

Analýza vhodných metod zpracování obrazu pro řízení pohybu kolového robota

Patrik Láník

Bakalářská práce
2021



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Patrik Láník**
Osobní číslo: **A20935**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Inteligentní systémy s roboty**
Forma studia: **Prezenční**
Téma práce: **Analýza vhodných metod zpracování obrazu pro řízení pohybu kolového robota**
Téma práce anglicky: **An Analysis of Suitable Image Processing Methods for Wheeled Robot Motion Control**

Zásady pro vypracování

1. Provedte analýzu literárních pramenů vztahujících se k problematice zpracování obrazu v reálném čase.
2. Vyberte vhodné metody pro řešení základního rozpoznání 3D scény pro strategii řízení rovinného pohybu kolového robota při známé vzdálenosti překážek (tj. bez měření vzdálenosti).
3. Vhodné metody otestujte s použitím nástroje MATLAB a vyberte ty, které splňují odpovídající požadavky.
4. Vytvořte na základě poznatků z předchozího vlastní program s využitím dat z reálně zvolené kamery v jazyku C/C++.
5. Otestujte obrazová data a odlaďte implementované algoritmy zpracování obrazu tak, aby odpovídaly požadavkům na řízení pohybu v reálném čase.
6. Provedte podrobné vyhodnocení výsledků při různých světelných podmínkách.

Rozsah bakalářské práce: –
Rozsah příloh: –
Forma zpracování bakalářské práce: **Tištěná/elektronická**

Seznam doporučené literatury:

1. SANKOWSKI, Dominik a Jacek NOWAKOWSKI. Computer vision in robotics and industrial applications. Singapore: World Scientific, [2014], 563 s. Series in computer vision. ISBN 9789814583718.
2. CORKE, Peter I. Robotics, vision and control: fundamental algorithms in Matlab. Berlin: Springer, 2011, 570 s. Springer tracts in advanced robotics. ISBN 9783642201431.
3. ŠONKA, Milan, Václav HLAVAC a Roger BOYLE. Image processing, analysis, and machine vision. Second edition. Pacific Grove: PWS Publishing, (1999), 770 s. ISBN 053495393X.
4. BURGER, Wilhelm a Mark James BURGE. Principles of digital image processing: advanced methods. London: Springer, 2013. Undergraduate topics in computer science (Springer). ISBN 978-1848829183.
5. ZHI ENG, L. Qt5 C++ GUI Programming Cookbook, 2nd ed.; Packt Publishing: Birmingham, England, 2019; pp. 428.
6. MALLICK, Satya. Object Tracking using OpenCV (C++/Python) [online]. 13 February 2017, s. 12 [cit. 2020-11-24]. Dostupné z: <https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/>

Vedoucí bakalářské práce: **Ing. Jiří Zátopek**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **15. ledna 2021**
Termín odevzdání bakalářské práce: **17. května 2021**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2021

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 12.5.

Patrik Láník, v. r.

ABSTRAKT

Cílem bakalářské práce je zanalyzovat a naprogramovat v jazyku C++ vhodné metody zpracování obrazu pro řízení pohybu kolového robotu. V první části se práce zabývá analýzou vhodných metod zpracování obrazu. Metody jsou otestovány s využitím nástroje MATLAB a jsou vybrány pouze ty, které splňují odpovídající požadavky. Druhá část práce se zabývá samotnou implementací vybraných metod s využitím knihovny OpenCV, aplikované přímo na snímky z reálně zvolené kamery. Poté jsou metody odladěny pro potenciální řízení pohybu v reálném čase. Poslední část práce se zabývá vyhodnocením výsledků při různých světelných podmínkách.

Klíčová slova: zpracování obrazu, řízení pohybu, kamera, robot, navigace

ABSTRACT

The aim of the bachelor thesis is an analysis of suitable image processing methods for wheeled robot motion control and its realization in the language of C++. The first part deals with a selection of appropriate image processing methods. The approaches are tested using the MATLAB tool, and only those that fulfill the corresponding requirements are selected. The second part deals with the implementation of chosen methods using the C++ library OpenCV. After that, the approved methods are debugged for possible motion control. The last part deals with tests under different lighting conditions.

Keywords: image processing, motion control, camera, robot, navigation

Děkuji Ing. Jiřímu Zátopkovi za konzultace a odborné vedení bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 METODY PRO ROZPOZNÁNÍ 3D SCÉNY	10
1.1 BAREVNÉ PROSTORY	10
1.1.1 RGB.....	10
1.1.2 CMY.....	11
1.1.3 HSV.....	11
1.1.4 LAB.....	13
1.2 PREPROCESSING	14
1.2.1 Diskrétní dvourozměrná konvoluce/Gaussovo rozmazání.....	14
1.2.2 Filtr Mean-shift	15
1.3 SEGMENTACE OBRAZU	16
1.3.1 Prahování.....	16
1.3.2 Rostoucí regiony	17
1.4 POSTPROCESSING	17
1.4.1 Morfologické operace dilatace a eroze.....	18
II PRAKTICKÁ ČÁST	20
2 VÝBĚR VHODNÝCH METOD ZPRACOVÁNÍ OBRAZU	21
2.1 POPIS POUŽITÉHO KAMEROVÉHO SYSTÉMU	21
2.1.1 Kamera AXIS 206W	21
2.1.2 WIFI router ASUS WL-530GV2	22
2.2 TESTY METOD ZPRACOVÁNÍ OBRAZU V JAZYKU MATLAB	23
2.2.1 Program v jazyku MATLAB	23
2.2.2 Získaná data	25
2.3 VYHODNOCENÍ VÝSLEDKŮ	27
3 REALIZACE METOD ZPRACOVÁNÍ OBRAZU V C++	29
3.1 PROGRAM.....	29
3.2 GRAFICKÉ PROSTŘEDÍ PROGRAMU.....	31
4 OTESTOVÁNÍ A ODLADĚNÍ ALGORITMŮ PRO ŘÍZENÍ V REÁLNÉM ČASE A VYHODNOCENÍ VÝSLEDKŮ	34
4.1 TESTY ZPRACOVÁNÍ OBRAZU V REÁLNÉM ČASE	34
4.2 VYHODNOCENÍ VÝSLEDKŮ PŘI RŮZNÝCH SVĚTELNÝCH PODMÍNKÁCH	36
ZÁVĚR	42
SEZNAM POUŽITÉ LITERATURY	43
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	45
SEZNAM OBRÁZKŮ	46
SEZNAM PŘÍLOH	48

ÚVOD

Zpracování obrazu je nedílnou součástí při aplikaci autonomních komerčních a průmyslových robotů za účelem jejich navigace, bez které by nemohly z velké části fungovat. Proto jsem si vybral téma bakalářské práce zabývající se právě zpracováním obrazu.

Cílem této práce je analýza a naprogramování vhodných metod zpracování obrazu pro potenciální řízení pohybu kolového robotu v reálném čase.

Teoretická část bakalářské práce, tedy první kapitola, se zabývá popisem vhodných barevných prostorů a metod zpracování obrazu. Druhou kapitolou se přesouváme do praktické části, kde jsou v programovém prostředí MATLAB testovány a vybrány pouze ty barevné prostory a metody zpracování obrazu, které jsou použitelné pro řízení v reálném čase. Třetí kapitola se zabývá naprogramováním vybraných metod z druhé kapitoly v jazyku C++. Je vytvořeno grafické prostředí, pomocí kterého může uživatel v reálném čase vybírat a měnit jednotlivé metody zpracování obrazu a upravovat jednotlivé parametry metod. Nakonec je poslední, čtvrtá kapitola věnována odladění algoritmů pro řízení v reálném čase a vyhodnocení výsledků při různých světelných podmínkách.

Bakalářská práce je vypracována tak, aby byla možná použít pro budoucí bakalářské a diplomové práce. Samotné zpracování obrazu lze použít při sestavení nového robotického systému nebo po oživení systému aktuálního.

I. TEORETICKÁ ČÁST

1 METODY PRO ROZPOZNÁNÍ 3D SCÉNY

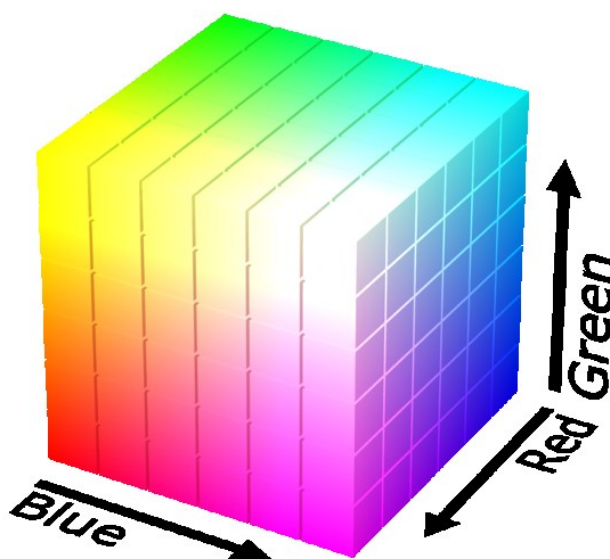
V této části budou představeny vybrané barevné prostory, jednotlivé části zpracování obrazu, jejich porovnání a výběr vhodných prostorů a metod pro zpracování obrazu za účelem řízení pohybu.

1.1 Barevné prostory

Barevný prostor je matematický model, který popisuje barvy na základě podílů jednotlivých složek barev. Nejdříve budou popsány nejznámější barevné modely, tedy RGB a CMYK. Dále se kapitola zaměřuje na barevné modely HSV a LAB, které se velmi často využívají v praxi v oblasti zpracování obrazu.

1.1.1 RGB

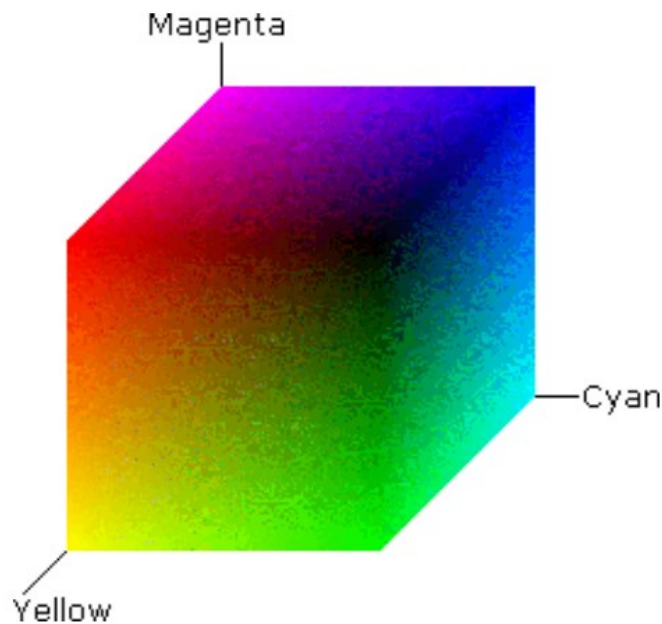
Barevný prostor RGB je tvořen třemi barevnými složkami (Red – červená, Green – zelená, Blue – modrá). Barvy jsou definovány vektorem o třech složkách v intervalu $\langle 0,1 \rangle$, popřípadě v celočíselném rozsahu (0-255). Při hodnotě 0 tuto složku výsledná barva neobsahuje a naopak při hodnotě 255 je obsažena v nejvyšší intenzitě. Pokud bude každá složka rovna hodnotě 0, získáme černou barvu. Při maximální hodnotě 255 u všech složek získáme barvu bílou.



Obrázek 1: Barevný prostor RGB [1]

1.1.2 CMY

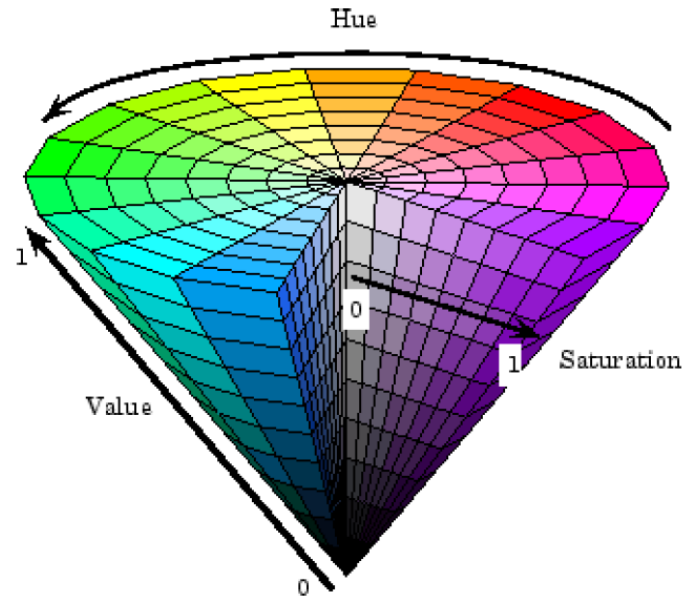
Barevný prostor CMY (Cyan – azurová, Magenta – purpurová, Yellow – žlutá) je zejména používán při tisku. Pracuje na opačném principu jako prostor RGB, tedy přidáváním intenzity barevné složky se snižuje hodnota jasu bílé barvy a výsledná barva se ztmavuje [2].



Obrázek 2: Barevný prostor CMY [3]

1.1.3 HSV

Velmi oblíbeným barevným prostorem využívaným pro segmentaci obrazu je HSV – celým názvem Hue-Saturation-Value v překladu Barva-Sytost-Hodnota. Důležitá vlastnost tohoto prostoru je, že má oddělenou složku hodnoty jasu od sytosti, tudíž k sobě dokážeme přiřadit barevné body nezávisle na jejich intenzitě jasu.



Obrázek 3: Barevný prostor HSV [4]

Převod z RGB do HSV:

$$\theta = \cos^{-1} \left(\frac{(R - G) + (R - B)}{2 \cdot \sqrt{(R - G)^2 + (R - B) \cdot (G - B)}} \right) \quad (1)$$

$$H = \begin{cases} \theta & \text{pokud } B \leq G \\ 360 - \theta & \text{pokud } B > G \end{cases} \quad (2)$$

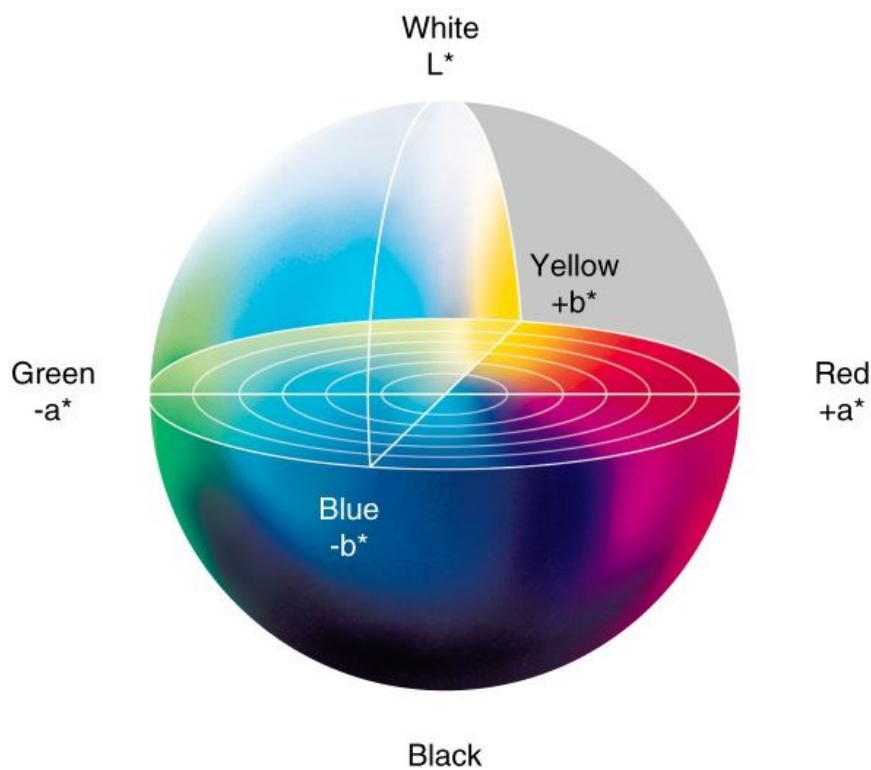
$$S = 1 - \frac{3 \cdot \min(R, G, B)}{R + G + B} \quad (3)$$

$$V = \frac{R + G + B}{3} \quad (4)$$

Hodnota H reprezentuje úhel v kruhovém HSV prostoru a je definována v rozsahu $\langle 0, 360^\circ \rangle$, saturace S i intenzita V jsou definovány v rozsahu $\langle 0, 1 \rangle$. Před převodem z RGB do HSV musíme normalizovat složky RGB z rozsahu $\langle 0, 255 \rangle$ do rozsahu $\langle 0, 1 \rangle$ [4].

1.1.4 LAB

Barevný prostor CIELAB nebo také L^*A^*B se velmi často aplikuje v oblastech zabývajících se zpracováním obrazu. Oproti barevnému prostoru RGB, který je tvořen 3 složkami barvy, je tvořen jednou složkou světlosti a dvěma složkami barvy (Lightness - světlost na intervalu (0% až 100%), A – barevná osa zelená až červená na intervalu $(-a$ až $+a)$, B – barevná osa modrá až žlutá na intervalu $(-b$ až $+b)$). Hodnota 0% u světlosti odpovídá černé barvě, hodnota 100% odpovídá barvě bílé. [5, 6].



Obrázek 4: Barevný prostor LAB [6]

1.2 Preprocessing

V obrazové analýze je preprocessing oblast, která se zabývá úpravou obrazových dat před jejich zpracováním. V našem případě se bude jednat o potlačování nežádoucích vlastností obrazu, tedy budeme odstraňovat z obrazu šum. Představíme si metody Diskrétní dvou-rozměrné konvoluce využívající Gaussovo konvoluční jádro nebo metodu využívající filtr Mean-shift.

1.2.1 Diskrétní dvourozměrná konvoluce/Gaussovo rozmazání

Konvoluce je operátor, který ze dvou funkcí vytvoří funkci novou. V případě diskrétní dvourozměrné konvoluce a zpracování obrazu má následující tvar:

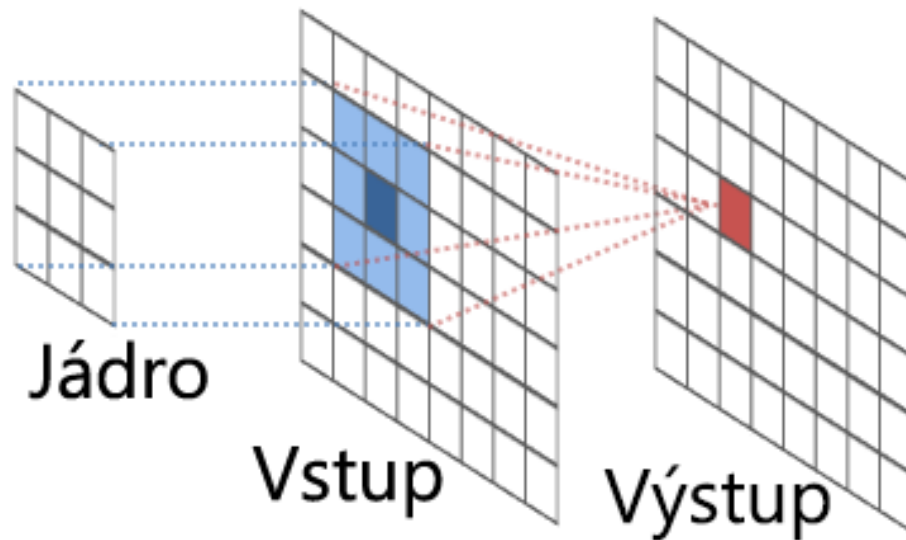
$$I(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(x-i, y-j) \cdot g(i, j) \quad (5)$$

kde $I(x, y)$ značí intenzitu výsledného pixelu na pozici $[x, y]$, $f(x, y)$ značí intenzitu vstupního obrázku na pozici $[x, y]$ a $g(i, j)$ je intenzita bodu v masce (také nazývaná jako konvoluční jádro) na pozici $[i, j]$.

Pro potlačení šumu v obraze je použito Gaussovo konvoluční jádro s následujícím tvarem:

$$G(x, y) = \frac{1}{2 \cdot \pi \cdot \sigma^2} \cdot e^{-\frac{x^2+y^2}{2 \cdot \sigma^2}} \quad (6)$$

Kde (x, y) označují relativní pozici v konvolučním jádře od prostředního prvku. σ je směrodatná odchylka. Aby nedocházelo ke zvyšování jasů, musí být součet všech prvků v jádře roven jedné. Zvolením velikosti masky a diskretizací ve středových bodech jednotlivých pixelů získáme Gaussovu matici pro filtraci obrazu (masku) [7].



Obrázek 5: Na vstupní obraz je aplikováno konvoluční jádro, které upraví výstupní obraz podle nastavených parametrů jádra [8]

1.2.2 Filtr Mean-shift

Zobrazením bodů do vícedimenzionálního prostoru příznaků, získáme kvantifikovatelné vlastnosti obrazu. Tyto vlastnosti jsou například souřadnice bodů nebo jejich barevná informace. Po provedení tohoto zobrazení pro celý obraz získáme shluky, které představují jednotlivé objekty obrazu. Analýzou tohoto prostoru lze tyto shluky určit.

Filtr Mean-shift posouvá barevnou hodnotu aktuálního obrazu ke střední hodnotě okolních bodů. Na počátku algoritmu je náhodně zvolena pozice v obraze, pomocí zvoleného kernelu se vypočítá gradient hustoty bodů v daném okolí a tuto pozici posuneme ve směru gradientu. Tento postup opakujeme do okamžiku dosáhnutí lokálního maxima. Filtr má tu výhodu, že zachovává ostré hrany mezi objekty s rozdílnou barvou [9, 10].



Obrázek 6: Porovnání obrázku před a po použití filtru mean-shift [11]

1.3 Segmentace obrazu

Segmentace obrazu se skládá z několika kroků, s pomocí kterých můžeme obraz rozdělit do několika oblastí, podle určitých společných vlastností. Nejčastěji se jedná o oddělení objektu od pozadí.

1.3.1 Prahování

Metoda prahování je nejjednodušší způsob segmentace obrazu. Jako určující parametr u metody prahování se ve většině případů využívá jas obrazu. Metoda prahování funguje na principu určení prahu a následném porovnáním každého pixelu, kdy pixely s menší hodnotou intenzity jasu jsou považovány za pixely pozadí a pixely s větší intenzitou jsou považovány jako pixely objektu. Tato metoda je časově nenáročná, díky získání výsledku již po prvním průchodu obrazu.

Problém může nastat v případě, že intenzita vyšší než prahová, tedy teoreticky pixel reprezentující objekt se může nacházet i v oblasti pozadí a vzniknou tedy nepřesnosti. Tento problém lze odstranit v dalším zpracování pomocí postprocessingu [9].



Obrázek 7: Porovnání originálního obrázku s obrázkem, na který je aplikována metoda prahování

1.3.2 Rostoucí regiony

Metoda rostoucích regionů pracuje s každým bodem obrazu zvlášť, vytváříme tedy skupiny podobných bodů podle jejich vlastností. Hlavním kritériem metody je homogenita. Metodu lze založit na mnoha vlastnostech jako barva, intenzita jasu, tvar nebo textura. Cílem metody je získat co největší homogenní oblast bez možnosti spojení s jinou sousední oblastí. Metoda porovnává vybrané body se sousední homogenní oblastí, a pokud jsou splněny podmínky homogenity, tak se bod sloučí s porovnávanou oblastí. Při nesplnění požadavků se vytvoří nová oblast [9].



Obrázek 8: Porovnání originálního obrázku s obrázkem, na který je aplikována metoda rostoucích regionů [12].

1.4 Postprocessing

Tato oblast se zabývá dodatečnou úpravou dat po tom, co obraz projde celým cyklem zpracování. Může to být například vyřešení problému u segmentace obrazu, kdy se pixely přiřadí do špatné oblasti, sloučení menších oblastí do oblastí, které mají podobné vlastnosti nebo odstraňování míst, které se vlastnostmi nikam nehodí [13].

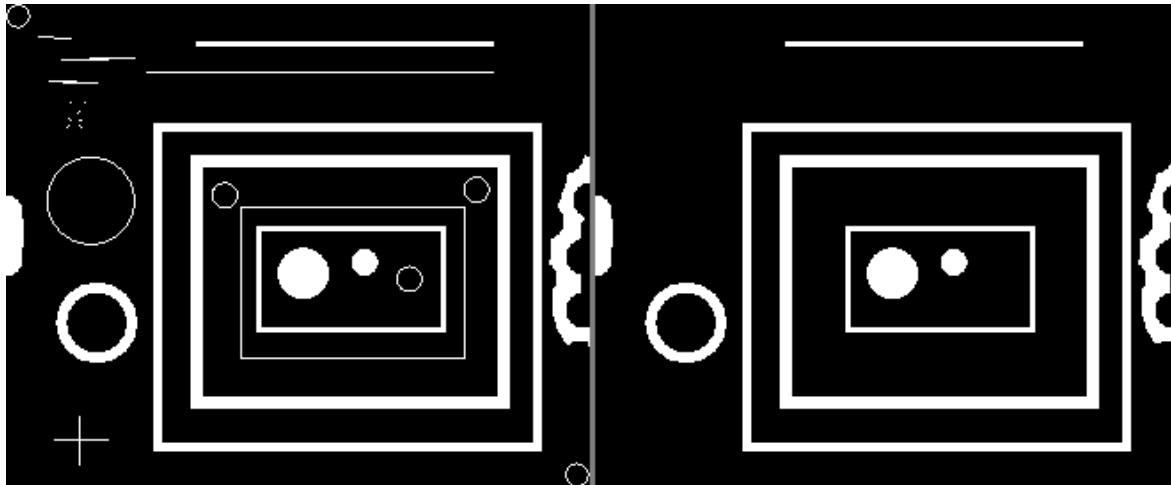
1.4.1 Morfologické operace dilatace a eroze

Dilatace a eroze jsou operace z oblasti matematické morfologie, které se aplikují jako dodatečná úprava obrazu. Tyto operace se využívají k zaplnění děr a zjednodušení celkové struktury objektu.

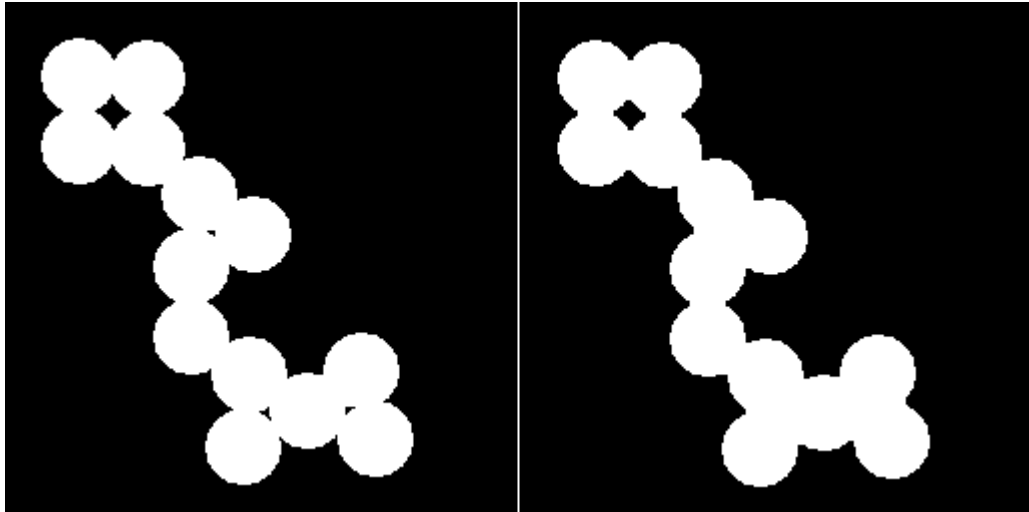
Dilatace je vektorový součet dvou množin, kdy vybraná oblast převezme vlastnosti z oblasti v blízkém okolí s maximální hodnotou. Dilatace obraz zvětší o jednu vrstvu a zaplní malé díry.

Eroze je vektorový rozdíl dvou množin, kdy vybraná oblast převezme vlastnosti z oblasti v blízkém okolí s minimální hodnotou. Eroze odstraní objekty jednotkové tloušťky, spojí objekty blízko u sebe a rozdělí složité objekty na objekty jednodušší.

Pro získání obrysů objektu stačí odečíst erodovaný obraz od originálu. Tyto operace můžeme také společně kombinovat a získáme zjednodušený obraz. Při erozi a následné dilataci získáme takzvané morfologické otevření, které rozděljuje objekty spojené tenkou čarou a odstraní šum. Naopak dilatací a následnou erozí dostáváme morfologické uzavření, které spojí blízké objekty, zaplní díry a vyhladí obrysy [14, 15].



Obrázek 9: Ukázka morfologického otevření – vlevo originál a vpravo po použití morfologického otevření [15]



Obrázek 10: Ukázka morfologického uzavření – vlevo originál a vpravo po použití morfologického uzavření [15]

II. PRAKTICKÁ ČÁST

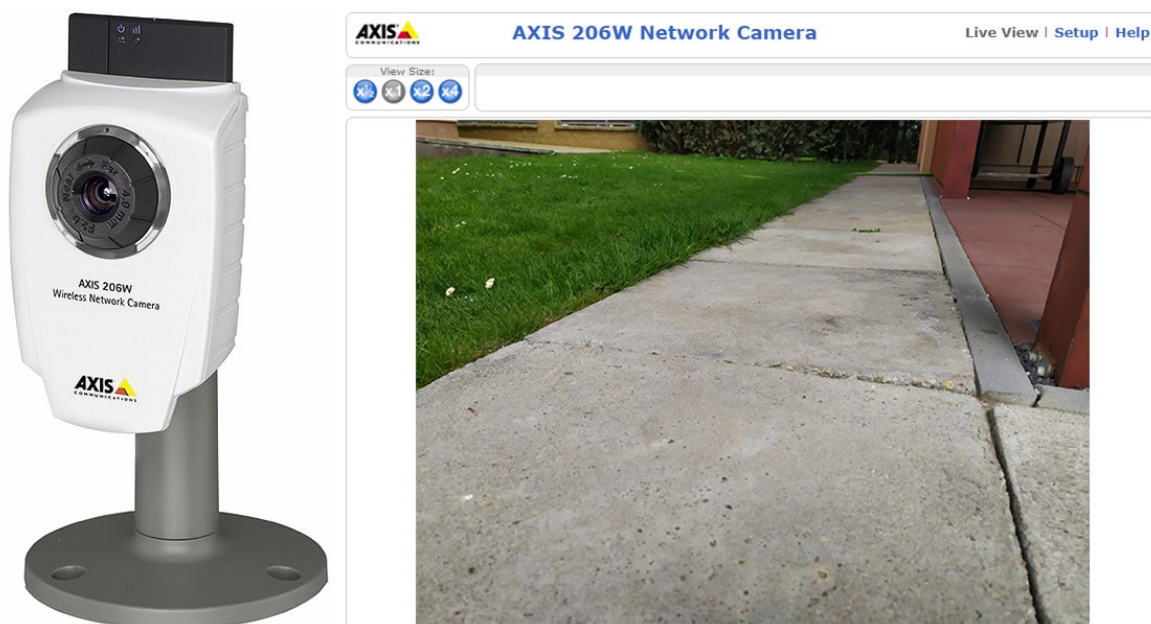
2 VÝBĚR VHODNÝCH METOD ZPRACOVÁNÍ OBRAZU

2.1 Popis použitého kamerového systému

Pro získání obrazové informace byla použita kamera AXIS 206W, která je součástí již sestaveného robotického systému použitého v jiných bakalářských a diplomových pracích. Samotná implementace nové komunikace mezi kamerou a motory robotu a tedy řízení robotu, by mohla být tématem navazující diplomové práce.

2.1.1 Kamera AXIS 206W

Kamera AXIS 206W zpracovává 30 snímků za sekundu v maximálním rozlišení 640x480. Ke sledování obrazu můžeme využít webový prohlížeč, jelikož má kamera zabudovaný webový server. [16]



Obrázek 11: kamera AXIS 206W [16] a webové prostředí kamery

Po přihlášení do webového prostředí kamery si můžeme nastavit hodnoty nahrávaného obrazu. Lze nastavit kvalita obrazu změnou komprese, nastavení rozlišení, popřípadě výběr mezi kontinuálním záznamem snímků tvořící video nebo generováním jednoho snímku při aktualizaci stránky.

2.1.2 WIFI router ASUS WL-530GV2

Připojení kamery k internetu obstarává WIFI router ASUS WL-530GV2. Využívá standard IEEE 802 a zabezpečení je obstaráváno 128bitovým WPA šifrováním.

V prohlížeči se dostaneme na adrese 192.168.1.1. do webového prostředí nastavení routeru. Zjistíme tady potřebné údaje, které následně vložíme do softwarového prostředí webové kamery AXIS Setup Tool. Do prostředí se dostaneme připojením kamery do PC pomocí USB. Nastavíme statickou IP adresu, pomocí které si ve webovém prohlížeči zobrazíme obraz z kamery.



Obrázek 12: WIFI router ASUS WL-530GV2

2.2 Testy metod zpracování obrazu v jazyku MATLAB

Testy byly prováděny na obrazových informacích pořízených kamerou v reálném prostředí na chodníku v pěti polohách. Testy se provedly v programovém prostředí MATLAB na všech kombinacích barevných prostorů a prvních dvou částí zpracování obrazu, tedy metod preprocessingu a metod segmentace obrazu. Samotné testy s kombinací metod postprocessingu už nejsou potřeba, protože kvalitu zpracování obrazu neovlivňují. Po otestování všech kombinací se vyberou barevné prostory a metody zpracování obrazu, které splňují naše požadavky pro zpracování obrazu za účelem navigace robotu.

2.2.1 Program v jazyku MATLAB

První část programu se zabývá připojením ke kameře, kdy se zadává postupně zleva IP adresa, přihlašovací jméno a přihlašovací heslo. Druhý řádek zajišťuje samotné pořízení snímku, který se bude dále zpracovávat.

```
1: camera=ipcam('http://192.168.1.3/axis-  
cgi/mjpg/video.cgi?resolution=640x480', 'root', 'admin');  
2: rgb=snapshot(camera);
```

Následuje část s preprocessingem, kde se zavolá jedna z funkcí pro předzpracování obrazu. Na výběr je metoda Gaussova rozmazání nebo metoda Mean-shift. U metody Gaussova rozmazání se jako druhý parametr zadává velikost jádra.

```
1: %%gaussian blur%%  
2: preprocessing = imgaussfilt(rgb,2);  
3:  
4: %%mean-shift filter%%  
5: preprocessing = Msfilter(rgb);
```

Dále se vybere, do kterého barevného prostoru se obraz převede.

```
1: %%rgb2hsv%%  
2: hsv = rgb2hsv(preprocessing);  
3: image = hsv;  
4: %%rgb2cmy  
5: cmy = RGB2CMY(preprocessing);  
6: image = cmy;  
7: %%rgb2lab  
8: lab = rgb2lab(preprocessing);  
9: image = lab;  
10: %%rgb  
11: image = preprocessing;
```

Po převodu barevného prostoru se aplikuje jedna ze segmentačních metod. V našem případě metoda prahování nebo rostoucích regionů. U metody prahování nás zajímá druhý parametr ve funkci `fspecial`, která vytváří předdefinovaný filtr. Ve druhé parametru upřesňujeme velikost samotného filtru. Potom se od originálního obrazu odečte obraz vyfiltrovaný a výsledný obraz se převede na černobílý. U metody rostoucích regionů do vstupních parametrů zadáváme souřadnice počátečního bodu a maximální rozdíl intenzity sousedního bodu. Nakonec jako u metody prahování převedeme výsledný obraz na černobílý

```
1:      %%threshold metod%%
2:      mean_image =
imfilter(image,fspecial('average',[300,300]),'replicate');
3:      subtract = image-(mean_image+0.005);
4:      black_white = im2bw(subtract,0);
5:
6:      %%region growing metod%%
7:      I = image;
8:      x = 320;
9:      y = 400;
10:     J = regiongrowing(I,x,y,0.05);
11:     rg = I+J;
12:     black_white = im2bw(rg-image,0);
```

Nakonec je obraz dodatečně upraven v části postprocessingu, kde se odstraní malé objekty, zaplní se díry a výsledný zpracovaný obraz se zobrazí na obrazovce. U morfologických operací se jako první parametr zadává tvar strukturního prvku, se kterým se bude dále pracovat, druhý parametr specifikuje poloměr tohoto prvku.

```
1:      %%morphological operations - dilation and erosion%%
2:      %%morphological closing%%
3:      originalBW = black_white;
4:      se = strel('disk',10);
5:      closeBW = imclose(originalBW,se);
6:
7:      %%morphological opening%%
8:      se = strel('disk',10);
9:      openBW = imopen(closeBW,se);
10:
11:     %%reversing black and white%%
12:     reverseBnW=imcomplement(openBW);
13:
14:     %%deleting every object except the biggest%%
15:     BW4 = bwareafilt(reverseBnW,1);
16:     BW5=imfill(BW4, 'holes');
17:
18:     %%printing processed image%%
19:     close
20:     imshow(BW5);
```


2.2.2 Získaná data

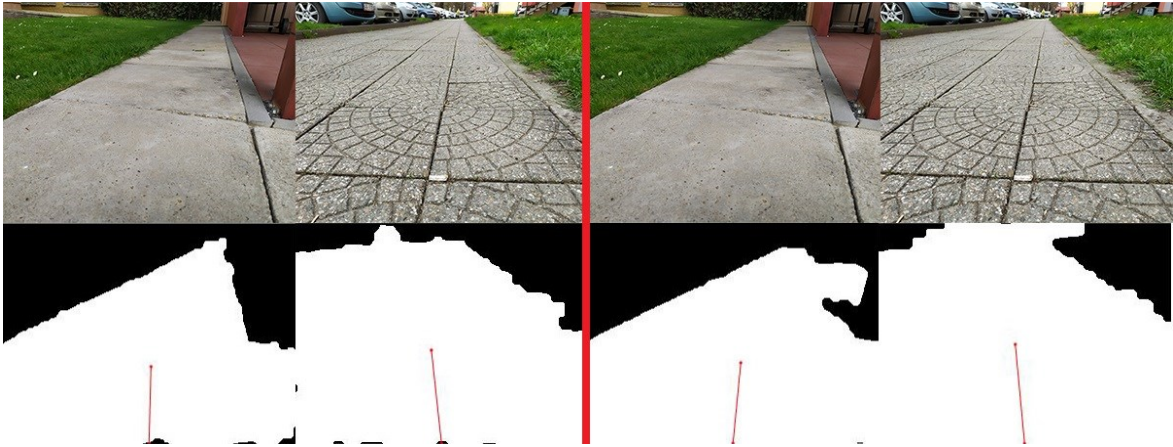
Pro testy byly zvoleny fotografie zahradního a pouličního chodníku u parkoviště. Zahradní chodník má jemnou povrchovou texturu, z jedné strany je obklopen travnatým porostem a z druhé strany je členitý prostor venkovního posezení. Oproti tomu pouliční chodník je tvořen dlaždicemi se vzorem a nepravidelným pozadím. Další fotografie obsahují stejné chodníky, ale navíc se stínem, který zasahuje přes celou šířku chodníku.



Obrázek 13: Testovaný zahradní a pouliční chodník

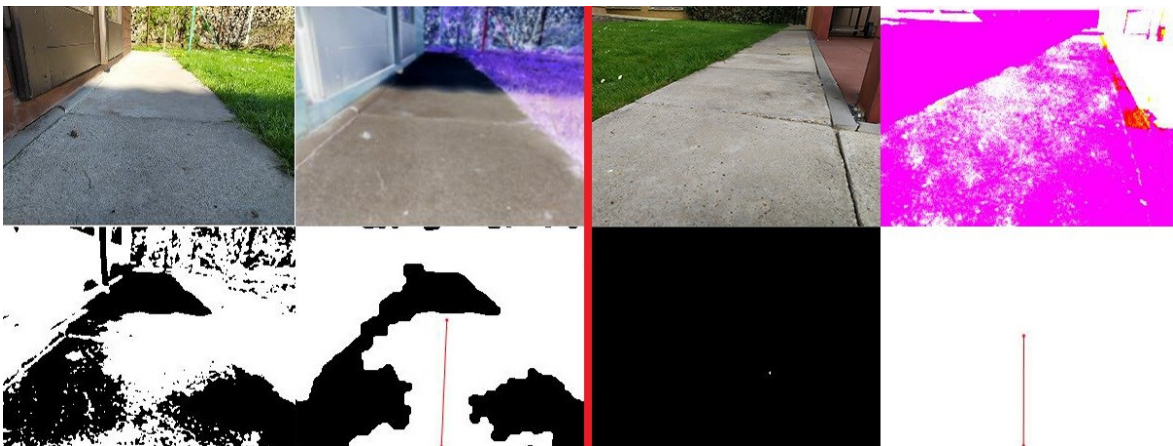
Jako nejspolehlivější se ukázala kombinace preprocessingu pomocí Gaussova rozmazání, barevného prostoru HSV a segmentační metody rostoucích regionů viz Obrázek 14.

Další velmi dobrá kombinace metod zpracování obrazu byla kombinace preprocessingu pomocí filtru Mean-shift, převodem do barevného prostoru LAB a použití segmentační metody rostoucích regionů viz Obrázek 14.



Obrázek 14: Levá čtveřice: aplikace Gaussova rozmazání, převodu do HSV a metody rostoucích regionů. Pravá čtveřice: aplikace filtru Mean-shift, převodu do LAB a metody rostoucích regionů

Překvapivě dobrých výsledků jsme dosáhli i s použitím základního RGB barevného prostoru. Oproti tomu nejhorší výsledky byly získány s použitím barevného prostoru CMY v jakékoli kombinaci, kdy byla u většiny případů nalezena necelistvá cesta nebo jednobarevný výstupní obraz, který nelze použít k navigaci. Také jako nepoužitelná se jeví kombinace Gaussova rozmazání s rostoucími regiony ve chvíli, kdy je použit barevný prostor LAB. Výsledný zpracovaný obraz je vždy celý pouze jednobarevný a tedy nepoužitelný z hlediska zpracování obrazu viz Obrázek 15.



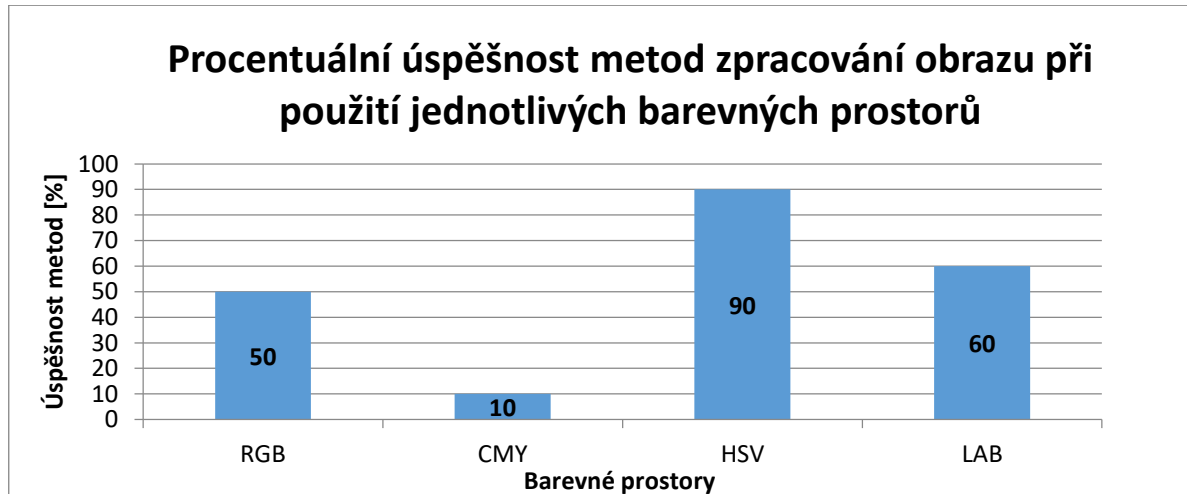
Obrázek 15: Výše popsané zpracování: levá čtveřice s prostorem CMY, pravá s prostorem LAB.

2.3 Vyhodnocení výsledků

Celkově bylo provedeno 16 různých kombinací mezi všemi barevnými prostory a všemi metodami preprocessingu a segmentace obrazu na pěti fotografiích. Dohromady bylo získáno 80 výsledků, tedy pro každý barevný prostor 20.

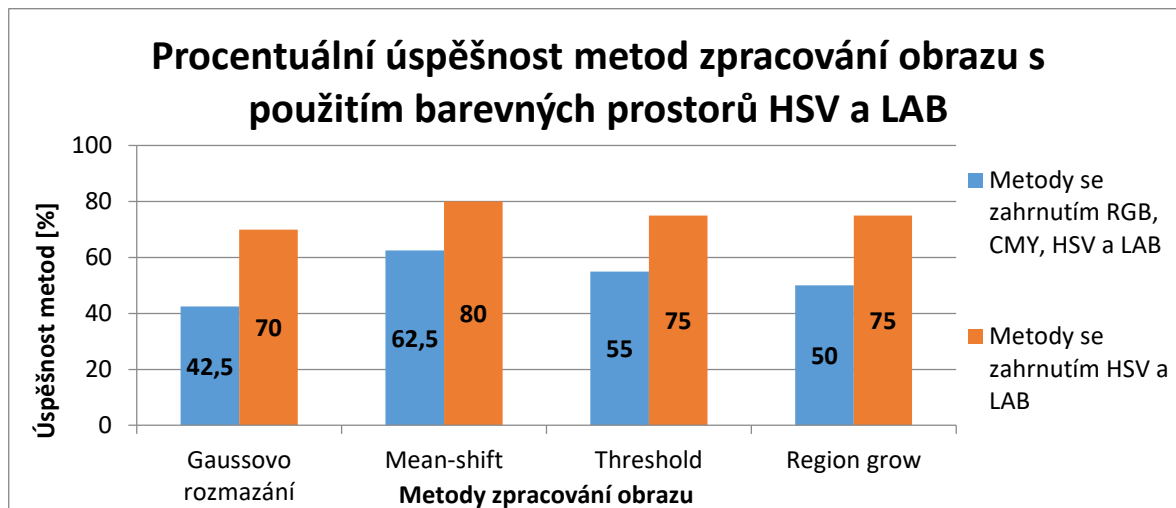
Jakožto dostatečný výsledek pro posouzení správného zpracování obrazu za účelem navigace stačilo, aby byla nalezena celistvá cesta ve spodní třetině obrazu, tedy v nejbližších místech, kterými případně robot projede.

Nejlépe si se zpracováním poradil barevný prostor HSV, který správně našel cestu v 18 z 20 případů. Další barevný prostor LAB správně zpracoval 12 z 20 fotografií, kdy měl velké potíže při kombinaci s Gaussovým rozmazáním a metodou rostoucích regionů. Barevný prostor RGB si překvapivě dokázal poradit s polovinou kombinací, tedy 10 z 20 fotografií, kdy jsme získali velmi dobré výsledky s použitím filtru mean-shift a metodou rostoucích regionů. Nakonec při použití barevného prostoru CMY jsme našli cestu ve 2 případech z 20. Z výsledků vyplývá, že nemá cenu implementovat barevný prostor CMY do jazyku C++, viz Obrázek 16.



Obrázek 16: Graf procentuální úspěšnosti metod zpracování obrazu pro všechny barevné prostory při použití všech kombinací metod zpracování obrazu

Pokud porovnáme výsledky jednotlivých metod zpracování obrazu pouze u barevných prostorů HSV a LAB a vyřadíme barevné prostory CMY a RGB, které by výsledky zkreslovaly, tak získáme téměř totožné výsledky u všech. Jak metody preprocessingu, tak segmentace mají (70-80)% úspěšnost zpracování obrazu pro nalezení cesty viz Obrázek 17.



Obrázek 17: Graf porovnávající procentuální úspěšnost metod zpracování obrazu s použitím barevných prostorů HSV a LAB

V následující kapitole realizace metod zpracování obrazu v jazyku C++ budou tedy využity všechny zmíněné metody zpracování obrazu a barevné prostory HSV a LAB, se kterými jsme získali konzistentně dobré výsledky v oblasti vizuální kvality zpracování obrazu.

3 REALIZACE METOD ZPRACOVÁNÍ OBRAZU V C++

Realizace metod zpracování obrazu v jazyku C++ byla uskutečněna ve vývojovém prostředí Qt Creator, s využitím open source knihoven OpenCV, které jsou zaměřeny na zpracování obrazu.

3.1 Program

V první části programu se zabýváme načtením kamery z adresy a následným čtením snímků.

```
1:   VideoCapture vcap;
2:   Mat image;
3:   const string videoStreamAddress =
"http://root:admin@192.168.1.3/axis-
cgi/mjpg/video.cgi?resolution=640x480";
4:   while(true){
5:       vcap.read(image);
6:       .
7:       .
```

Jakmile jsme získali obraz z kamery, můžeme začít na obraz aplikovat metody zpracování obrazu. Nejdříve je vybrána metoda preprocessingu (Gaussovo rozmazání nebo metoda Meanshift).

```
1:       if (Preprocessing_type == 0){
2: //gaussian blur
3:         gauss_odd=Gauss_value;
4:         if (gauss_odd == 0){
5:             setTrackerPos ("Gauss",window_name,Gauss_value+1);
6:         }else{
7:             count_odd = gauss_odd % 2;
8:             if(count_odd == 0){
9:                 setTrackerPos ("Gauss",window_name,Gauss_value+1);
10:            }
11:        }
12:        GaussianBlur(image, img_Blur, Size(Gauss_value,Gauss_value), 0, 0);
13:        gauss_odd = getTrackerPos ("Gauss",window_name);
14:    }else{
15: //meanShift
16:        MeanShift MSProc(ms_bandwith,ms_clr_bandwith);
17:        MSProc.MSFiltering(img_Blur);
18:    }
```

Následně se vybere barevný prostor. Na výběr je tedy originální RGB, HSV nebo LAB.

```
1:         if(Color_type == 2){
2: //rgb2lab
3:         cvtColor(img_Blur, img_final_color, COLOR_BGR2Lab);
4:         }else if(Color_type == 1){
5: //rgb2hsv
6:         cvtColor(img_Blur, img_final_color, COLOR_BGR2HSV);
7:         }else{
8: //original - rgb
9:         img_final_color = img_Blur;
10:        }
```

Poté se vybere metoda segmentace obrazu (metoda Prahování nebo Rostoucích regionů).

```
1:         if(Segmentation_type==0) {
2: //threshold
3:         Canny(img_final_color, img_Segmented,
Canny_value1, Canny_value2);
4:         }else{
5:         if(Clusters_value == 0){
6:         setTrackbarPos("Km
clstrs", window_name, Clusters_value+1);
7:         }
8: //k-means (region growing)
9:         Input_Image = img_final_color;
10:        img_Segmented = K_Means(Input_Image, Clusters_value);
11:        cvtColor(img_Segmented, img_Segmented, COLOR_HSV2BGR);
12:        cvtColor(img_Segmented, img_Segmented, COLOR_BGR2GRAY);
13:        }
```

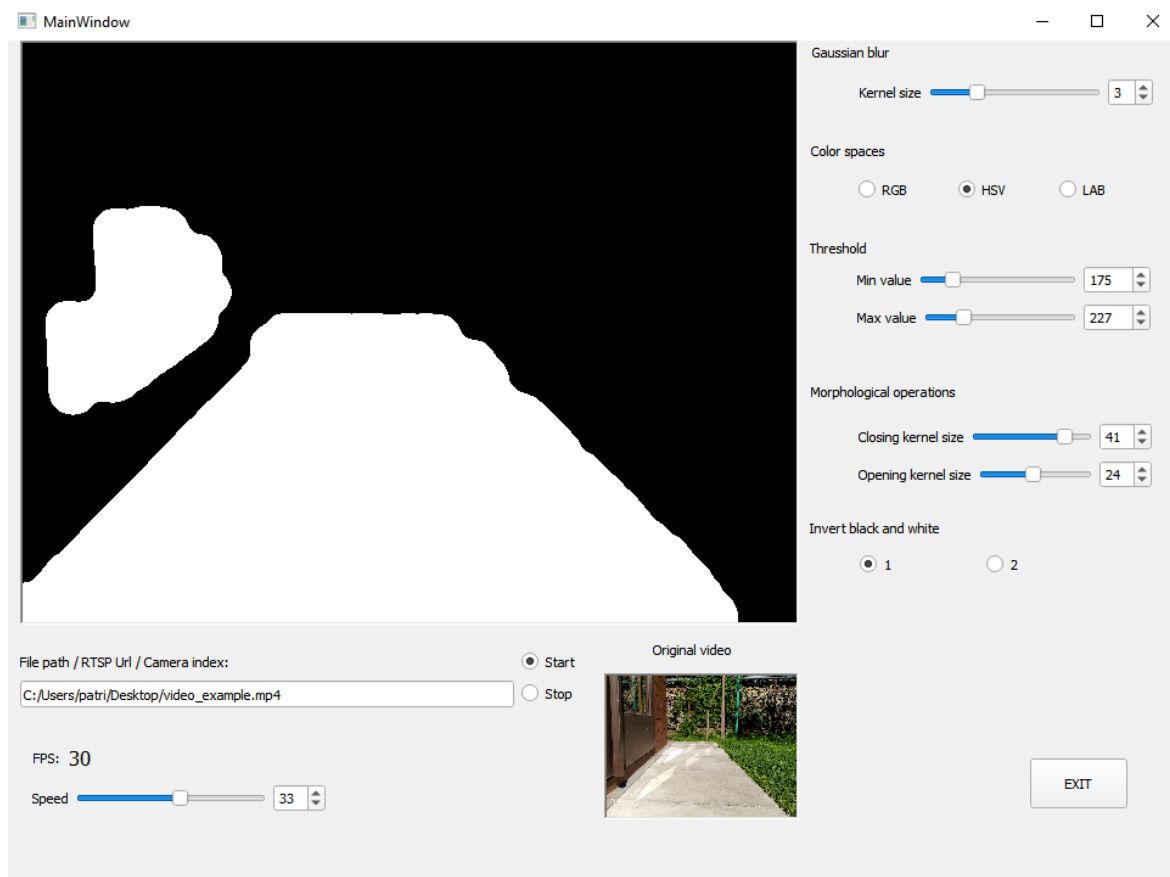
Nakonec se na obraz aplikují metody postprocessingu. Uzavřou se díry, odstraní se malé objekty, vymění se černé a bílé pixely a uloží se právě zpracovaný snímek na disk pro případné použití v oblasti navigace a řízení pohybu robota.

```
1:         if(kernel_size==0) {
2:         setTrackbarPos("MorKernel", window_name, kernel_size+1);
3:         }
4:         kernel = getStructuringElement(MORPH_ELLIPSE,
Size(kernel_size, kernel_size));
5: //morphological closing
6:         morphologyEx(img_Segmented, img_Close, MORPH_CLOSE, kernel);
7: //morphological opening
8:         morphologyEx(img_Close, img_Open, MORPH_OPEN, kernel);
9:         bitwise_not(img_Open, img_Open);
10:        imwrite("processed_image.jpg", img_Open);
```

3.2 Grafické prostředí programu

Grafické prostředí programu bylo vytvořeno ve vizuálním editoru Qt Creator Design, který je integrovaný v samotném vývojovém prostředí Qt Creator. Do této aplikace byly implementovány všechny metody zpracování obrazu a barevné prostory, které prošly přes testy zpracování obrazu v Matlabu a jejich následné vyřazování bude probíhat v kapitole 4.1. Dále budou popsány jednotlivé oblasti grafického prostředí a jejich význam. Hlavní část okna zabírá zpracovaný obraz, v dolní části se ovládá chod samotného videa a pravý sloupec obsahuje atributy k nastavení.

Grafické prostředí obsahuje v levé horní části výstup právě zpracovávaného obrazu, na který jsou aplikovány již všechny metody zpracování obrazu. Dole vpravo se nachází tlačítko EXIT, pomocí kterého zastavíme načítání obrazu a zavřeme okno aplikace.



Obrázek 18: Náhled na celou plochu grafického prostředí aplikace

Vlevo dole se nachází část, ve které ovládáme chod samotného vstupního obrazu. Hned pod zpracovávaným obrazem se nachází textové pole, do kterého uživatel zadává cestu k souboru s videem, RTSP adresu IP kamery nebo index USB kamery popřípadě integrované notebookové kamery. Jakmile máme zadanou cestu k obrazu, můžeme pomocí tlačítka start nebo stop spouštět a vypínat čtení obrazu z právě zadané adresy. Ve spodní části se nachází informativní ukazatel počtu snímků za sekundu právě zpracovávaného vstupního obrazu. Pod tímto ukazatelem si můžeme pomocí posuvníku nastavit rychlost přehrávání zdroje obrazu. Jedná se o nastavení hodnoty funkce *waitKey(int delay)*, pomocí které můžeme garantovat minimální dobu zpoždění dalšího cyklu, tedy dalšího načtení snímku. Hodnota se zadává v milisekundách. Pokud chceme zpracovávat video umístěné na disku v originální rychlosti, zadáme zpoždění vypočítané pomocí následujícího vztahu:

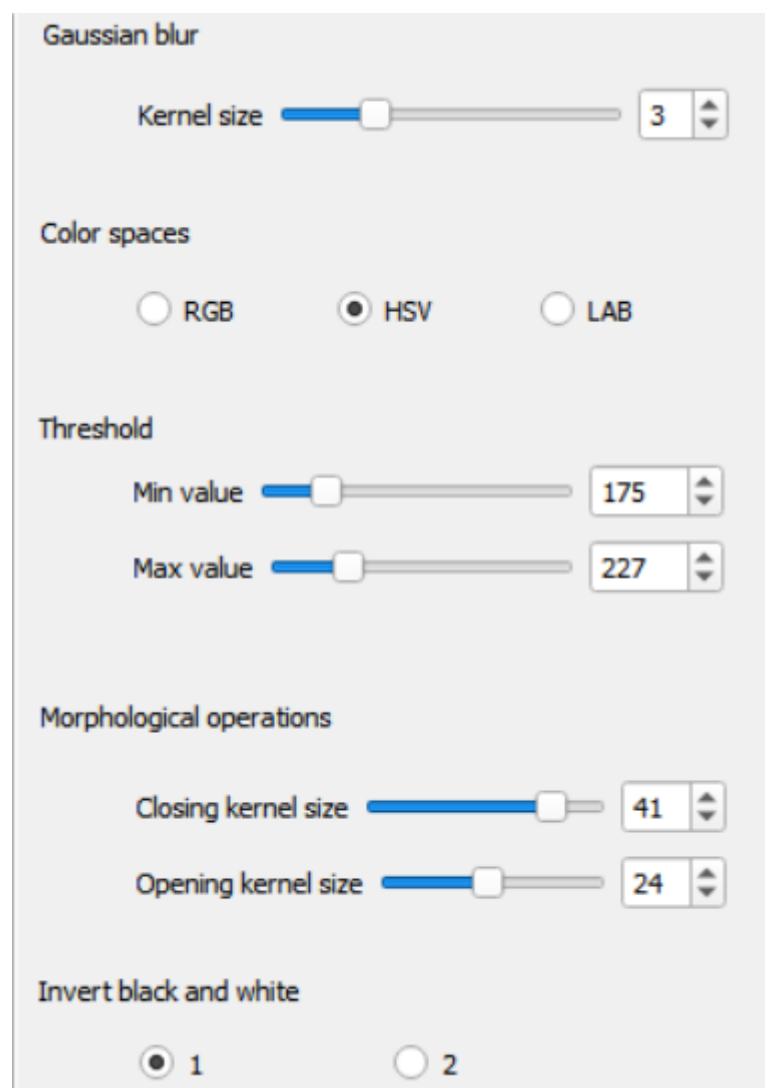
$$\text{zpoždění} \doteq \frac{1}{\text{fps originálního videa}} \cdot 1000 \quad (7)$$

Nakonec napravo od tlačítek start a stop se nachází náhled na originální snímek obrazu před zpracováním.



Obrázek 19: Bližší náhled na spodní panel grafického prostředí

V pravém panelu grafického prostředí se nachází celé nastavení metod zpracování obrazu a jejich atribut. Nejprve si lze nastavit velikost jádra Gaussova rozmazání. Potom si může uživatel zvolit, v jakém barevném prostoru se bude obraz dále zpracovávat. Pod výběrem barevného prostoru se nachází část s nastavením horní a dolní meze prahování. Dále si lze nastavit velikosti jader pro morfologické operace otevření a uzavření. Nakonec můžeme, pokud je potřeba, vyměnit černé a bílé pixely finálního zpracovaného obrazu. Číselné hodnoty lze nastavovat buďto posuvníkem nebo zadat přesnou hodnotu to přírůstkového pole. Výběr barevného prostoru nebo invertování barev je uskutečněno pomocí skupinových přepínačů.



Obrázek 20: Bližší náhled na pravý panel grafického prostředí

4 OTESTOVÁNÍ A ODLADĚNÍ ALGORITMŮ PRO ŘÍZENÍ V REÁLNÉM ČASE A VYHODNOCENÍ VÝSLEDKŮ

Jelikož je práce zaměřená na zpracování obrazu za účelem řízení v reálném čase, je potřeba jednotlivé metody otestovat, jestli budou tento požadavek splňovat. Následně budou vybrané metody, které tento požadavek splní otestovány v reálném prostředí při různých světelných podmínkách a výsledky budou vyhodnoceny.

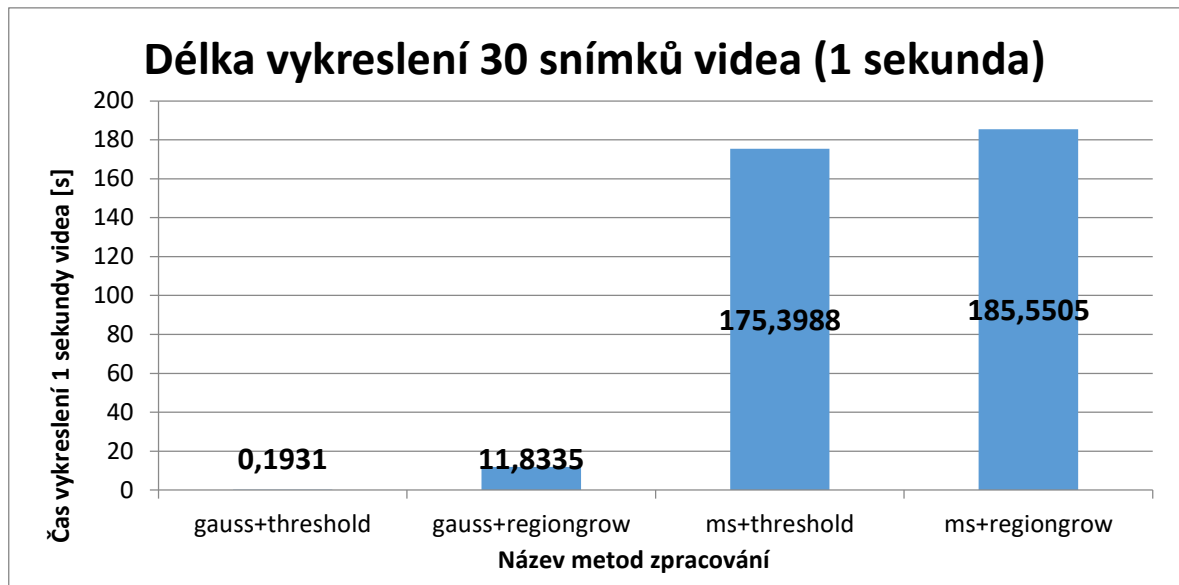
4.1 Testy zpracování obrazu v reálném čase

Jakožto dostačující požadavek pro řízení v reálném čase je, aby metody dokázaly zpracovávat obraz minimálně ve 30 snímcích za sekundu a program se nezpožďoval za snímaným obrazem.

Testy probíhaly na mém osobním počítači s procesorem Intel Core i5-8600K 3,6GHz (turbo boost 4,3GHz) a operačním systémem Windows 10 Education na verzi systému 19042.928. Proces měl nastavenou nejvyšší prioritu.

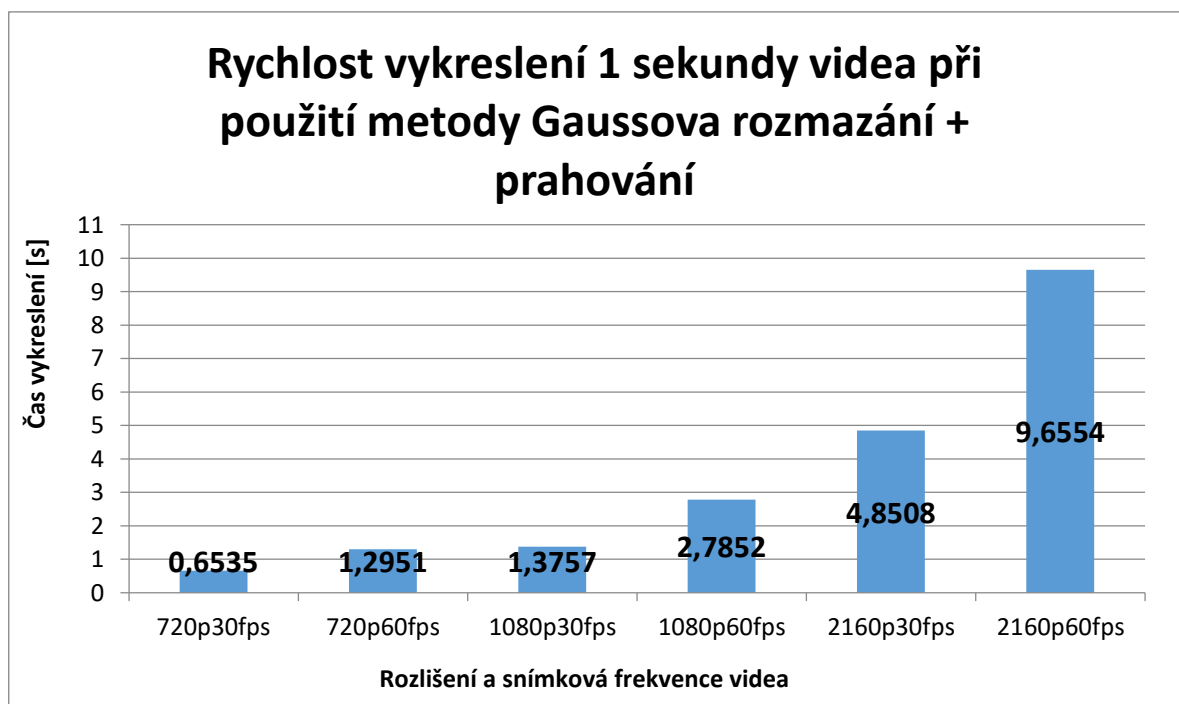
Bylo vytvořeno pětisekundové video, které se nahrálo do programu a měřilo se, za jak dlouho se celé video zpracuje, nehledě na snímkovou frekvenci videa. Stačilo pouze, aby se video zpracovalo do pěti sekund, potom se dá předpokládat, že při použití u snímání obrazu v reálném čase bude program obraz stíhat zpracovávat a nebude se opožďovat.

Všechny kombinace metod byly naměřeny pro všechny barevné prostory, tedy RGB, HSV i LAB ve třech iteracích. Výsledky pro jednotlivé barevné prostory byly téměř totožné, a proto byly výsledky jednotlivých kombinací metod zpracování pro jednotlivé barevné prostory zprůměrovány. Jakožto jediná použitelná kombinace metod zpracování obrazu se ukázala Gaussova metoda preprocessingu společně se segmentační metodou prahování, kdy bylo celé video zpracováno za 0,9 sekundy. Ostatní kombinace metod jsou, jak je z Obrázku 21 zřejmé, nepoužitelné pro zpracování v reálném čase, kdy vykreslit jediný snímek obrazu trvalo minimálně 11,8 sekundy při použití metody rostoucích regionů. Maximální čas vykreslení jednoho snímku se pohyboval okolo 180 sekund, při použití preprocessingové metody Mean-shift.



Obrázek 21: Graf závislosti kombinace metod zpracování obrazu na čase vykreslení 1 s videa ve 30 fps

Pro případ metody Gaussova rozmazání zároveň s metodou prahování byly vytvořeny testy pro otestování nejvyšší možné kvality obrazu za účelem dostatečné rychlosti zpracování v reálném čase. Jako zdroj bylo použito 2160p60fps video, u kterého bylo postupně zmenšováno rozlišení a snímková frekvence.



Obrázek 22: Graf rychlosti vykreslení 1 sekundy videa při různých kvalitách zdroje

Z Obrázku 22 vyplývá, že nejvyšší možná kvalita videa, kterou dokáže program dostatečně rychle zpracovat pro účely řízení v reálném čase je 720p30fps, tedy rozlišení 1280 x 720 px se snímkovací frekvencí 30 fps.

Z důsledku nepříznivých výsledků, které jsme získali při testování zpracování obrazu v reálném čase, byly metody Mean-shift a Rostoucí regiony odstraněny z programu a nebudou tedy použity u finálních testů při různých světelných podmínkách.

4.2 Vyhodnocení výsledků při různých světelných podmínkách

Na závěr byly provedeny testy při různých světelných podmínkách na barevných prostorech a metodách zpracování obrazu, které splňují naše požadavky vizuální kvality zpracování obrazu a zároveň dostatečné rychlosti zpracování. Jako hlavní proměnný parametr, který se při testech nastavoval, byla velikost jádra Gaussova rozmazání v rozmezí 1-9, kdy se následně upravovaly hodnoty prahování a morfologických operací, takovým způsobem než byla nalezena nejlepší možná kombinace, která podávala nejlepší možné výsledky.

Hodnoty v následujících grafech znázorňují, jak zvolený barevný prostor ovlivnil kvalitu zpracování obrazu v závislosti na světelných podmínkách. Modře jsou označeny situace, kdy se obraz zpracoval velmi dobře a byla nalezena cesta. Oranžovou barvou jsou označeny situace, kdy byla nalezena cesta, ale zároveň se k cestě připojilo také pozadí a tudíž nebyl získat dostatečně přesný výsledek. Nakonec šedou barvou jsou označeny ty výsledky, kdy se nepodařilo cestu nalézt vůbec, nebo byla spojena s pozadím takovým způsobem, že je takto zpracovaný obraz nepoužitelný viz Obrázek 23.



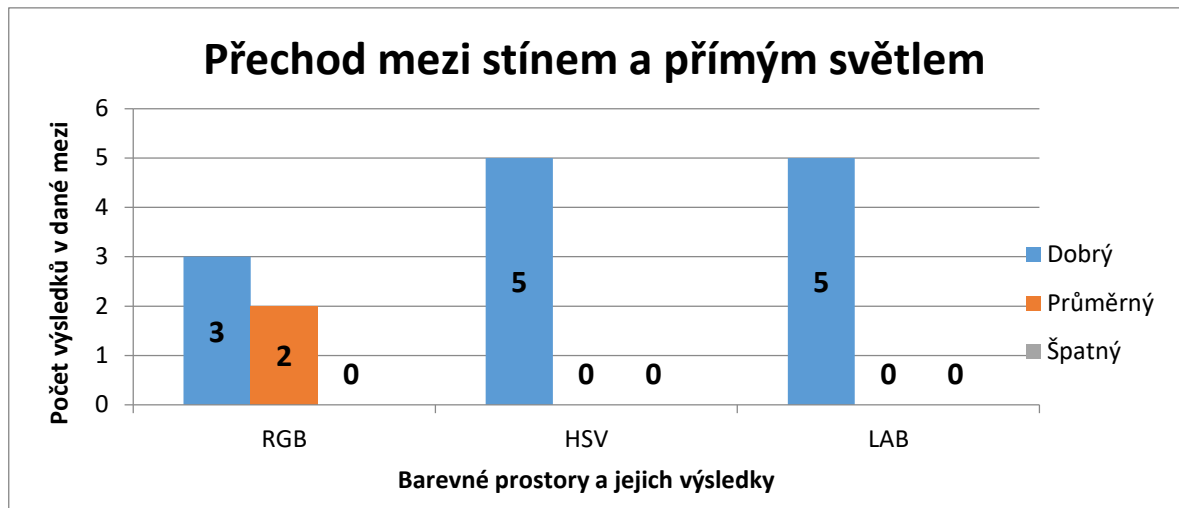
Obrázek 23: Porovnání kvality zpracování zleva: dobrý, průměrný, špatný

Nejprve byly provedeny testy na chodníku, jehož celá plocha i s nejbližším okolím byla ve stínu. U barevného prostoru RGB nastavení velikosti Gaussova rozmazání téměř neovlivňovalo kvalitu zpracování obrazu a získávali jsme velmi podobné výsledky cesty, místy propojené s pozadím. Barevný prostor HSV zajistil konzistentně dobré výsledky a našel cestu při všech velikostech rozmazání. Zároveň čím více byl obraz rozmazán (do velikosti jádra 7 včetně), tím menší interval prahování se mohl nastavit, díky tomu se ještě více vyfiltroval šum z pozadí a byly nalezeny přesnější hrany chodníku. Barevný prostor LAB zajistil výsledky velmi podobné jako při použití barevného prostoru RGB, kdy při použití Gaussova rozmazání s velikostí jádra 7 a více, se pozadí propojilo s chodníkem takovým způsobem, že byl zpracovaný obraz nepoužitelný.



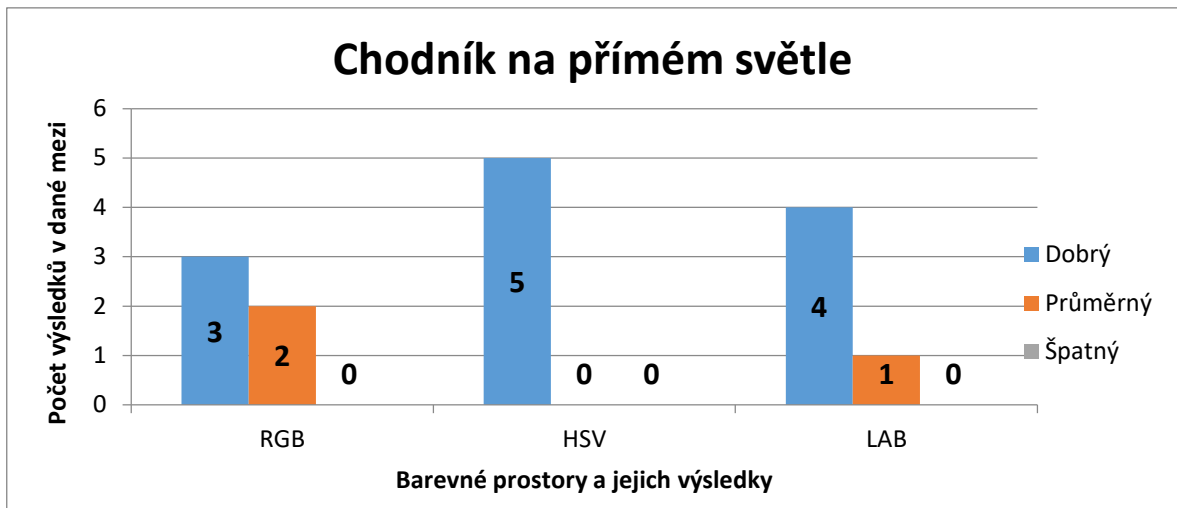
Obrázek 24: Graf porovnání chování jednotlivých barevných prostorů při celé ploše chodníku ve stínu

Následně se provedly testy v místě přechodu mezi stínem a přímým slunečním světlem, kdy stín zasahoval přes celou šíři chodníku. Se samotným místem přechodu neměl žádný barevný prostor problém a chodník ve stínu propojil s chodníkem na přímém světle. Problém nastal u dvou případů barevného prostoru RGB, kdy se nedokázal nastavit zbytek parametrů zpracování obrazu takovým způsobem, aby bylo vyfiltrováno pozadí jak ve stinné části, tak osvětlené sluncem. Vždy jedna z možností zůstala a chodník se spojil s pozadím. Tato situace nastala až při nastavení Gaussova rozmazání na hodnoty 7 a 9.



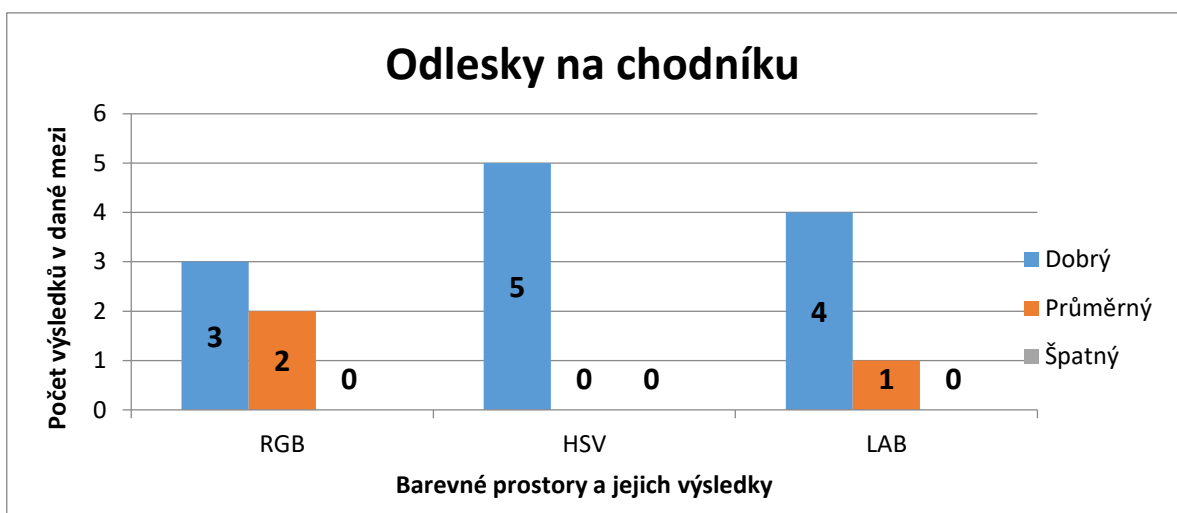
Obrázek 25: Graf porovnání chování jednotlivých barevných prostorů při přechodu mezi stínem a přímým světlem

Při testech, kdy byla celá plocha chodníku i okolí na přímém slunečním světle si nejhůře vedl barevný prostor RGB, kdy jsme cestu správně našli jen při nastavení Gaussova jádra v intervalu hodnot 3 až 7, při moc malém popř. moc velkém rozmazání se chodník začal propojovat s pozadím. O něco lépe si vedl barevný prostor LAB, kdy byl chodník nalezen ve všech případech až do chvíle, kdy se nastavilo moc velké rozmazání, a začal se propojovat s pozadím. Nejlépe si se zpracováním obrazu poradil barevný prostor HSV, který našel cestu vždy. S rostoucí velikostí Gaussova rozmazání se mohl zmenšit interval hodnot prahování, společně se zmenšením účinku morfologického uzavření a byl získán přesnější obraz chodníku.



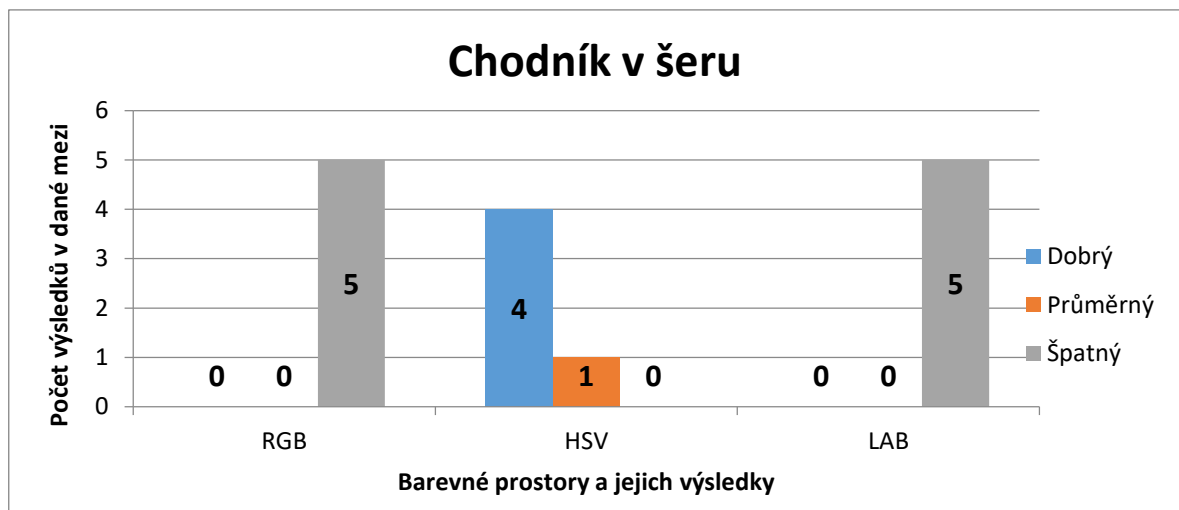
Obrázek 26: Graf porovnání chování jednotlivých barevných prostorů při celé ploše chodníku na přímém světle

Jako další byla zkoumána přítomnost odlesků na chodníku. Kdy nastala stejná situace, jako při přechodu mezi stínem a přímým světlem, kdy si s odlesky poradily všechny barevné prostory, ale zároveň s nastavením Gaussova jádra na hodnotu 7 a výše nastalo propojení chodníku s pozadím.



Obrázek 27: Graf porovnání chování jednotlivých barevných prostorů při přítomnosti odlesků na chodníku

Nakonec byly metody zpracování obrazu a barevné prostory otestovány ve večerních hodinách v šeru, chvíli před úplnou tmou. Barevné prostory RGB a LAB cestu vůbec nenašly a výsledný výstupní obraz byl téměř jen šum s malým náznakem chodníku. Barevný prostor HSV podal velmi dobré výsledky a s nalezením cesty neměl vůbec žádný problém po nastavení Gaussova rozmazání na hodnotu 3 a výše, viz Obrázek 29.



Obrázek 28: Graf porovnání chování jednotlivých barevných prostorů na chodníku večer v šeru

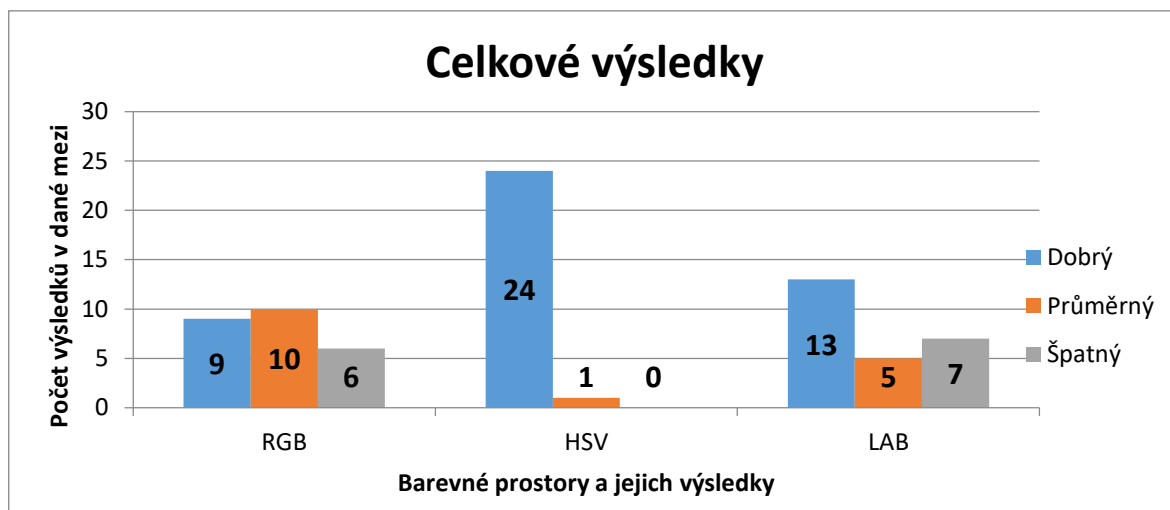


Obrázek 29: Ukázka zpracování obrazu pro zmíněné světelné podmínky, zleva: chodník celý ve stínu, přechod mezi stínem a přímým světlem, přímé světlo, odlesky, šero

Celkově byly provedeny testy při pěti různých světelných podmínkách na jedné kombinaci metod zpracování obrazu a třech barevných prostorech. U metod zpracování obrazu se postupně měnily a nastavovaly jednotlivé parametry ovlivňující kvalitu zpracování obrazu do doby, než byl nalezen nejlepší výsledek.

Nejlépe si se zpracováním obrazu poradil barevný prostor HSV, který bez problémů našel cestu ve 24 případech z 25, pouze při malém osvětlení obrazu bylo třeba aplikovat větší Gaussovo rozmazání a obraz byl následně zpracován správně. Druhý barevný prostor LAB správně našel cestu ve 13 případech z 25, kdy měl problém s nalezením cesty při velkém rozmazání obrazu nebo špatných světelných podmínkách. Nakonec barevný prostor RGB, který dobře fungoval pouze za dobrého osvětlení, kdy byla nalezena cesta v 9 případech z 25, a v ostatních případech nebyly získány dobré výsledky.

Z výsledků Obrázku 30 vyplývá, že nejspolehlivější barevný prostor, který podává nejvíce konzistentní výsledky, je barevný prostor HSV. Dále se ukázalo, že při kombinaci s vyšším Gaussovým rozmazáním (hodnoty 3-7), se může zmenšit interval hodnot při segmentaci obrazu prahováním a z pozadí bude vyčištěn šum. Při sníženém šumu v obrazu se může zmenšit velikost jádra morfologického uzavření a získáme přesnější okraje chodníku. Obecně lze říct, že eliminací šumu v pozadí před samotným zpracováním, lze výrazně zlepšit kvalitu výsledného zpracovaného obrazu.



Obrázek 30: Graf celkových výsledků závislosti kvality zpracování obrazu na barevném prostoru a světelných podmínkách

ZÁVĚR

V této práci byly představeny barevné prostory a metody zpracování obrazu, které byly otestovány v programovém prostředí MATLAB a byly vybrány pouze ty, které splňovaly naše požadavky pro využití v oblasti navigace a řízení pohybu kolového robota. Z původního výběru byl vyřazen pouze barevný prostor CMY, který naše požadavky nesplňoval.

Následně byly barevné prostory a metody zpracování obrazu přeprogramovány do jazyka C++ s využitím open source knihoven openCV, které se zabývají zejména zpracováním obrazu v reálném čase. Zároveň bylo vytvořeno grafické prostředí pro ovládání programu ve vizuálním editoru Qt Creator Design.

Vybrané metody a barevné prostory podstoupily další testy, kde jsme se zabývali případnou možností řízení robota, tedy zpracováním obrazu v reálném čase. Všechny kombinace metod zpracování obrazu a barevných prostorů podstoupily testy, kde se měřila rychlost zpracování obrazu v odpovídajícím časovém úseku. Z výsledků vyplývá, že preprocesingová metoda Mean-shift a segmentační metoda Rostoucích regionů zpracování obrazu v reálném čase nezvládá z důsledku enormního času zpracování jediného snímku, a proto byly tyto metody také vyřazeny.

Nakonec zůstaly tři barevné prostory (RGB, HSV a LAB) a v každé oblasti zpracování obrazu jedna metoda (Gaussovo rozmazání, Metoda prahování a Morfologické operace otevření a uzavření). Na tomto finálním výběru byly provedeny poslední testy, které se zabývaly, jak zvolený barevný prostor ovlivnil kvalitu zpracování obrazu v závislosti na světelných podmínkách. Z výsledků vyplynulo, že nejspolehlivější a nejvíce konzistentní barevný prostor pro zpracování obrazu za účelem navigace je barevný prostor HSV, který si dokáže poradit se šumem v pozadí, odlesky i všemi světelnými podmínkami, které byly v této práci otestovány. Jakožto nejlepší metoda preprocesingu byla vybrána metoda Gaussova rozmazání a v oblasti segmentace obrazu metoda Prahování, společně s využitím postprocessingových metod morfologického otevření a uzavření.

Výsledky této práce lze v budoucnu použít při implementaci samotného řízení robotického systému.

SEZNAM POUŽITÉ LITERATURY

- [1] RGB Color Model [online]. [cit.:2021-11-4]. Dostupné z:
https://favpng.com/png_view/space-rgb-color-model-rgb-color-space-png/c2YBj7e1
- [2] The CMYK colour space [online]. [cit.:2021-23-3]. Dostupné z:
<https://blogs.bl.uk/collectioncare/2013/10/what-the-cmyk-colour-spaces-and-printing.html>
- [3] CMY Color Space [online]. [cit.:2021-11-4]. Dostupné z:
<https://docs.microsoft.com/en-us/windows/win32/wcs/cmy-and-cmyk-color-spaces>
- [4] NIRAPURE, D., UDAYPAL R. *Fast Retrieval of Images Using Filtered HSV Color Level Detection*. International Journal of Emerging Techn-ology and Advanced Engineering, Vol. 3, No. 3, PP. 414-419, 2013.
- [5] Color spaces in OpenCV (C++/Python) [online]. [cit.:2021-23-3]. Dostupné z:
<https://learnopencv.com/color-spaces-in-opencv-cpp-python/>
- [6] CIE-LAB COLOR SPACE [online]. [cit.:2021-23-3]. Dostupné z:
<http://sheriffblathur.blogspot.com/2013/07/cie-lab-color-space.html>
- [7] Diskrétní dvourozměrná konvoluce [online]. [cit.:2021-23-3]. Dostupné z:
<http://cgtucna.blogspot.com/2013/11/diskretni-dvourozmerna-konvoluce.html>
- [8] Understanding convolutions [online]. [cit.:2021-23-3]. Dostupné z:
<http://colah.github.io/posts/2014-07-Understanding-Convolutions/>
- [9] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. Second edition. Pacific Grove: PWS Publishing, [1999], xxiv, 770 s. ISBN 053495393X
- [10] CHENG, YIZONG. *Mean shift, mode seeking, and clustering*. Vol. 17. NO. 8, s. 790-799. [online]. [cit.:2021-23-3]. Dostupné z:
http://home.ku.edu.tr/mehyilmaz/public_html/mean-shift/00400568.pdf
- [11] Image Segmentation using Mean Shift [online]. [cit.:2021-23-3]. Dostupné z:
<https://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained>

- [12] Image Segmentation Using Region Growing and Shrinking [online]. [cit.:2021-27-4]. Dostupné z: <https://slideplayer.com/slide/5279829/>
- [13] Implementing a Connected Component Labeling algorithm from scratch [online]. [cit.:2021-23-3]. Dostupné z: <https://towardsdatascience.com/implementing-a-connected-component-labeling-algorithm-from-scratch-94e1636554f>
- [14] Morfologické operace [online]. [cit.:2021-23-3]. Dostupné z: <http://vision.uamt.feec.vutbr.cz/?course=ZVS&type=ex&page=ex11>
- [15] Types of Morphological Operations [online]. [cit.:2021-23-3]. Dostupné z: <https://ch.mathworks.com/help/images/morphological-dilation-and-erosion.html>
- [16] AXIS 206 Family Network Camera User's Manual. 2nd ed. Axis, [2005] [online]. [cit.:2021-23-3]. Dostupné z: https://netcam.cz/produkty/ip-kamery/pdf/206_um.pdf

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MATLAB	Matrix Laboratory
RGB	Red Green Blue
CMY	Cyan Magenta Yellow
HSV	Hue Saturation Value
LAB	Lightness A B
WIFI	Wireless Fidelity
IEEE	the Institute of Electrical and Electronics Engineers
WPA	WIFI Protected Access
PC	Personal Computer
USB	Universal Serial Bus
IP	Internet Protocol
RTSP	Real Time Streaming Protocol
GHz	GigaHertz
S	Sekunda
FPS	Frames Per Second
Gauss	Gaussovo rozmazání
Ms	Mean-shift
P, px	Pixel

SEZNAM OBRÁZKŮ

Obrázek 1: Barevný prostor RGB [1]	10
Obrázek 2: Barevný prostor CMY [3]	11
Obrázek 3: Barevný prostor HSV [4]	12
Obrázek 4: Barevný prostor LAB [6]	13
Obrázek 5: Na vstupní obraz je aplikováno konvoluční jádro, které upraví výstupní obraz podle nastavených parametrů jádra [8].....	15
Obrázek 6: Porovnání obrázku před a po použití filtru mean-shift [11].....	15
Obrázek 7: Porovnání originálního obrázku s obrázkem, na který je aplikována metoda prahování	16
Obrázek 8: Porovnání originálního obrázku s obrázkem, na který je aplikována metoda rostoucích regionů [12].....	17
Obrázek 9: Ukázka morfologického otevření – vlevo originál a vpravo po použití morfologického otevření [15].....	18
Obrázek 10: Ukázka morfologického uzavření – vlevo originál a vpravo po použití morfologického uzavření [15]	19
Obrázek 11: kamera AXIS 206W [16] a webové prostředí kamery.....	21
Obrázek 12: WIFI router ASUS WL-530GV2	22
Obrázek 13: Testovaný zahradní a pouliční chodník.....	25
Obrázek 14: Levá čtveřice: aplikace Gaussova rozmazání, převodu do HSV a metody rostoucích regionů. Pravá čtveřice: aplikace filtru Mean-shift, převodu do LAB a metody rostoucích regionů	26
Obrázek 15: Výše popsané zpracování: levá čtveřice s prostorem CMY, pravá s prostorem LAB.....	26
Obrázek 16: Graf procentuální úspěšnosti metod zpracování obrazu pro všechny barevné prostory při použití všech kombinací metod zpracování obrazu	27
Obrázek 17: Graf porovnávající procentuální úspěšnost metod zpracování obrazu s použitím barevných prostorů HSV a LAB	28
Obrázek 18: Náhled na celou plochu grafického prostředí aplikace	31
Obrázek 19: Bližší náhled na spodní panel grafického prostředí	32
Obrázek 20: Bližší náhled na pravý panel grafického prostředí	33
Obrázek 21: Graf závislosti kombinace metod zpracování obrazu na čase vykreslení 1 s videa ve 30 fps	35

Obrázek 22: Graf rychlosti vykreslení 1 sekundy videa při různých kvalitách zdroje.....	35
Obrázek 23: Porovnání kvality zpracování zleva: dobrý, průměrný, špatný.....	36
Obrázek 24: Graf porovnání chování jednotlivých barevných prostorů při celé ploše chodníku ve stínu.....	37
Obrázek 25: Graf porovnání chování jednotlivých barevných prostorů při přechodu mezi stínem a přímým světlem.....	38
Obrázek 26: Graf porovnání chování jednotlivých barevných prostorů při celé ploše chodníku na přímém světle.....	39
Obrázek 27: Graf porovnání chování jednotlivých barevných prostorů při přítomnosti odlesků na chodníku.....	39
Obrázek 28: Graf porovnání chování jednotlivých barevných prostorů na chodníku večer v šeru.....	40
Obrázek 29: Ukázka zpracování obrazu pro zmíněné světelné podmínky, zleva: chodník celý ve stínu, přechod mezi stínem a přímým světlem, přímé světlo, odlesky, šero.....	40
Obrázek 30: Graf celkových výsledků závislosti kvality zpracování obrazu na barevném prostoru a světelných podmínkách	41

SEZNAM PŘÍLOH

P I Obsah CD

PŘÍLOHA P I: OBSAH CD

/text – text práce

/code – zdrojové kódy

/data – výsledky MATLAB testů, měření zpracování v reálném čase v C++, ukázkové video chodu aplikace