodnotenie		Parameter:
	FP	TP	MinQuality
Real1	8,3%	34,1%	0,0018
Real2	6,7%	38,1%	0,0018
MPOV_1	5,6%	23,9%	0,0400
MPOV_2	10,8%	13,5%	0,0400
MPOV_3	14,1%	48,7%	0,0400
db_destivo	9,1%	7,6%	0,0400
db_park_slunecno	39,4%	12,1%	0,0400
db_park_zatazeno	10,7%	33,0%	0,0400
db_slunecno	33,3%	7,4%	0,0400

Tabuľka 5: Úspešnosť a parametre algoritmu využívajúceho BRISK deskriptor

Opäť je ale možné vidieť isté zlepšenie na galériách *real1* a *real2*, ktoré sú najrovnorodejšie. To dáva deskriptoru BRISK čiastočné možnosti na uplatnenie, avšak s porovnaním inými rohových detektorov a časovej náročnosti výpočtu to nebude najvýkonnejšia a najspoľahlivejšia možnosť.



Obrázok 43: Ukážka vysokej odozvy deskriptora BRISK na trávnatú plochu. Detektor pracuje na princípe detektora FAST a má vysokú odozvu v značne štruktúrovaných oblastiach kde sa nachádza množstvo ostrých rohov.

Ďalšími obmedzeniami ktoré stoja za zmienku sú pomerne vysoká odozva deskriptora v trávnatých oblastiach ako ukazuje vyššie zobrazený obrázok. V danom prípade je detegovaných viac ako 800 bodov a stále nie sú detegované rohové body ŠPZ (prípade obrázku vyššie). Znížiť hodnoty parametrov tak aby dané body vyhoveli, niekoľko násobné zväčší počet bodov nakoľko pokiaľ sa kvalitou dostaneme na body ŠPZ bude stále detegované množstvo bodov v trávinatej ploche. Táto situácia je charakteristická aj pre samotný FAST detektor. Pre našu experimentálnu úlohu lokalizácie ŠPZ by bolo efektívne využiť nejakú nadstavbu nad daným detektorom, ktorá znižuje nerovnomerné rozloženie detegovaných oblastí v obraze.



Obrázok 44: Príklad nepomeru počtu získaných BRISK bodov pre dve snímky z rovnakej galérie získaných za rovnakých podmienok. Vo vrchom rade snímky bez predspracovania, v rade spodnom snímky filtrované Gaussovským filtrom (veľkosť 5x5 a smerodajná odchýlka $\sigma = 1$).

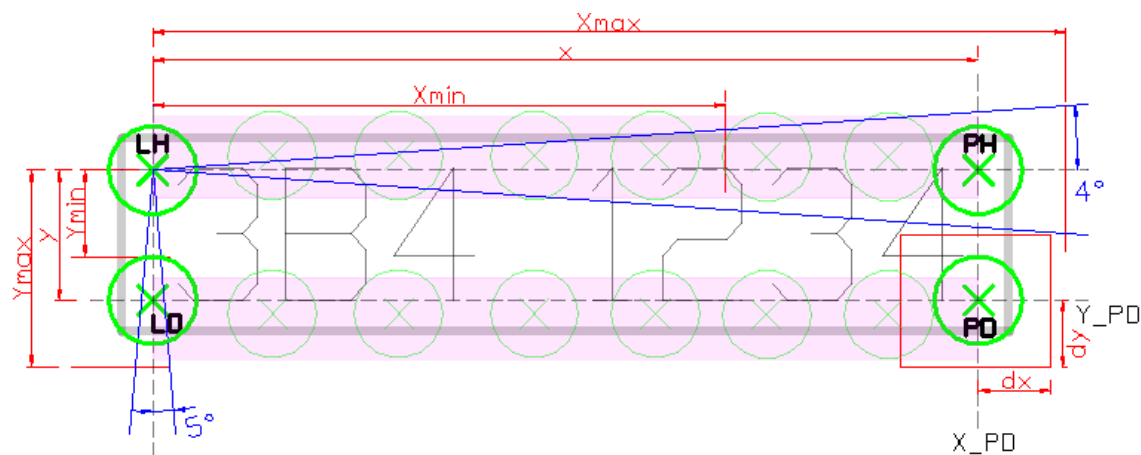
Pomerne zaujímavou je situácia, ktorá nastávala v rámci galérie real1 a real2, kde sa nachádzajú snímky získane v približne rovnakom časovom horizonte a za rovnakých podmienok. Na niektorých z týchto snímok je pri rovnakej hodnote parametrov získaných približne 370 významných bodov, na iných sa toto číslo prekračuje 1400 (prípady vyššie uvedených snímok). Väčšinou sa jedná o snímky so svetlým a naopak tmavým vozidlom. Tieto rozdiely boli bežné u snímok s nehomogénnymi okolím vozidla, ale dané galérie majú na pozadí rovnakú vozovku. Táto vlastnosť je na rôznorodých galériách ešte umocnená a spôsobuje neúmerne predĺženie chodu algoritmu. Základným predspracovaním je možné dosiahnuť výrazné zníženie odozvy detektora v oblasti vozovky v porovnaní s oblasťou vozidla. Záverom je bohužiaľ pre našu testovaciu úlohu potrebné zhodnotiť, že algoritmus nie je zrovna citlivý na rohové body ŠPZ.

3.8 Lokalizácia ŠPZ použitím deskriptoru SURF

3.8.1 Popis algoritmu

Posledným z vyvinutých a aktívne testovaných algoritmov je algoritmus využívajúci deskriptor a deskriptor SURF. Teoreticky je tento pomerne mladý deskriptor popísaný v kapitole 2.3. Daný deskriptor s vlastným detektorom je opäť priamo implementovaný

v MATLAB-e a je možné ho zavolať príkazom *detectSURFFeatures*. Jeho odozva na snímky bola podrobená analýze na základe výsledkov ktorej bol následne implementovaný algoritmus lokalizácie ŠPZ pomocou významných bodov MSER. U danej metódy boli implementované dve riešenia, ale nakoľko prvé riešenie neposkytovalo dostatočne dobré možnosti presnej lokalizácie ŠPZ, bol vytvorený nový algoritmus. Obidva algoritmy vychádzajú z nasledujúceho obrázku.

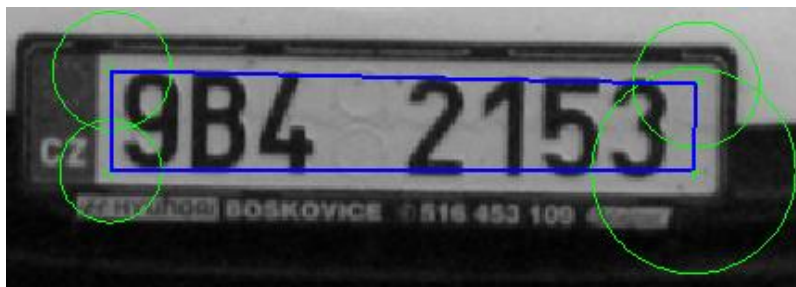


Obrázok 45: Názorné zobrazenie geometrie algoritmu lokalizácie ŠPZ v obraze pomocou SURF oblastí

Z analýzy odozvy detektora a deskriptora SURF bolo možné pozorovať oblasti, kde má algoritmus silnú odozvu. V oblasti ŠPZ metóda silne reaguje na oblasti zmeny kontrastu, napríklad na hranách ŠPZ kde sa v rámci vyšetrovanej oblasti nachádza svetlá plocha po stranách ohraničená tmavou oblasťou (na jednej strane ohraničená tmavým rámčekom ŠPZ, na druhej strane ohraničená tmavým znakom). Tieto spomínané body sú zobrazené vo fialovom rámčeku v Obrázok 45. Prvotný prístup v riešení lokalizácie bol založený práve na vyhľadávaní dvoch horizontálnych pruhov obsahujúcich tieto body v danej vzdialenosti od seba. Ako sa ale následne ukázalo, dané riešenie neposkytovalo ďalší potenciál nakoľko v rámci každej oblasti ŠPZ bolo detegované rozličné množstvo týchto bodov. Ďalším problémom bolo, že aj v prípade kedy boli tieto dve horizontálne oblasti nájdené, bolo veľmi náročné určiť ich dĺžku v horizontálnom smere nakoľko každá z nich obsahovala rozličný počet bodov. Daný prístup poskytoval možnosti vertikálnej lokalizácie ŠPZ s miernym upravením horizontálnej veľkosti, ale presná lokalizácia ŠPZ nebola možná, čo znižovalo jeho praktické využitie smerom ďalšieho spracovania úlohy rozpoznávania ŠPZ.

Po ďalšom zvážení možnosti detekcie ŠPZ použitím SURF deskriptoru sa ako možnosť ukázala opäť akási modifikácia algoritmu 4 rohových bodov. V tomto prípade sa ale v skutočnosti nejedná priamo o rohové body, ale o body nachádzajúce sa v rámci svetlej oblasti pod uhlom 45° smerom k najbližšiemu znaku ŠPZ. Bola vytvorená zmiešaná galéria na základe ktorej boli analyzované tieto 4 body a vytvorený algoritmus pracujúci na základe Obrázok 45 vyhľadávajúci kombináciu 4 bodov (LH, PH, LD a PD) v požadovanej dispozícii. Na základe analýzy viacerých ŠPZ a odozvy SURF bolo

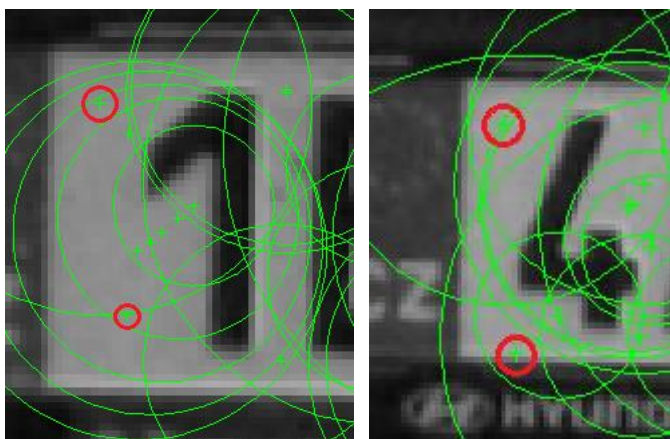
potrebné algoritmus nastaviť mierne benevolentnejšie ako priamo pri rohových bodoch čo sa týka pomeru vertikálnej a horizontálnej vzdialenosti, ktorá sa pohybovala od hodnoty 5 po približne hodnotu 6,5.



Obrázok 46: Ukážka požadovaného výsledku pri lokalizácii ŠPZ pomocou deskriptora SUFR (zelené oblasti)

Spomalenie algoritmu vyhľadávania týchto 4 bodov spôsobuje to, že nie je možné body kategorizovať nakoľko sa nenachádzajú na žiadnom kontrastnom rozhraní ale uprostred homogénnej plochy a neposkytujú žiadny údaj na základe ktorého by to bolo možné. Táto nevýhoda je mierne kompenzovaná faktom, že je možná eliminácia kombinácií, ktoré neležia v oblasti s približne rovnakou hodnotou jas. To značne redukuje falošných geometrických kandidátov. Samozrejme platia požiadavky na uhly a vzdialenosti medzi horizontálnymi a vertikálnymi kombináciami bodov.

3.8.2 Vyhodnotenie



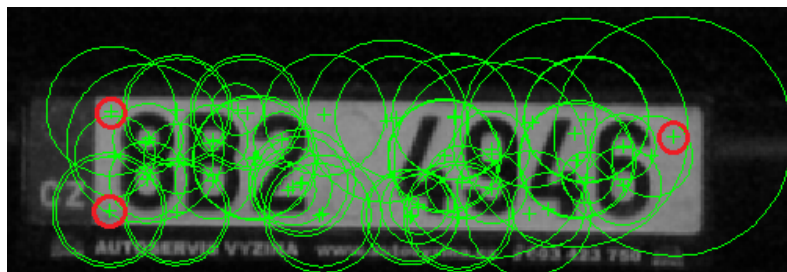
Obrázok 47: Príklady nepravidelného rozloženia SURF významných bodov v okolí znakov 1 a 4

Prvým z problémov, ktoré boli objavené nerovnomerné rozloženie bodov v okolí znakov 1,4 a 7 v ľavej časti ŠPZ ako zobrazuje obrázok vyššie. Parametre implementovaného algoritmu boli nastavené tak, že dané body geometrickým testom nemali možnosť prejsť. Riešenie zvýšením voľnosti algoritmu by ale viedlo k zvýšeniu počtu falošných kandidátov a predĺženiu chodu celého algoritmu nakoľko by bolo v závere testovaných mnoho kandidátov.

SURF	Vyhodnotenie:		Parametre:		
	FP	TP	MetricThreshold	NumOctaves	NumScaleLevels
Real1	2,8%	67,4%	2000	2	6
Real2	6,6%	50,0%	2000	2	6
MPOV_1	19,7%	12,7%	2200	3	3
MPOV_2	16,2%	2,7%	2200	3	3
MPOV_3	7,7%	15,4%	1800	3	3
db_destivo	14,9%	0,0%	2200	3	3
db_park_slunecno	13,3%	1,7%	1800	3	3
db_park_zatazeno	11,6%	0,0%	2200	3	3
db_slunecno	29,6%	0,0%	2200	3	3

Tabuľka 6: Úspešnosť a parametre algoritmu využívajúceho SURF deskriptor

Z tabuľky zobrazujúcej štatistické výsledky implementovaného algoritmu je možné vidieť, že algoritmus nedosahuje na danej experimentálnej úlohe ani zďaleka optimálne výsledky. Jednak je možné vidieť nízku úroveň úspešného lokalizovania ŠPZ (nulovú na viacerých galériách) a na druhú stranu pomerne vysoké hodnoty „false positive“ výsledkov, teda prehlásenia falošného kandidáta za ŠPZ. Snímky, na ktorých ŠPZ nebola lokalizovaná boli analyzované a ako najväčší dôvod zlyhania sa ukázala absencia minimálne jedného zo 4 rohových bodov. Je teda na mieste konštatovať, že robustnosť týchto 4 významných bodov je nižšia ako by si úloha vyžadovala. V snímku na nachádzajú robustnejšie významné body, konkrétne sa jedna o body na rozhraní vozidlo-vozovka.



Obrázok 48: Príklad snímku ŠPZ s chýbajúcim SURF bodom v pravom dolnom rohu. Chýbajúci bod je najčastejšou príčinou zlyhania algoritmu

Tento fakt navyše potvrdzuje porovnanie výsledkov medzi galériami real1 a real2 a ostatnými. Je možné pozorovať jasné zníženie úspešnosti algoritmu na týchto rôznorodých galériách. Body detegované na okrajoch budov, v silne štruktúrovaných oblastiach ako napríklad koruny stromov sú robustnejšie a pravidelne detegované. Samotný SURF v kombinácii s vytvoreným algoritmom absolútne nie je vhodný na riešenie danej úlohy.

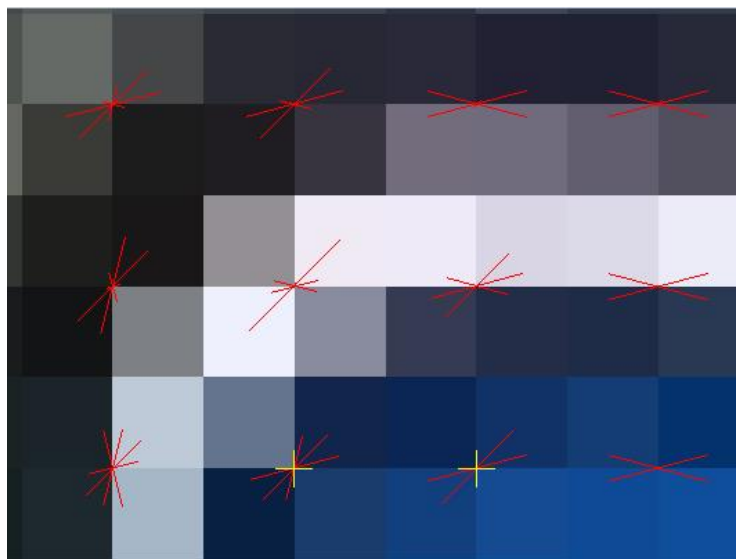
3.9 Použitie ostatných algoritmov k lokalizácii

3.9.1 HoG

Histogram of oriented gradients je jeden z deskriptorov, ktorý bol popísaný v druhej kapitole venovanej deskriptorom významných oblastí. Algoritmus počíta zmenu intenzity jasu v definovaných smeroch a následne ju kategorizuje do istých intervalov. Algoritmus zo svojho princípu umožňuje použitie na detegovanie pravidelných geometrických tvarov. Nakoľko bola zvolená úloha lokalizácie pravidelnej ŠPZ algoritmus ponúkal istý potenciál na riešenie tejto úlohy a preto bol podrobený testovaniu na možnosti jeho využitia. Tieto pôvodné dohady boli bohužiaľ vyvrátené, nakoľko samotný HoG nepopisuje len body významné, ale popisuje všetky body obraze. To v praxi znamená, že by bola potrebná veľmi pracná filtrácia týchto popísaných oblastí, aby sme získali oblasti, ktoré sú významné a mohli by odpovedať istým pravidelnostiam v rámci ŠPZ ako sú rohy poprípade hrany.

Metóda bola v MATLAB-e testovaná tak, že bola vytvorená sieť histogramov pre polia o veľkosti 5x5 pixelov. Ako sa ukázalo, aby sme získali aspoň čiastočne distingvované výsledky pre rohy ŠPZ, bolo potrebné zvoliť menšie pole pre výpočet histogramov a to konkrétne 2x2 pixely. Rozdelenie do tzv. binov (košov) bolo nastavené tak, aby boli pokryté uhly 45° v kladnom a zápornom smere od vodorovnej osy, nakoľko pod týmito uhlami je najvyššia hodnota histogramu v rohových bodoch ŠPZ, samozrejme závislé na rozlíšení obrázku a vzdialenosti ŠPZ od objektívu. Pre testovacie účely bol uvažovaný štandardný obrázok s rozlíšením 800x600 pixelov a vozidlo vo vzdialenosti približne 3 metrov od objektívu.

Testovací algoritmus rozdeľuje body do dvoch kategórií. Body, ktoré majú majoritný smer gradientu pod uhlom 45° v smere hodinových ručičiek od vodorovnej osy, teda body ktoré sú potenciálnymi kandidátmi na ľavý horný alebo pravý dolný roh ŠPZ. Analogicky ale s uhlom proti smeru hodinových ručičiek sú vyhľadávaní kandidáti na ľavý dolný alebo pravý horný roh. Hodnota majoritného gradientu je pre daného kandidáta porovnávaná s hodnotou gradientu pod uhlom 45° v opačnom smere od horizontálnej osy ako je smer majoritného gradientu, nazvime pre účely popisu tento smer smerom inverzným. Na základe experimentov sa ukázalo, že veľkosť gradientu v majoritnom smere by mala byť 10-krát väčšia ako veľkosť v smere inverznom. Následnej je porovnaný gradient v majoritnom smere s gradientami v smere $\pm 20^\circ$ od inverzného smeru. Pre tieto by malo platiť, že majoritný gradient je aspoň 5-krát väčší ako tieto dva gradienty. Tieto násobky, získané experimentálne zaručili, že v oblasti rohov ŠPZ bol označený aspoň jeden bod ako kandidát na daný roh. Situáciu pre ľavý horný roh ŠPZ ukazuje nasledujúci obrázok. Žltým krížom sú označené body, ktoré vyhovel podmienkam.



Obrázok 49: Odozva HoG na ľavý horný roh ŠPZ a kandidáti na rohový bod

Pomocou vyššie popísaného algoritmu je možné nájsť 4 body, ktoré zodpovedajú rohom ŠPZ. Algoritmus ale nebol ďalej testovaný nakoľko bolo v experimentálnej snímke označených 8280 bodov ako kandidátov na LH/PD roh a 8060 bodov ako kandidáti na LD/PH roh. Hľadať kombináciu 4 bodov, vyhovujúcej geometrickej kombinácie medzi takým počtom je z výpočtového hľadiska takmer nezmyslom a algoritmus by vyžadoval vysokú časovú náročnosť. Algoritmus značne zlyháva v prípade, keď nie je ŠPZ v obraze umiestnená presne horizontálne. Z tohto dôvodu je potrebné voliť tak malé pole pre výpočet histogramov. Ďalším aspektom, pre ktorý je daný algoritmus nevhodný pre danú úlohu je aj fakt, že pri testovacích podmienkach popísaných vyššie vráti príkaz `extractHOGFeatures` vektor o veľkosti 1x720000 prvkov a jeho následná úprava na tvar vhodný k spracovaniu údajov a vyhľadanie potenciálnych kandidátov trvá na testovacom snímku približne 28 sekúnd. Tento fakt sa samozrejme odvíja od toho, že metóda ako taká pracuje plošne na celom snímku. HoG môže nachádzať uplatnenie s kombináciou detektora rohových významných bodov, pre ktoré by následne boli spočítané histogramy a body by mohli byť kategorizované do skupín odpovedajúcich daným rohom ŠPZ čo by následnej zrýchlilo vyhľadávanie vyhovujúcej kombinácie týchto bodov.

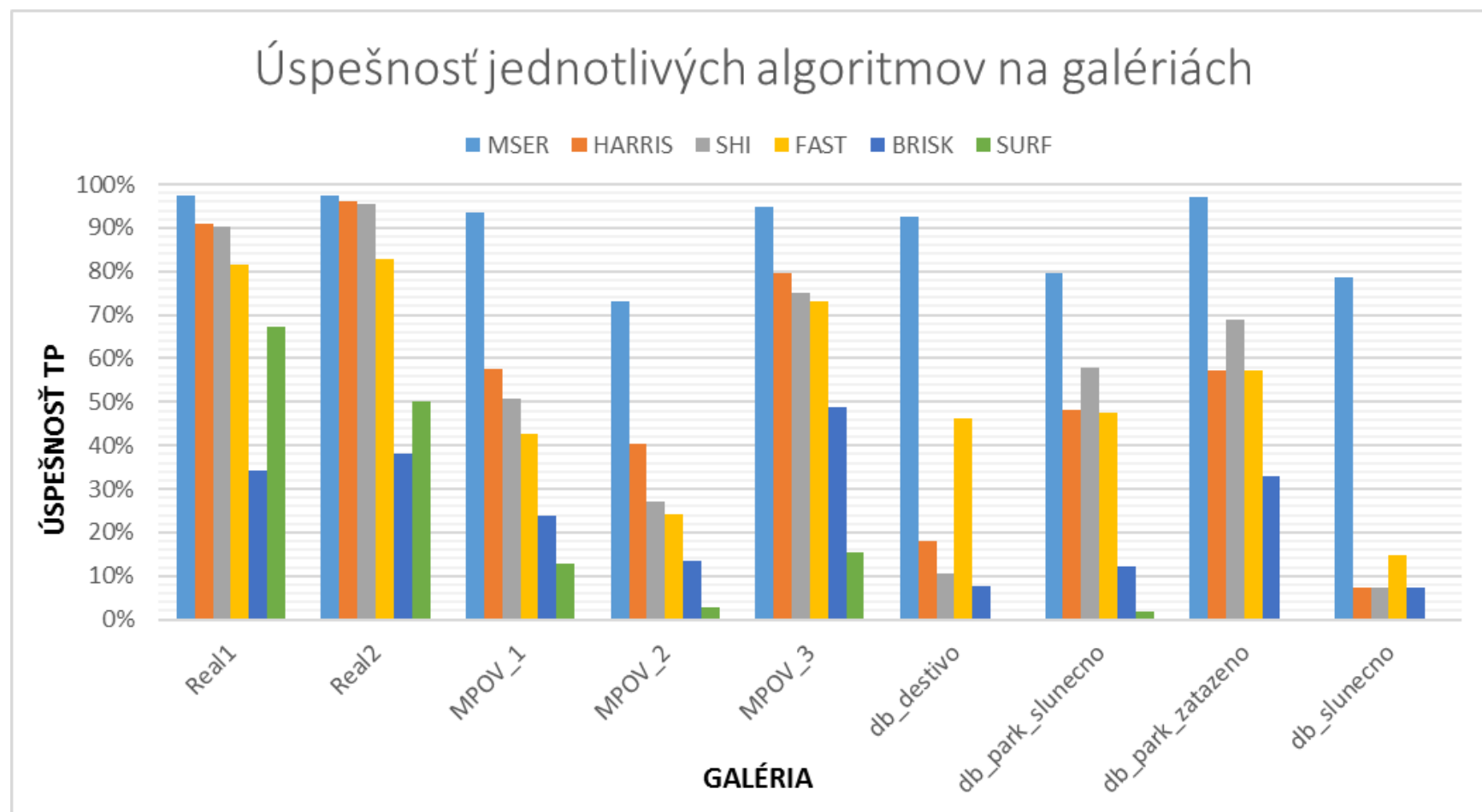
3.9.2 SIFT

Druhým s algoritmov popísaných v teoretickej časti tejto diplomovej práce, ktorý nebol ďalej testovaný je algoritmus SIFT. Dôvodom pre tento krok je fakt, že algoritmus je čo do použiteľnej časti pre riešenie úlohy lokalizácie ŠPZ totožný s algoritmom SURF. Výhodou algoritmu SURF je zníženie výpočtovej náročnosti a zvýšenie robustnosti a preto sa dá očakávať, že výsledky získané v kapitole 3.7 budú približne korešpondovať s výsledkami, ktoré by boli získané použitím deskriptora SIFT.

3.10 Sumarizácia dosiahnutých výsledkov

Výsledok →	TP						FP					
↓Galéria / Metóda→	MSER	HARRIS	SHI	FAST	BRISK	SURF	MSER	HARRIS	SHI	FAST	BRISK	SURF
Real1	97,3%	91,1%	90,4%	81,7%	34,1%	67,4%	0,0%	1,8%	4,1%	7,3%	8,3%	2,8%
Real2	97,4%	96,1%	95,4%	82,9%	38,1%	50,0%	0,0%	1,3%	0,7%	1,9%	6,7%	6,6%
MPOV_1	93,4%	57,7%	50,7%	42,6%	23,9%	12,7%	0,0%	8,4%	11,3%	16,9%	5,6%	19,7%
MPOV_2	73,0%	40,5%	27,1%	24,3%	13,5%	2,7%	0,0%	13,5%	21,6%	13,5%	10,8%	16,2%
MPOV_3	94,8%	79,5%	74,9%	73,1%	48,7%	15,4%	0,0%	5,1%	7,7%	6,4%	14,1%	7,7%
db_destivo	92,4%	17,9%	10,6%	46,3%	7,6%	0,0%	0,0%	6,0%	12,1%	15,0%	9,1%	14,9%
db_park_slunecno	79,5%	48,1%	58,0%	47,4%	12,1%	1,7%	0,0%	13,3%	10,7%	5,8%	39,4%	13,3%
db_park_zatazeno	97,2%	57,1%	68,8%	57,1%	33,0%	0,0%	0,9%	8,0%	12,5%	10,7%	10,7%	11,6%
db_slunecno	78,6%	7,4%	7,4%	14,8%	7,4%	0,0%	0,0%	11,1%	29,6%	22,2%	33,3%	29,6%

Tabuľka 7: Tabuľka zobrazujúca získané výsledky pre dané galérie použitím rôznych detektorov a deskriptorov. Výsledky sú rozdelené na „true positive“ teda úspešné lokalizovanie ŠPZ a „false positive“ teda falošné lokalizovanie ŠPZ. Výsledky označené zeleným písmom indikujú najlepší výsledok na danej galérii, naopak výsledky označené červeným písmom indikujú najhoršie výsledky. Je možné pozorovať najlepší výsledok pre detektor MSER naopak najhoršie pre SURF a BRISK.



Obrázok 50: Graf zobrazujúci porovnanie úspešnosti TP lokalizácie ŠPZ v obraze na jednotlivých galériách

ZÁVER

Diplomová práca s názvom Detektory a deskriptory oblastí v obraze sa venuje oblasti počítačového videnia, ktorá je využiteľná pri analýze obrazu, konkrétne v časti segmentácie kam je možné zaradiť detekciu významných bodov a následne v časti popisu kam možno zaradiť deskriptory oblastí v obraze. Toto zaradenie vychádza z Obrázok 1. Táto diplomová práca bola spracovaná ako pokračovanie rovnomenného semestrálneho projektu a je koncepčne delená do 3 častí.

Prvá časť práce sa venuje detektorom oblastí v obraze. Cieľom bolo naštudovať problematiku detektorov a zostaviť prehľad s príkladmi. Danej problematike sa venuje 1. kapitola, kde bola uvedená charakteristika významnej oblasti pre detekciu a následne vytvorený prehľad najznámejších detektorov významných oblastí. V tejto kapitole sú spracované najznámejšie a najvyužívanejšie detektory, menovite: Moravcov detektor, Harrisov detektor, FAST a MSER. Jedným z bodov zadania bolo vykonať s danými detektormi experiment a vyhodnotiť ho. Tento úvodný experiment pozostával z využitia a otestovania detekcie daných metód na snímku z dopravy so zameraním na oblasť ŠPZ. V druhej časti práce som sa následne venoval matematickému popisu týchto významných oblastí a zoznámil som sa s najznámejšími a najvýkonnejšími deskriptormi. Nakoľko sa jedná o výpočtovo náročnú oblasť je možnú ju označiť za značné živú a hlavne u posledne spomínaných algoritmov je možné vidieť, že sa táto oblasť stále aktívne vyvíja a je zaujímavou pre ďalší vývoj. Teoreticky boli spracované a rovnako ako v predchádzajúcej kapitole aj experimentom podrobené nasledujúce deskriptory: HoG, SIFT, SURF, BRIEF, BRISK, ORB a FREAK.

U väčšiny algoritmov bol uvedený ilustratívny výstup získaný pomocou MATLAB-u. Tieto obrázky síce samostatne nemajú veľkú výpovednú hodnotu, poskytujú však čitateľovi aspoň mierne priblíženie čo je výstupom z týchto algoritmov a je možné pozorovať, ktoré problémy môžu vzniknúť. Rovnako tak poskytujú ilustráciu ich možného využitia v ďalšej časti tejto práce. Prvé dve kapitoly práce sú zosumarizovaním aktuálnej situácie na poli detektorov a deskriptorov oblastí v obraze a čitateľovi poskytujú ucelený prehľad najznámejších prístupov.

Jadrom práce je ale kapitola tretia, nazývaná experimentálna časť. Na základe dohody s vedúcim diplomovej práce bolo potrebné vybrať experimentálnu úlohu, kde by bolo možné porovnať vybrané z prezentovaných metód. Zvolenou úlohou bola lokalizácia ŠPZ v obraze. Vybrané detektory a deskriptory pre ktoré boli implementované algoritmy pre lokalizáciu ŠPZ a ktoré boli následne testované sú: Harrisov detektor, Shi-Tomasi detektor, FAST, BRISK, MSER a SURF. Algoritmy, ktoré boli ďalej podrobené analýze avšak bez následného hĺbkového testovania boli HoG a SIFT. Ako ukazujú tabuľky s úspešnosťou jednotlivých algoritmov je jasným víťazom pri riešení úlohy lokalizácie ŠPZ detektor MSER. Tento detektor vykazuje veľmi vysokú úspešnosť na reálnych galériách (približne 95%). Jeho úspešnosť na ostatných galériách

síce klesá, avšak nie tak rapídne ako je to pozorované u iných testovaných algoritmov. Daný algoritmus je navyše pomerne rýchly vďaka možnosti eliminácie falošných MSER oblastí kedy vo výsledku vyhľadávame 7 elíps z približne 60. Jeho slabou stránkou je fakt, že daný algoritmus je nastavený na presné rozloženie elíps, teda na jeden typ ŠPZ a nie je ním možné detegovať samotný obdĺžnik oblasti ŠPZ. Ďalším obmedzením je jeho znížená odolnosť na značne presvetlených galériách alebo v prípade tieňa zasahujúceho do oblasti ŠPZ. Všeobecne sú ale MSER oblasti vznikajúce na znakoch ŠPZ značne robustné a stabilné.

Druhým typom vyhľadávania ŠPZ v obraze bolo vyhľadávanie použitím 4 rohových bodov rámčeka ŠPZ. Principiálne boli týmto spôsobom testované Harrisov detektor, Shi-Tomasi detektor, FAST detektor a BRISK deskriptor. Najlepšie si v danej veci poradili najstaršie zo spomínaných metód a to detektor Harrisov a Shi-Tomasi, ktoré na galériách *real1* a *real2* dosiahli 90-96% úspešnosť. Výsledky a príčiny zlyhaní sú u všetkých testovaných metód uvedené v príslušných kapitolách. Dôležité je ale upozorniť na fakt, že úspešnosť daných algoritmov značne klesala na ostatných galériách, ktoré obsahovali štruktúrované pozadie napríklad stromy a budovy, vytvárajúce množstvo rohových bodov. Nižšia úspešnosť bola zaznamenaná u detektora FAST a deskriptora BRISK, ktorý už takmer vôbec neposkytuje potenciál na úspešné riešenie danej úlohy (približne 35% na reálnych galériách).

Posledným implementovaným algoritmom bol algoritmus využívajúci deskriptor SURF k lokalizácii ŠPZ v obraze. Jeho úspešnosť sa pohybuje na reálnych galériách v okolí hodnoty 50 a 67% a na ostatných galériách opäť značne klesá, na niektorých galériách dosahuje 0% úspešnosť. To je spôsobené tým, že významne body použité pre lokalizáciu ŠPZ majú nižšiu stabilitu ako významné body na rozhraní budov a okolia, stromov a oblohy a podobne.

Zadanie diplomovej práce bolo splnené a dané metódy boli porovnané na úlohe lokalizácie ŠPZ. Slabší článok práce je algoritmus testujúci kandidátov na skutočné oblasti ŠPZ. Tento algoritmus produkoval množstvo falošných kandidátov a odhadom 80% z týchto falošných kandidátov poskytovalo potenciál (dostupnosť významných bodov a oblastí) pre správnu lokalizáciu ŠPZ. Tento fakt ale neovplyvňuje samotné výsledky práce, nakoľko bol rovnaký vyhodnocovací algoritmus použitý pre všetky testované metódy. Samotná úloha lokalizácie ŠPZ, ktorej vyriešenie nebolo cieľom tejto práce stále poskytuje priestor na zlepšenie. Využitím kvalitného predspracovania obrazu v kombinácii s detektorom MSER a kvalitnejším vyhodnocovaním kandidátov by bolo možné dosiahnuť lepšie výsledky. V kombinácii so záložným vyhľadávaním ŠPZ napríklad pomocou hrán by táto kombinácia mohla plniť svoju úlohu v reťazci celého rozpoznávania ŠPZ.

Zoznam literatúry

- [1] GONZALES, R.C., WOODS, R.R.: Digital image processing (3rd edition). Prentice-Hall, Inc., 2008. 954 pages. ISBN 978-0131687288.
- [2] YOUNG,I.T., GERBRANDS,J.J., VLIET,L.J.: Fundamentals of Image Processing. TU Delft, 1998. 113 pages. ISBN 9075691017.
- [3] SZELISKI, R.: Computer Vision: Algorithms and Applications. Springer,2010. ISBN 978-1848829343.
- [4] ŠTENČEL, J. Měření aktuální rychlosti vozidla. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 60 s. Vedoucí bakalářské práce Ing. Karel Horák, Ph.D.
- [5] VINCZE, M., WOLF, D., ZILLICH, M.: Machine Vision and Cognitive Robotics. Prezentace prezentovaná na: Automatisierungs- und Regelungstechnik Institut Technische Universität Wien.
- [6] BÍLEK, P. Významné body v obraze: detekce, korespondence a lokalizace ve 3D. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2007. 47 s. Vedoucí bakalářské práce Ing. Daniel Sýkora, Ph.D. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/bilekp3_2007bach.pdf
- [7] TRÁVNÍČEK, V. Porovnávání významných bodů pro detekci objektů v obraze. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 42s. Vedoucí bakalářské práce Ing. Vratislav Harabiš.
- [8] PARKS, D., GRAVEL, J.P., Corner detection. Prednáška.
- [9] HARRIS C., STEPHENS M., A Combined Corner and Edge Detection, 1988, Proceedings of The Fourth Alvey Vision Conference. Dostupné z: <http://www.ic.unicamp.br/~rocha/teaching/2013s1/mc851/aulas/additional-material-harris-detector.pdf>
- [10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, 2004. Dostupné z: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

- [11] H. Bay, T. Tuytelaars, and L. van Gool. SURF: Speeded up robust features. In Proceedings of European Conference on Computer Vision, pages I: 404–417, 2006. Dostupné z: <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
- [12] DRUMMOND. T., ROSTEN. E., Machine learning for high-speed corner detection. In Proceedings of the 9th European conference on Computer Vision - Volume Part I (ECCV'06), 2006. s.430-443. Dostupné z: http://www.edwardrosten.com/work/rosten_2006_machine.pdf
- [13] DRUMMOND. T., ROSTEN. E., Fusing points and lines for high performance tracking. In 10th IEEE International Conference on Computer Vision. Volume 2., Peking, Čína. 2005. s.1508-1515. Dostupné z: http://www.edwardrosten.com/work/rosten_2005_tracking.pdf
- [14] CALONDER, M. et al., Brief: Binary robust independent elementary features. Pattern Analysis and Machine Intelligence, IEEE Transactions. Volume 34. Issue 7. 2011. s. 1281-1298. Dostupné z: <https://www.robots.ox.ac.uk/~vgg/rg/papers/CalonderLSF10.pdf>
- [15] MATAS. J., CHUM. O., URBAN. M., PAJDLA. T., Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. Proceedings of the British Machine Vision Conference. 2002. s. 36.1-36.10. Dostupné z: https://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/matas_bmvc2002.pdf
- [16] LEUTENEGGER. S., CHLI. M., SIEGWART. R., BRISK: Binary Robust Invariant Scalable Keypoints. International Conference on Computer Vision (ICCV), 2011. s.2548-2555. Dostupné z: <https://www.robots.ox.ac.uk/~vgg/rg/papers/brisk.pdf>
- [17] RUBLEE. E., et al. ORB: An efficient alternative to SIFT or SURF. Computer Vision (ICCV) International Conference. 2011. s.2564-2571. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6126544>
- [18] ALAHI. A., et al. Freak: Fast retina keypoint. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference. s. 510–517. Dostupné z: <http://infoscience.epfl.ch/record/175537/files/2069.pdf>
- [19] VEVALDI. A., SIFT for MATLAB. 2014. Dostupné z: <http://vision.ucla.edu/~vedaldi/code/sift.html>
- [20] DALAL. N., TRIGGS. B., Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer vision nad Pattern Recognition, 2005. Dostupné z: <https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>

- [21] KALOVÁ. I., Předzpracování obrazu. Prezentácia prezentovaná na: Ústav automatizace a měřící techniky; FEKT VUT Brno.
- [22] MIKOLAJCZYK. K., SCHMID. C., Scale and Affine Invariant Interest Points Detctors, International Journal of Computer Vision. 2004. Dostupné z: <http://www.cse.unr.edu/~bebis/CS773C/ObjectRecognition/Papers/Mikolajczyk04a.pdf>
- [23] Česká republika. Zákon č. 243/2001 Sb. - o registraci vozidel. In: 2001.

Zoznam skratiek

3D – 3-rozmerný

2D – 2-rozmerný

BRIEF – Binary robust independent elementary features

BRISK – Binary robust invariant scalable keypoints

FAST – Features from accelerated segment test

FP – False positive

FREAK – Fast retina keypoints

HoG – Histogram of oriented gradients

LD – ľavý dolný

LH – ľavý horný

MSER – Maximally stable extremal regions

PD – pravý dolný

PH – pravý horný

ORB – Oriented FAST and rotated BRIEF

SIFT – Scale-invariant feature transform

SURF – Speeded up robust features

TP – True positive

ZOZNAM PRÍLOH

Príloha č.1 – DVD-ROM

- Elektronická verzia diplomovej práce
- Testovacie galérie
- Zdrojové kódy vytvorených testovacích algoritmov