



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

DETEKCE MOBILNÍHO ROBOTU ZPRACOVÁNÍM OBRAZU

MOBILE ROBOT DETECTION USING IMAGE PROCESSING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. STANISLAV NOVOTNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ KREJSA, Ph.D.

BRNO 2012

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky

Akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Stanislav Novotný

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Mechatronika (3906T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Detekce mobilního robotu zpracováním obrazu

v anglickém jazyce:

Mobile robot detection using image processing

Stručná charakteristika problematiky úkolu:

Při ověřování lokalizačních systémů mobilních robotů je pro určení kvality lokalizace nutné znát skutečnou polohu robotu. Úkolem je vypracovat metodu automatické detekce polohy robotu v obrazovém prostoru s využitím obrazů snímaných pomocí CCD snímačů v rovině rovnoběžné s rovinou, ve které se robot pohybuje.

Cíle diplomové práce:

1. Navrhněte metodu automatické detekce polohy robotu v obraze
2. Implementujte metodu pro obrazy v běžných formátech, včetně RAW formátu
3. Implementujte software pro dávkové zpracování sekvence obrazů
4. Vyhodnoťte kvalitu detekce na záznamu pohybu reálného robotu.

Seznam odborné literatury:

J.C.Russ: The Image Processing Handbook, CRC Press, 3rd edition

Vedoucí diplomové práce: Ing. Jiří Krejsa, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 18.10.2011

L.S.

prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

Abstrakt

Tato diplomová práce se zabývá zpracováním sekvence snímků robota ze statické kamery umístěné nad rovinou pohybu robota. Nejprve jsou popsány metody segmentace obrazu a metody vhodné pro nalezení polohy. V další části jsou vybrané metody implementovány a porovnány na jednotlivých snímcích. Nakonec jsou vybrané metody dále implementovány v algoritmu pro dávkové zpracování sekvence snímků.

Klíčová slova

zpracování obrazu, segmentace obrazu, prahování, lokalizace objektu

Abstract

This master's thesis deals with processing of image sequence taken by statically placed camera over plane of robot movement. At first there are methods for image segmentation and localization methods described. In the next part, selected methods are implemented and compared to individual images. In the final part, selected methods are further implemented in algorithm for batch processing of image sequence.

Keywords

image processing, image segmentation, thresholding, object localization

Bibliografická citace

NOVOTNÝ, S. *Detekce mobilního robotu zpracováním obrazu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 52 s. Vedoucí diplomové práce Ing. Jiří Krejsa, Ph.D..

Prohlášení

Já, Stanislav Novotný, prohlašuji, že jsem diplomovou práci vypracoval samostatně pod vedením svého vedoucího práce Ing. Jiřho Krejso, Ph.D.. a že jsem uvedl všechny použité prameny a literaturu.

V Brně dne 23.5. 2012

podpis:

Poděkování

Na tomto místě chci poděkovat své rodině za podporu po celou dobu mého studia. Dále chci poděkovat za ochotu a dobré vedení svému vedoucímu práce, Ing. Jirímu Krejsovi, Ph.D. Také děkuji každému, kdo mě morálně podporoval při tvorbě této práce.

Obsah

1. Úvod.....	9
2. Formulace problému a cíle řešení.....	11
3. Analýza problému.....	13
3.1. Charakteristika prostředí.....	13
3.2. Předpoklady řešení.....	15
4. Zpracování obrazu.....	16
4.1. Datové typy digitálního obrazu.....	16
4.1.1. Rastrový obraz.....	16
4.2. Formát RAW.....	17
4.3. Histogram.....	18
4.4. Předzpracování obrazu.....	19
4.4.1. Odstranění šumu.....	20
4.5. Metody segmentace.....	21
4.5.1. Segmentace prahováním.....	21
4.5.2. Metody hledání prahu.....	22
4.5.2.1. Určení prahu experimentálně.....	22
4.5.2.2. Určení prahu z histogramu.....	22
4.5.2.3. Poměrné prahování.....	22
4.5.2.4. Určení prahu hledáním lokálního minima mezi dvěma maximy.....	23
4.5.2.5. Metoda podle Otsu.....	24
4.5.2.6. Adaptivní prahování.....	25
4.5.3. Region-based segmentace.....	25
4.6. Rozlišení segmentovaných oblastí.....	26
4.6.1. Pojmenování skupin pixelů (Connected components labeling).....	26
4.6.2. Pojmenování oblastí pomocí trasování kontur.....	27
4.7. Nalezení středu značky.....	29
5. Implementace vybraných metod.....	30
5.1. Popis programu.....	30
5.2. Preprocessing.....	33
5.3. Zpracování samostatného snímku.....	33
5.3.1. Snímek z fotoaparátu Canon.....	33
5.3.2. Snímek z kamery Pixelink.....	37
5.4. Zpracování sekvence snímků.....	39
5.4.1. Zpracování sekvence z fotoaparátu Canon.....	41
5.4.2. Zpracování sekvence z fotoaparátu Canon se sníženým rozlišením.....	43
5.4.3. Zpracování sekvence z kamery Pixelink.....	44
6. Závěr.....	47
Seznam použité literatury.....	49
Seznam ilustrací.....	51
Seznam tabulek.....	51
Obsah přílohy.....	52

1. Úvod

Počítačové vidění (Computer Vision) je speciální odvětví výpočetní techniky, které je v současné době na obrovském vzestupu díky stále levnějším a výkonnějším kamerovým systémům a snadné dostupnosti potřebného výpočetního výkonu. Praktická aplikace technologií počítačového vidění je stále častější v nejrůznějších odvětvích průmyslu, ale stále častěji i v domácnostech.

Nejčastější použití počítačového vidění jsou součástí řídicích systémů (např. u průmyslových robotů), detekce událostí (např. sledování bezpečnostními kamerami, počítání lidí), pro uspořádání informací (např. indexování obrazových databází), vývoj software pro detekci a rozpoznávání obličejů, registračních značek automobilů a sledování pohybů a technologických procesů ve výrobě.

Lze se tak dnes již relativně běžně setkat například s tím, že notebooky, případně i chytré mobilní telefony, rozeznají svého uživatele na základě identifikace jeho obličeje. Tímto způsobem pak může fungovat například zabezpečení daného zařízení nebo přihlášení se za příslušný uživatelský účet, a tak podobně.

Vedle identifikace lidského obličeje se však počítačové vidění využívá i v jiných způsobech tzv. biometrické identifikace nebo identifikace osoby na základě unikátních parametrů jejího těla. Lze hovořit například o identifikaci sítnice, snímání otisku prstu a dalších.

Další z častých aplikací počítačového vidění je automatické navigování a orientace strojů v reálném prostředí. Jak už to s novými technologiemi bývá, v počátcích našlo počítačové vidění uplatnění především v hi-tech hračkách (např. slavný robotický pes Aibo), avšak s postupem času dochází k aplikaci této technologie v užitečných přístrojích i v domácnosti. Namátkou lze jmenovat například relativně novou technologii robotických vysavačů, kde dražší modely již počítačové vidění využívají k mapování svého okolí.

Do budoucna lze jednoznačně očekávat rozšíření technologie identifikace tváře a dalších praktických aplikací technologie počítačového vidění i do dalších oblastí lidského života.

Lidské vidění na rozdíl od počítačového vidění dovoluje člověku vnímat a porozumět světu, který je obklopuje. Počítačové vidění usiluje o technická řešení, i když se někdy inspiruje v biologickém vidění. Biologické vidění zvládá úlohy, na něž je počítačové vidění stále krátké. Přesto poskytuje biologické vidění inspiraci i pro technická řešení. Stále však napodobování biologického vidění nemusí být nejlepším příkladem řešení technický úlohy.

Stále snadnější dostupnost různých knihoven a materiálů zaměřujících se na zpracování obrazu umožňuje vývojářům po celém světě snadnější přístup do celé problematiky počítačového vidění. V současné době jsou volně dostupné knihovny s funkcemi počítačového

vidění i strojového rozpoznávání předmětů.

Ve své práci řeším problematiku sledování pohybu robota v rovině za pomoci snímkování kamer umístěných nad rovinou pohybu. Za tímto účelem jsou na povrchu robota umístěny dva bílé terče pro snazší detekci polohy robota. Vyvinutý systém by měl dokázat zpracovat běžné obrazové formáty včetně formátu RAW.

2. Formulace problému a cíle řešení

Cílem této práce je splnit požadavek na lokalizaci robota pomocí záběrů jeho pohybu po horizontální rovině při snímání digitálním snímačem umístěným nad rovinou pohybu.

Pro takovýto typ úlohy existuje celá řada algoritmů určených pro detekci pohybujících se objektů pomocí analýzy digitálního obrazu z kamery. V první části této práce je nutné prozkoumat jaké algoritmy lze využít a jaké jsou jejich vlastnosti z hlediska robustnosti a výpočetní náročnosti. V tomto případě se jedná o nalezení dvou bílých značek nalepených na robota (Obrázek 1).



Obrázek 1: Pozorovaný robot

V tomto případě tedy nemůžeme využít přístupu s vyhledáváním specifické barvy, protože bílou barvu značek nelze oddělit od pozadí pouze na základě nalezení barevného odstínu. Dále mohou detekci stěžovat malé bílé plochy jako jsou například odlesky od kovových ploch nebo přsvícené oblasti. Výhodu však je, že používáme statickou kameru. Proto lze před-

pokládat, že nebude docházet k rychlým změnám osvětlení scény. K natočení pohybu robota byly použity dvě různé kamery:

Fotoaparát Canon EOS 350D DIGITAL

CMOS snímač v tomto fotoaparátu je formátu APS-C s rozměry 22,2 x 14,8 mm s rozlišením 8 megapixelů tedy 3456x2304 pixelů, crop faktor ohniska je 1,6. Byl použit objektiv Canon EF 20 s ohniskovou vzdáleností 20 mm a světelností 2,8, (přepočtená ohnisková vzdálenost je 32). Při snímání obrazu bylo použito sekvenční snímání s časovou prodlevou 1s.

a Pixelink PL - B762G

je vybavena snímačem CMOS s rozlišením 752 x 480 pixelů. Použitý objektiv je Edmund Optics SY110M s ohniskovou vzdáleností 1,67 mm a světelností 1,8. Kamera je řízena přes Ethernet. Maximální rychlost snímání v tomto rozlišení je 60 snímků za vteřinu, s ohledem na aplikaci byla použita rychlost 500ms mezi jednotlivými snímky.

3. Analýza problému

V této části popisují prostředí, ve kterém se robot pohybuje, a vlastnosti kamer, které jsou pro tuto úlohu použity.

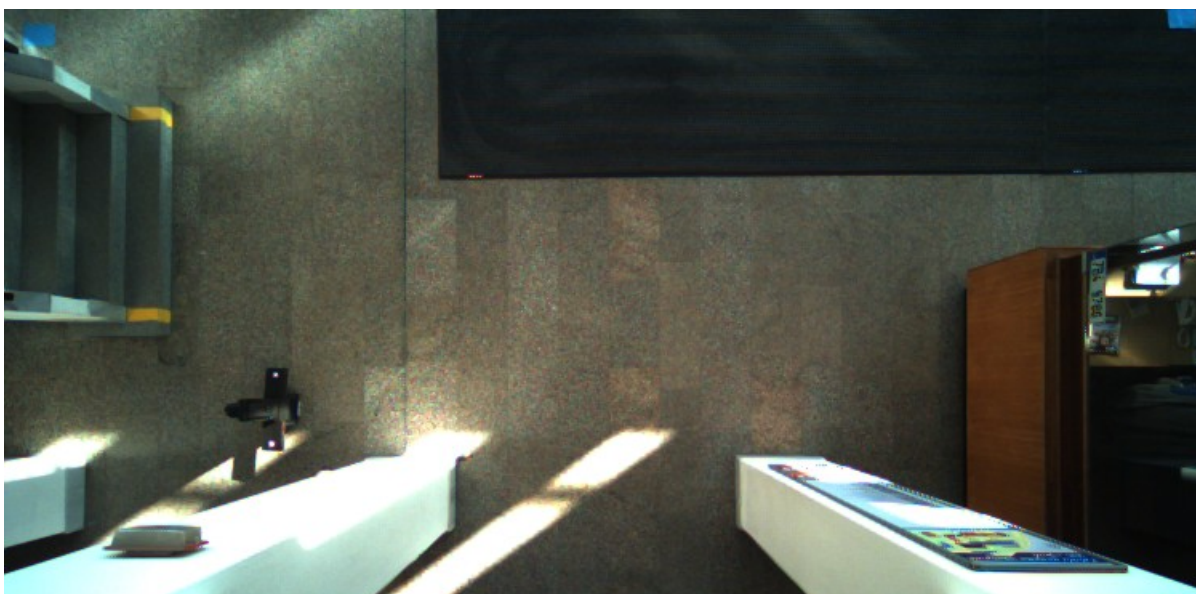
3.1. Charakteristika prostředí

Prostředí z hlediska zpracování pomocí počítačového vidění lze rozdělit podle různých charakteristik. V zásadě se užívá rozdělení na prostředí uvnitř budov, neboli indoor, a venkovní, neboli outdoor, prostředí.

Obecně platí, že indoor prostředí je na zpracování obrazu méně náročné, jelikož terén zde bývá výrazně méně členitý, podlahy mají zpravidla jednu barvu, případně jednu opakující se texturu, kterou je snadné pomocí počítačového vidění identifikovat a odlišit tak podlahu od jednotlivých objektů.

Naproti tomu outdoor prostředí je náročnější, jelikož terén bývá členitější, podloží obsahuje mnoho unikátních textur a drobných objektů (kamínky, spadané listí, a podobně). I z tohoto důvodu jsem se ve své práci zaměřil na počítačové vidění a rozpoznávání především v indoor prostředí.

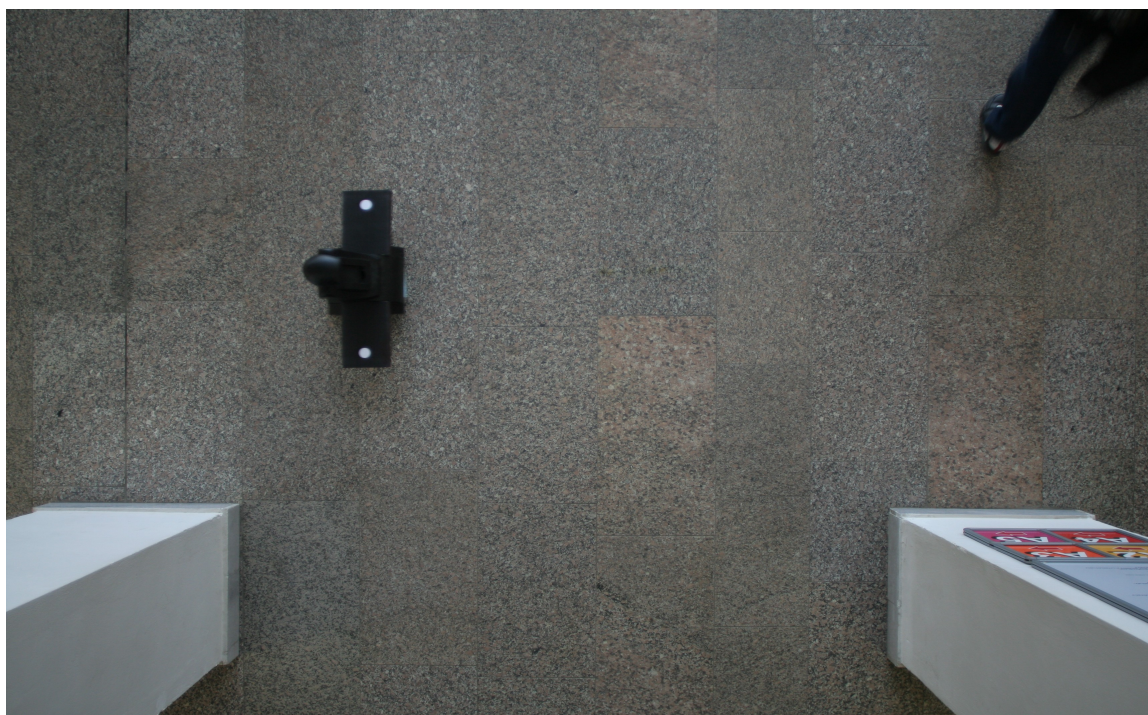
Další komplikací rozpoznávání obrazu ve venkovním prostředí je skutečnost, že se tam častěji objevuje ostrý přechod mezi světlem a stínem, zatímco ve vnitřních prostorách je obecně rovnoměrnější osvětlení.



Obrázek 2: Ukázka záběru pozorované scény

Záběry jsou pořízeny ve vnitřních prostorách budovy. Podlaha je jednodolitá vodorovná plocha s jemnou texturou bez výrazných změn v intenzitě. Tento typ podlahy je pro tuto úlohu velice vhodný neboť robot je neustále dostatečně kontrastní. Největší problém způsobují přsvícené oblasti od slunce, které svítí okny a vytváří tak ostré přechody mezi světlem a stínem.

Vhodnější by samozřejmě bylo laboratorní prostředí s jednodolitou podlahou a rovnoměrným světlem, nicméně je nutno uvažovat rovněž o tom, že pro případnou praktickou aplikaci této technologie je nezbytné, aby si dokázala poradit i s náročnějšími podmínkami. Jistá míra tolerance k specifikám daného prostředí je proto nezbytná.



Obrázek 3: Záběr vyfocený kamerou Canon

Na snímcích z kamery Pixelink PL - B762G je záběr poměrně širokouhlý, takže zabírá velkou plochu podlahy. Nevýhodou je malé rozlišení, které v kombinaci se širokouhlým objektivem znamená, že identifikační značky jsou velmi malé. Také vzhledem k nerovnoměrnému osvětlení dochází při průjezdu robota osvětlenou plochou k výraznému kontrastu mezi oběma značkami, což dále ztěžuje identifikaci robota. Příklad takového nerovnoměrného osvětlení je na obrázku 2.

Na záběru z fotoaparátu Canon 350D je vidět značně menší pozorovací úhel daný použitým typem objektivu a jeho ohniskovou vzdáleností. Zároveň má kamera vyšší rozlišení a značky jsou tak větší a lépe detekovatelné. Zároveň jsou v tomto případě značně lepší světelné podmínky, neboť do místnosti nevniká ostré slunce, které by způsobovalo nerovnoměrnost osvětlení a vysoký kontrast mezi značkami, jako tomu bylo v předchozím případě.

3.2. Předpoklady řešení

Pro správné řešení musí být algoritmus schopen rozlišit identifikační značky od okolního prostředí. Nejvhodnější metodou, pomocí které toho lze dosáhnout, je metoda segmentace prahováním. Aby bylo možné rozlišit identifikační značky na zpracovávaném snímku musí být segmentací oddělen od pozadí černý povrch robota. Pokud není mezi robotem a pozadím dostatečný kontrast, je možné prahováním odlišit celé pozadí snímku tak, aby v obraze zbyly pouze identifikační značky. V takové případě je nutné, aby obě značky byly dostatečně kontrastní vůči zbytku světlých částí obrazu.

4. Zpracování obrazu

V této části se zabývám základními metodami a funkcemi potřebnými pro zpracování obrazu. Nejprve popíšu základní datové typy digitálního obrazu. Poté se zabývám histogramem a konvolucí, z hlediska předzpracování obrazu. Následně se zabývám metodami pro segmentaci a nalezení hodnoty prahu pro prahování. Následně zpracováním segmentovaného obrazu.

4.1. Datové typy digitálního obrazu

4.1.1. Rastrový obraz

Jedná se o nejběžněji používaný způsob ukládání obrazové informace. Obraz tvořený dvourozměrným polem, kde v každém bodě je uložena informace o barvě pomocí některého z barevných modelů (například RGB). Mezi běžně používané rastrové obrazy patří mimo jiné soubory typu: .png, .jpg, .bmp

Soubor typu PNG – Portable Network Graphics (česky přenosná síťová grafika)

„Tento grafický formát má mnoho vlastností, které ho předurčují pro použití v pre-pressu, zpracování fotografií, prezentaci obrázků na webu atd. Mezi nejvýznamnější vlastnosti patří volitelná bitová hloubka (ta určuje maximální počet barev), ukládání průhlednosti pixelů (alfa kanál), prokládání pixelů umožňující rychlé náhledy na obrázek, podpora barevných profilů apod.“ [1]

PNG používá bezztrátovou kompresi. Obraz může být uložen v odstínech šedi, kde se nejčastěji využívá osmibitová barevná hloubka. Nebo v barevném modelu RGB, kde se nejčastěji používá barevná hloubka 24bitů, ale podporuje i 48bitovou hloubku. [1]

Soubor typu JPEG - Joint Photographic Experts Group

Jedná se o nejčastěji používaný formát pro ukládání fotografií. Může používat ztrátovou i bezztrátovou kompresi. Při použití ztrátové komprese lze dosáhnout u určitého typu velmi vysokého kompresního poměru bez výrazné ztráty kvality. Nejvhodnější použití je právě u fotografií, kde se vyskytují větší plochy stejné barvy. Není vhodný pro obraz s menší barevnou hloubkou, nebo mnoha barevnými přechody, jako je například text.

Soubor typu BMP – Bitová mapa

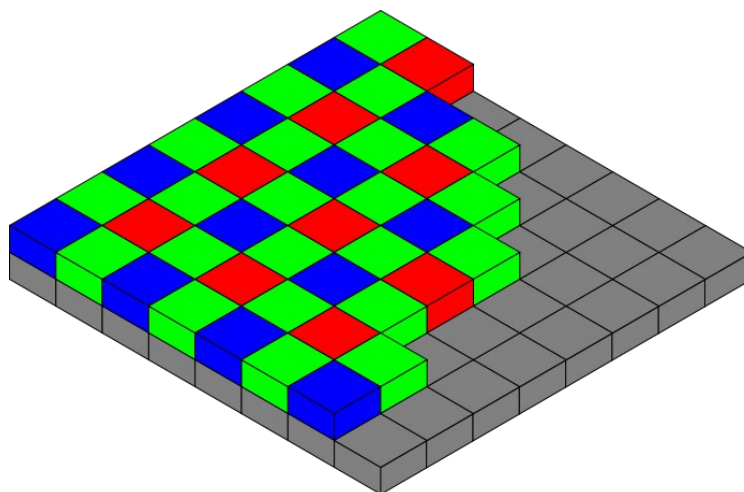
Tento typ se nejčastěji používá bez komprese. Požívá 4 různé barevné hloubky. 1Bitová, 4bitová, a 8bitová barevná hloubka používá k definici barvy barevnou paletu. V této barevné paletě jsou definovány barvy v modelu RGB. 24Bitová hloubka paletu nepoužívá, protože v každém pixelu je přímo hodnota barvy v RGB (uložena je však v opačném pořadí, tedy BGR).

4.2. Formát RAW

V podstatě se také jedná o rastrový obraz, reprezentuje však surová nezpracovaná data z obrazového snímače (odtud název raw – česky surový). Každý výrobce obrazového snímače používá svůj vlastní typ raw, z toho důvodu je nevhodný pro běžné použití. Soubory typu raw mají podle výrobce různé koncovky, například společnost Canon používá .CRW nebo .CR2, Nikon používá .NEF a Olympus .ORF. Následující text se týká především formátu .CR2

V souboru typu Raw je mimo vlastní obrazová data uloženo také mnoho dalších informací jako jsou například informace o modelu fotoaparátu, expoziční informace (expoziční čas, clonové číslo, nastavení citlivosti, a podobně), nastavení fotoaparátu, vyvážení bílé a datum a čas pořízení snímku.[2]

Vlastní obrazová data jsou uložena bez barvené informace. Každý pixel obsahuje pouze informaci o intenzitě světla, které zaznamenal během expozice. Barevná informace je dána použitou barevnou maskou na snímači. Nejčastěji je použita Bayerova maska (obrázek 4)



Obrázek 5: Bayerova maska

Výslednou barevnou informaci pro každý pixel je nutné získat interpolací z okolních pixelů. Metody skládání obrazu z matice barevného filtru (demosaicing) mohou využívat řadu různých postupů, nejčastěji je však používán sekvenční postup. Kdy je nejprve spočítána složka luminance pomocí které je následně dopočítána chrominance červené a modré barevné složky. Obraz je tak vytvořen v barevném modelu YCrCb.[3]

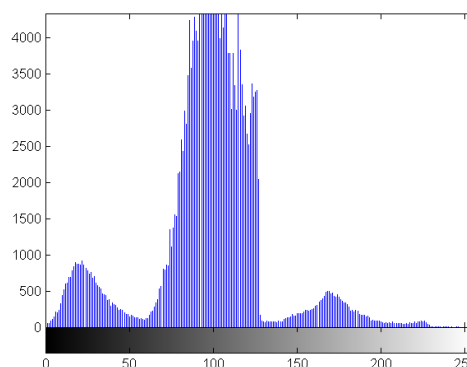
4.3. Histogram

„Histogram slouží k hodnocení rozložení intenzit v digitálním obraze. Z grafické podoby histogramu lze usuzovat nejen na stav digitálního obrazu, ale také na možnosti úpravy obrazu.“ [4]

Histogram je statistické zpracování rozdělení intenzity v obraze, obvykle graficky znázorněné jako sloupcový graf. Při tvorbě histogramu je nutné nejprve rozdělit vstupní data do tříd podle intenzity jednotlivých bodů, následně je spočítána četnost výskytu bodů náležících do každé třídy. V případě histogramu obrazu v odstínech šedi je samostatnou třídou každá hodnota odstínu v obraze. Histogram je tak znázornění četnosti výskytu každého odstínu šedi v obraze.

$$h(I_k) = n_k \quad (1)$$

Kde I_k je hodnota k -tého odstínu a n_k je počet bodů mající tento odstín. Součet všech hodnot četností je roven celkovému počtu obrazových bodů.[4]



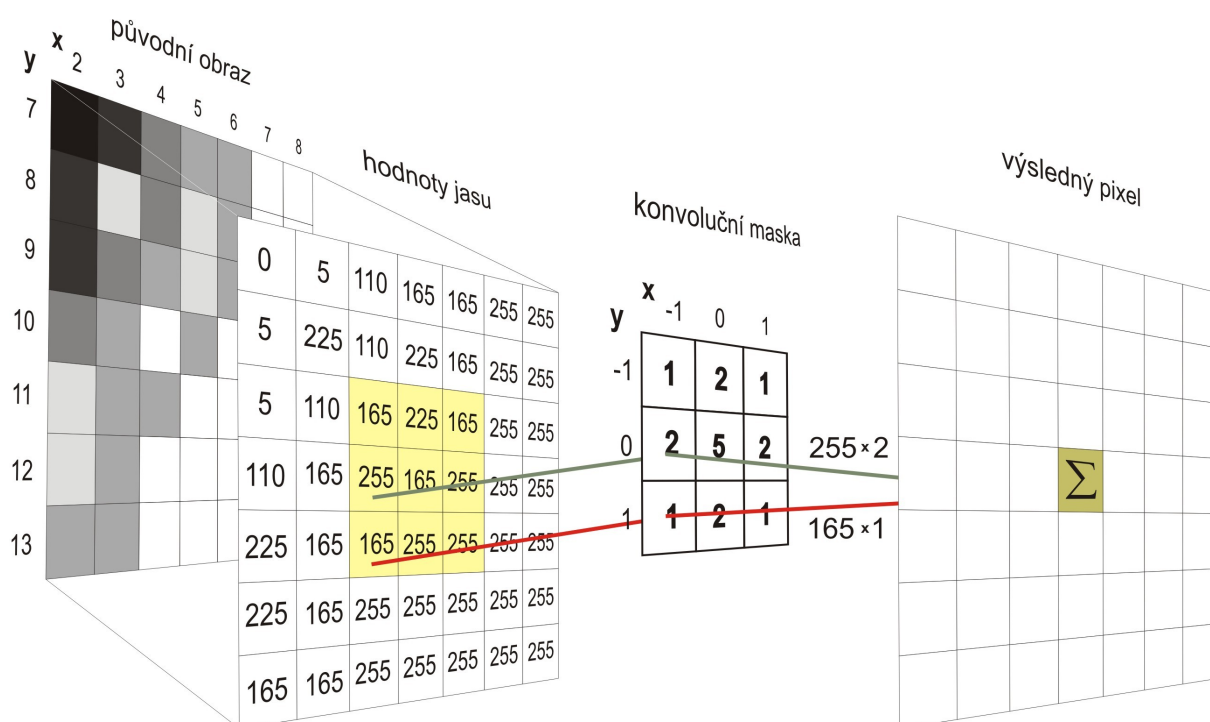
Obrázek 6: Histogram obrazu v odstínech šedi

4.4. Předzpracování obrazu.

Pro předzpracování obrazu se obvykle využívá dvourozměrné konvoluce. Jedná se o zpracování dvou funkcí jejich vzájemnou integrací. Konvoluce v zpracovávaném obraze se využívá k aplikaci celé řady efektů, kde jednou z funkcí pro konvoluci je upravovaný obraz a druhou je konvoluční jádro které udává aplikovaný efekt. Výsledný obraz je tak dán rovnicí pro diskretní dvourozměrnou konvoluci (2). [4][5]

$$(f * g)[m, n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) g(m-i, n-j) \quad (2)$$

Pomocí různých konvolučních jader lze dosáhnout celé řady obrazových efektů, jako jsou vyhlazení (filtr typu dolní propust), doostření (horní propust), nebo detekce hran. [6]



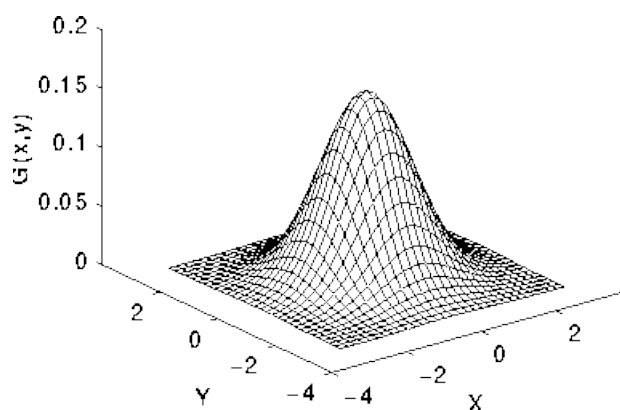
Obrázek 7: grafické znázornění funkce diskretní 2D konvoluce

4.4.1. Odstranění šumu

Nejjednodušší metoda odstranění šumu je filtrace pomocí lineárního filtru typu dolní propust. Například lze přiřadit každému pixelu průměrnou hodnotu všech pixelů v určitém okolí. K tomu lze využít dvourozměrné konvoluce, kde jádro má stejné hodnoty v celém rozměru. Například pro průměrování z oblasti o rozměrech 3x3 je použito jádro v rovnici (3)

$$g = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3)$$

Tento filtr výrazně snižuje ostrost obrázku. Rozostření lze snížit použitím jádra které dává větší váhu na právě upravovaný bod, nebo na jeho bližší okolí. Pro takové jádro může být použito Gaussovo rozložení (obrázek 8). Míra rozmazání je tak nižší, stále však dochází k rozmazání obrazu a tím jsou znehodnoceny hrany.



Obrázek 8: Gaussovo rozložení s průměrem 0 a rozptylem 1

Také lze pro odstranění šumu použít nelineární filtry.

„Nelineární filtry nepočítají intenzitu upravovaného bodu, ale vybírají z okolí vhodnou hodnotu, kterou pak dosazují do upravovaného bodu. Oproti lineárním filtrům mají tu výhodu, že nepřidávají do obrazu žádnou novou hodnotu intenzity.“ [4]

Takovým filtrem je například Medián. Tento filtr vybere z okolí hodnotu, která leží uprostřed posloupnosti po seřazení všech hodnot z okolí podle velikosti.

Nevýhodou je, že narušuje ostré rohy a tenké čáry a tím odstraňuje detaily [6]

4.5. Metody segmentace.

Segmentace obrazu je základním krokem zpracování obrazu pro počítačové vidění. Jejím cílem je rozdělit obraz na nepřekrývající se oblasti, které odpovídají pozorovaným objektům. Hlavním cílem je rozlišit objekt zájmu od pozadí. Dalším důležitým účelem segmentace je redukce zpracovávaných dat.

„Cíl segmentace:

- rozčlenit obraz do částí, které souvisí s předměty či oblastmi reálného světa = oddělení objektů od pozadí, analýza obsahu obrazu
- obraz chystáme pro další krok = popis
- redukce dat, zjednodušení“[7].

Existuje řada metod segmentace obrazu, mezi ně patří:

Segmentace prahováním.

Segmentace pomocí detekce hran (edge-based).

Metoda růstu regionů (region-based).

4.5.1. Segmentace prahováním

Segmentace prahováním je metoda převodu obrazu na binární informaci, kde každý bod obrazu udává pouze informaci, zda do objektu patří nebo nepatří. Tedy zda hodnota jeho odstínu překročila nebo nepřekročila daný práh. U barevných obrazů je možné nastavit prahy pro jednotlivé barevné složky. Nový binární obraz $g(x,y)$ má v bodě o souřadnicích x,y hodnotu 0 nebo 1 podle toho zda hodnota intenzity vstupního obrazu $f(x,y)$ na stejných souřadnicích překročila hodnotu prahu p (4).

$$g(x, y) = \begin{cases} 1, & f(x, y) > p \\ 0, & f(x, y) \leq p \end{cases} \quad (4)$$

Částečné prahování:

V případě částečného prahování nezískáme binární obraz, ale pouze označíme za černé určité pixely, které nedosahují dané hodnoty intenzity. Ostatní obrazové body zůstanou nezměněny. Toto je vhodné využít pro detekci středů oblastí pomocí metody váženého těžiště.

Obdobně je možné provést komplexnější prahování, pokud zavedeme více úrovní (prahování s více prahy). V podstatě tak lze získat obraz v odstínech šedi, kde počet odstínů odpovídá počtu prahů.

Hlavním problémem, se kterým se setkáme pokud chceme využít metodu segmentace prahováním, je volba vhodného prahu. Existuje celá řada algoritmů určených k nalezení optimální hodnoty prahu. Žádná z nich však nefunguje univerzálně a každá je vhodná pro jiný druh úlohy. Některé metody jsou popsány v následující kapitole

4.5.2. Metody hledání prahu

4.5.2.1. Určení prahu experimentálně.

Práh nalezneme ručně tak aby, segmentovaný obraz ideálně odpovídal potřebám dalšího zpracování.

Lze využít v případě, že nedochází ke změně světelných podmínek a lze tak použít jednou nalezený práh pro větší soubor dat. Případně lze využít experimentálně nalezený práh jako výchozí bod pro hledání prahu v snímcích, kde dochází k menším změnám světelných podmínek. Případně musí být metoda dalšího zpracování dostatečně robustní vůči takovýmto změnám.

4.5.2.2. Určení prahu z histogramu

Existuje řada metod prahování obrazu s využitím histogramu. Většinou se jedná o iterativní metody, které jsou nejvhodnější pokud chceme oddělit objekt od pozadí. Obvykle předpokládají že rozdělení odstínu šedi je v případě objektu dostatečně odlišné od hodnot odstínu šedi na pozadí. [8]

4.5.2.3. Poměrné prahování.

Jedná se o nejjednodušší neiterativní metodu, která určí práh pokud známe jakou část obrazu zabírá objekt který chceme oddělit od pozadí. Jednoduše nastavíme práh tak aby oddělil takový zlomek obrazu jaký chceme. Práh získáme z kumulativního histogramu (5).

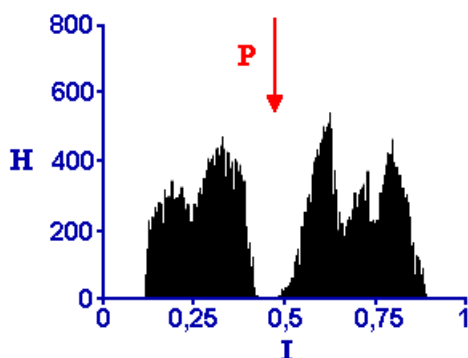
$$C(g) = \sum p(g) \quad (5)$$

Kde $p(g)$ je histogram obrázku. Velikost hledaného objektu v daném výřezu je známa jako $\frac{1}{q}$ pak práh T získáme jako $c(T) = \frac{1}{q}$ nebo $c(T) = 1 - \frac{1}{q}$ v případě že hledáme tmavý

objekt na světlém pozadí. V mém případě je nutné oddělit bílé značky nalepené na černém robotu od pozadí. Proto v podstatě hledám tmavý objekt na světlém pozadí.

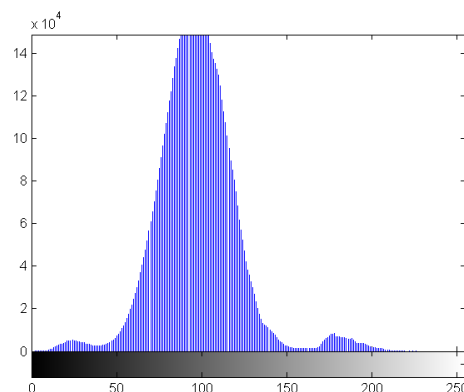
4.5.2.4. **Určení prahu hledáním lokálního minima mezi dvěma maximy.**

Tato metoda je vhodná především pokud jsou v obraze oblasti se dvěma rozdílnými oblastmi intenzity, má tedy bi-modální histogram. Je možné oddělit v tomto obraze prahem tyto podobné úrovně intenzity. V takovém případě se určí prahová hodnota jako střed vzdálenosti mezi těmito špičkami, nebo jako lokální minimum. Viz obrázek 9.[6, 8, 9]



Obrázek 9: Určení prahu z histogramu

Tato metoda není vhodná pro zpracování sekvence z kamery Canon kde jsou ideální světelné podmínky, protože na těchto snímcích není možné z histogramu rozlišit křivky odpovídající robotu od pozadí.



Obrázek 10: Histogram snímku z fotoaparátu Canon

4.5.2.5. Metoda podle Otsu

V této metodě se na obraz díváme jako na dvě skupiny bodů s různým rozsahem hodnot intenzity. problémem je, že tyto rozsahy intenzit se obvykle překrývají. Snahou metody je minimalizovat chybnou identifikaci pixelu pozadí jako popředí a naopak.[10]

Tato metoda se snaží dosáhnout optimálního prahu tím, že snižuje společný rozptyl obou skupin bodů.

$$\sigma_b^2(T) = n_b(T) n_o(T) [\mu_b(T) - \mu_o(T)]^2 \quad (6)$$

$$n_b(T) = \sum_{i=0}^{T-1} p(i) \quad (7)$$

$$n_o(T) = \sum_{i=T}^{N-1} p(i) \quad (8)$$

$$\mu_b(T) = \sum_{i=0}^{T-1} \frac{ip(i)}{n_b(t)} \quad (9)$$

$$\mu_o(T) = \sum_{i=T}^{N-1} \frac{ip(i)}{n_o(t)} \quad (10)$$

Kde $\sigma_b^2(T)$ je vzájemný rozptyl obou skupin bodů, T je aktuálně zkoumaná hodnota prahu a $[0, N - 1]$ je rozsah úrovní intenzity. $p(i)$ je počet pixelů o dané intenzitě

Pro každou uvažovanou hladinu prahu:

- Rozdělíme pixely do skupin podle hodnoty prahu.
- Spočteme průměr v každé skupině.
- Rozdíl průměrů vynásobíme na druhou.
- Vynásobíme počtem pixelu v každé skupině.

Hledaný práh je takový který má maximální hodnotu $\sigma_b^2(T)$

Tento postup je výpočetně náročný, můžeme však využít histogramu a podle toho, jak se postupně zvedá práh a přesouvají se pixely z jedné skupiny do druhé, lze pouze upravovat n a μ podle následujícího postupu k nalezení nových hodnot.

$$n_b(T+1) = n_b(T) + n_T \quad (11)$$

$$n_o(T+1) = n_b(T) - n_T \quad (12)$$

$$\mu_b(T+1) = \frac{\mu_b(T)n_b(T) + n_T T}{n_b(T+1)} \quad (13)$$

$$\mu_o(T+1) = \frac{\mu_o(T)n_o(T) + n_T T}{n_o(T+1)} \quad (14)$$

Kde n_T je četnost pixelů o intenzitě rovné prahu.[10, 11]

4.5.2.6. *Adaptivní prahování.*

Práh je funkcí polohy bodu v obraze. Existuje řada metod, které se snaží nalézt optimální práh v závislosti na vlastnostech bodů v daném okolí aktuálně upravovaného pixelu. Tyto metody jsou však značně výpočetně náročné.

Rozdělení obrazu na oblasti – artefakty na hranici oblastí

překrývající se hranice, interpolace mezi oblastmi odstraní artefakty

Tyto metody jsou vhodné především pro scény s nerovnoměrným osvětlením.

[7]

4.5.3. **Region-based segmentace**

Segmentace růstem regionů. Vytváří regiony postupně od předem definovaných bodů (seeds). V každém kroku jsou vyhledány všechny body sousedící s daným regionem. Pro každý sousední bod se určí, zda do daného regionu patří podle zadaných kritérií homogenity. Po přidání bodu se upraví celkové vlastnosti regionu a algoritmus pokračuje další iterací. Pokud právě přidáný bod již byl součástí některého regionu, pak jsou oba regiony sloučeny.

V případě, kdy pixel nevyhoví kritériu homogenity a tudíž nepatří do daného regionu, můžeme k tomuto bodu přistupovat dvěma způsoby. První možností je považovat tento bod za semeno nového regionu, tím získáme úplnou segmentaci celého obrazu kde každý bod bude patřit do regionu podobných bodů. Druhou možností je nechat tento bod za neoznačený a pokračovat v růstu předchozího regionu. Takto získáme částečnou segmentaci obrazu kde maximální počet regionů je shodný s počtem počátečních semen.

Této metody lze s výhodou využít pokud předem víme, kde se hledaný objekt nachází. A můžeme tedy této informace využít pro umístění počátečních bodů.[8]

4.6. Rozlišení segmentovaných oblastí

Pro nalezení polohy robota je nutné v segmentovaném obraze vybrat z možných světlých oblastí ty, které odpovídají světlým kroužkům tvořícím identifikační značky. Proto je nutné určit, které pixely patří ke stejným objektům. K tomu se využívají metody nazývané jako „connected components labeling“, což lze přeložit jako označování spojitých oblastí. Níže rozvádím dvě nejčastěji používané metody založené na connected components labeling.

4.6.1. Pojmenování skupin pixelů (Connected components labeling)

Algoritmus prohledávání obrazu za účelem nalezení a pojmenování skupin pixelů s podobnou hodnotou intenzity. Prohledávání probíhá pixel po pixelu a přiřazuje pixelu hodnotu podle sousedních pixelů. Toto lze nejlépe aplikovat na binární obraz

Sousedství pixelů může být definováno různými způsoby.

Čtyřokolí (z anglického 4-connectivity)

Jedná se o soubor čtyř sousedních pixelu. Jejich souřadnice vzhledem k výchozímu pixelu o souřadnicích (x,y) jsou:

$$N_4 = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\} \quad (15)$$

Osmiokolí (z anglického 8-connectivity)

Soubor osmi sousedních pixelů, tedy soubor 4 sousedních pixelů rozšířený o 4 pixely ležící diagonálně. Jejich souřadnice jsou:

$$N_8 = N_4 \cup \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\} \quad (16)$$

Při spojování pixelů do skupin se postupuje podle následujícího algoritmu (předpokládáme 8 sousedních bodů).

Procházíme obraz zleva do prava po řádcích odshora dolů, dokud nenarazíme na pixel který se má pojmenovat (v binárním obraze má hodnotu $V=1$). Poté se podíváme na sousední pixely, které již byly prohledány, tedy pixel nalevo od současné polohy a tři pixely o řádek výš. Mohou nastat následující možnosti:

- 1) Žádný pixel nemá hodnotu příslušnosti $V=1$, přiřadíme p nový název.
- 2) Jeden soused má hodnotu $V=1$, přiřadíme p název tohoto suseda.
- 3) Více sousedů má hodnotu $V=1$, Přiřadíme p název jednoho ze susedů a zavedeme ekvivalenci mezi názvy ostatních.

Poté sloučíme ekvivalentní názvy do tříd ekvivalence a každé třídě přiřadíme jednotný název. [6]

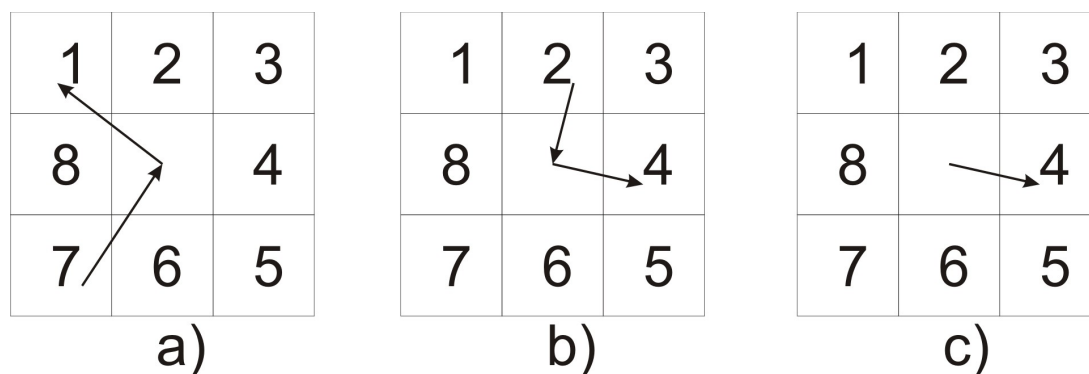
Tato metoda nemusí pracovat efektivně u větších obrazů, protože musí projít celou oblast dvakrát a tím se prodlužuje výpočtový čas.

4.6.2. Pojmenování oblastí pomocí trasování kontur.

Rychlejší metoda pro component labeling využívá detekce obrysu oblastí v binárním obraze a následně označí pixely uvnitř. Podle [12] je výpočtově rychlejší než předchozí metoda, protože algoritmus nalezne spojené oblasti a přiřadí jim jméno během jediného průchodu obrazem, přičemž obrysové body navštíví maximálně čtyřikrát. Další výhodou je současné nalezení obrysu.

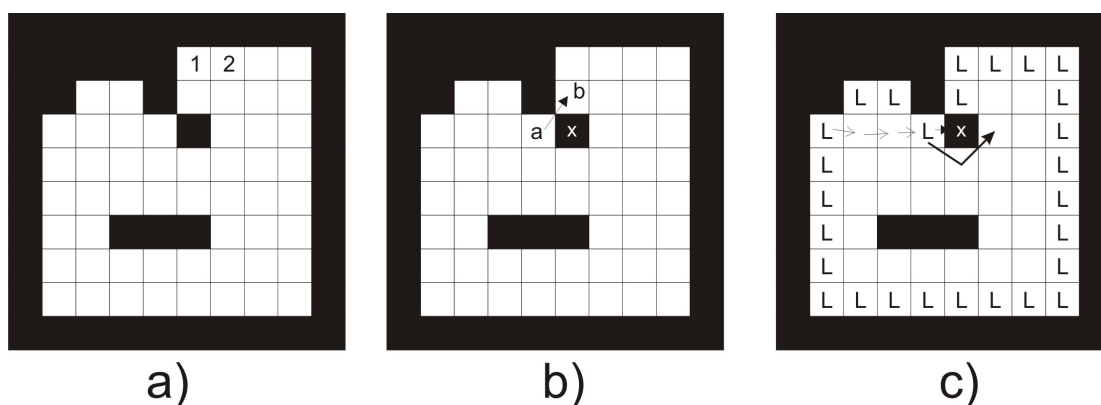
Algoritmus prochází obraz zleva doprava, po řádcích odshora dolů. Pokud narazí na nepojmenovaný nezamaskovaný bod, zkontroluje nejprve zda je již pojmenován levý sousední bod. Pokud ano, přidělí danému bodu stejný název. Pokud je levý sousední bod černý, znamená to, že se bod nachází v nové oblasti. Je proto bodu přiděleno nové jméno a spustí se funkce na nalezení obrysu. Tato funkce hledá mezi osmi sousedními body takový nečerný bod, který je první možný následný bod obrysu podle následujících pravidel:

- Body jsou prozkoumány po směru hodinových ručiček (obrázek 11).
- První bod k prozkoumání je určen podle předcházejícího bodu.
- Pokud se jedná o první bod nového obrysu, je prvním možným bodem obrysu bod napravo (obrázek 11,c).
- Pokud se jedná o první bod vnitřního obrysu, první možný bod je vpravo dole (na obrázku pozice 5)



Obrázek 11: Vyhledávání následujícího pixelu

Pokud například předcházející pixel leží na pozici 7 vzhledem k aktuálnímu pixelu, je třeba začít na pixelu na pozici 1 (obrázek 11,a). Pokud leží předcházející pixel na pozici 2, začneme hledat další pixel na pozici 4. Obecně lze říci, že pozice na které začneme se nachází o 2 pozice dál vůči předcházejícímu pixelu.



Obrázek 12: Příklad labelingu

Na obrázku 12 je příklad oblasti která je označována. Poprvé je oblast nalezena na pixelu označeném na obrázku 2,a číslem 1. Poté prochází funkce označení hranice postupně po směru hodinových ručiček. Když dojde k bodu označenému na obrázku 12,b písmenem a je za následující bod označen bod b. Bod označen písmenem x je tak z obrysu vynechán, okolí tohoto bodu je přiřazeno k té samé oblasti během následného procházení vnitřních bodů. Celá oblast je tak označena stejným názvem během jednoho průchodu obrazem. [12]

Z takto nalezených oblastí je nutné vybrat oblasti, které odpovídají lokalizačním značkám. Pro tyto značky jsou základními parametry jejich velikost a vzájemné vzdálenosti. Tyto parametry jsou prvotní limitací pro nalezení kandidátů. Při prvním pokusu o nalezení robota je

pravděpodobné, že se v celém záběru nalezne více dvojic které tyto požadavky splňují. Nalezení správných bodů pouze z jednoho obrazu by vyžadovalo algoritmus využívající komplexní znalosti o obraze. Proto je provedeno prohledání následujícího obrazu se stejnými parametry a předpokládáme, že hledaná dvojice bodů bude jediná která změní svou polohu.

V následujících oblastech je vždy prohledávána oblast úzkého okolí pozice robota z předcházejícího obrazu. Tím je výrazně zvýšena výpočetní rychlost jednoho obrázku a zároveň je omezena možnost že se v této oblasti objeví matoucí objekt.

4.7. Nalezení středu značky.

Nalezení středu značky se provádí pomocí váženého těžiště na černobílém obraze po provedení částečného prahování. Jedná se o metodu se subpixelovou přesností, jejíž přesnost může být výrazně ovlivněna šumem. Středové souřadnice se určí podle vzorce (2)

$$c(x, y) = \left(\frac{\sum_{x,y \in \omega} x \cdot w(x, y)}{\sum_{x,y \in \omega} w(x, y)}, \frac{\sum_{x,y \in \omega} y \cdot w(x, y)}{\sum_{x,y \in \omega} w(x, y)} \right) \quad (17)$$

Kde ω je daná oblast a $w(x,y)$ váha pixelu na daných souřadnicích x a y , v tomto případě světlost pixelu. Podle [13] prahováním zvyšujeme rozptyl určení středu, ale zároveň snižuje ustálenou odchylku. Poloha robota je pak dána jako bod ležící uprostřed na spojnici značek tzn.:

$$C(x, y) = \left[x_2 + \frac{x_1 - x_2}{2}, y_2 + \frac{y_1 - y_2}{2} \right] \quad (18)$$

5. Implementace vybraných metod

V této části nejprve popíšu vytvořený program a jeho hlavní části, dále pak porovnávám funkčnost zpracování jednotlivých snímků.

Pro zpracování daného problému jsem se rozhodl využít segmentace prahováním, kde pro nalezení hodnoty prahu využívám poměrné prahování, případně iterativní metodu, kde postupně zvyšuji hodnotu prahu, dokud nenaleznu první dva body vyhovující požadavkům.

Segmentovaný obraz dále popíšu pomocí metody pojmenování spojitých oblastí pomocí trasování kontur. Z takto nalezených oblastí identifikuji značky na základě známých vlastností. Hledanými vlastnostmi jsou počet pixelů, vzdálenost a kompaktnost oblasti.

U nalezených oblastí dále vypočítám střed pomocí váženého těžiště.

5.1. *Popis programu.*

Program na zpracování záběrů z kamery byl zpracován v programu Microsoft Visual Studio 2010, pomocí programovacího jazyka C#. V úvodní obrazovce lze vybrat samostatný soubor pro zpracování a vyzkoušení různých parametrů. Po kliknutí na tlačítko „Zpracovat složku“ se zobrazí okno pro dávkové zpracování sekvence snímků.

Program jsem testoval na zpracování obrázků ve formátech typu .jpg, .png, .bmp, a .gif. Přidat podporu pro jiný běžně používaný formát je triviální záležitost. Pro zpracování obrazu z formátu RAW fotoaparátu Canon (typ souboru .CR2) je nutné mít nainstalované kodeky z Microsoft Camera Codec Pack a použít mezi reference PresentationCore.dll. Protože tato knihovna využívá funkce z System.Windows.Media.Imaging je nutné nejprve převést obraz do proměnné typu System.Drawing.Bitmap. Toho je dosaženo pomocí funkce CopyPixels. Stejným způsobem by mělo být možné zpracovat RAW formát také z jiných kamer nebo fotoaparátů, které jsou podporovány výše uvedeným balíčkem kodeků. Nemám však k dispozici snímky z jiného fotoaparát než Canon, proto není tento způsob převodu otestován na Raw formát od jiného výrobce.

Program má dvě hlavní okna. Jejich popis je na obrázcích 12 a 13. V prvním okně je možné otestovat funkčnost nastavení různých parametrů na jednotlivých snímcích. Také lze v tomto snímku ručně určit hodnotu prahu a vyzkoušet jaké hodnoty prahu určí dvě použité metody hledání prahu.

V druhém okně lze zpracovávat sekvence snímků pomocí předem nastavených parametrů.

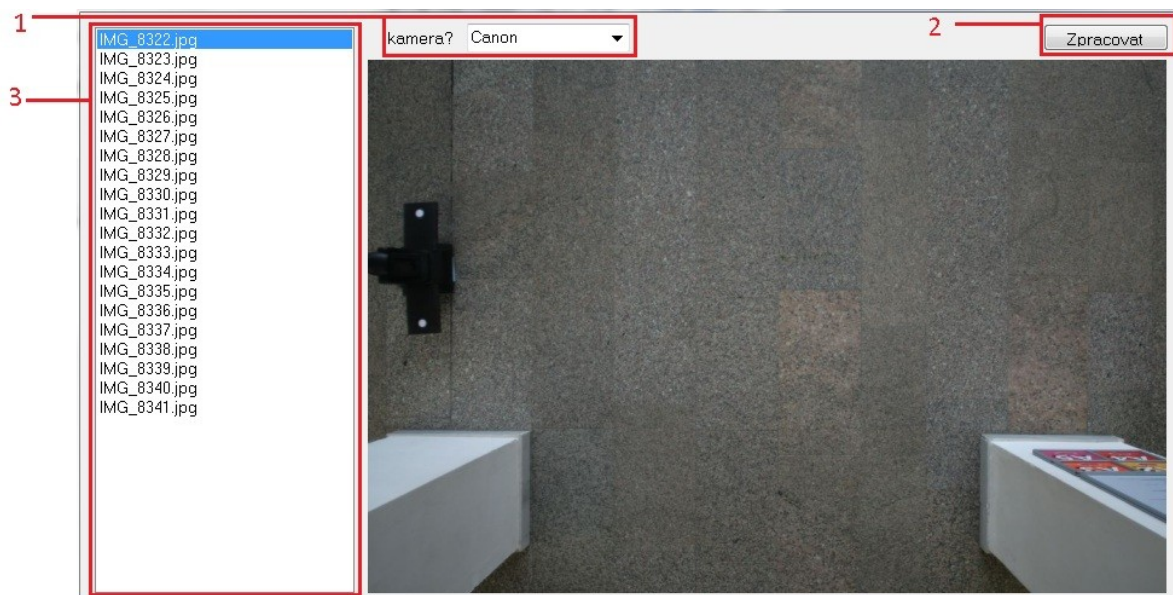


Obrázek 13: Obrázovka 1 - zpracování samostatného obrázku

Popis prvků na obrázku 13:

- 1) Tlačítka na zobrazení jednotlivých barevných kanálů a odstínů šedi.
- 2) Výběr oblasti zájmu (ROI – region of interest).
- 3) Zadání minimálního (nahore) a maximálního (dole) počtu pixelů v hledaných oblastech.
- 4) Zadání minimální (nahore) a maximální (dole) vzdálenosti hledaných oblastí.
- 5) Zadání zlomku pro poměrné prahování. Zadáním čísla 20 nalezne práh pro 1/20 nejtmavších pixelů.
- 6) Zadání hodnoty prahu v rozsahu 0-255.
- 7) Nalezení prahu Otsuho metodou.
- 8) Nalezení prahu metodou poměrného prahování.
- 9) Zpracování a nalezení bodů podle nastavených parametrů.
- 10) Aplikace konvoluce Gaussovým jádrem 3x3 pixely.

Pomocí okna na obrázku 13 je možné nalézt parametry, které jsou nejvhodnější pro daný snímek případně v jakém rozsahu parametrů ještě získáme dostatečně dobré výsledky nalezení hledaných značek. Středů všech nalezených oblastí jsou po zpracování zobrazeny ve snímku červenými křížky



Obrázek 14: Obrazovka 2 - zpracování sekvence snímků

Popis prvků na obrázku 14:

- 1) Výběr typu kamery a parametrů zpracování
- 2) Zpracování celé sekvence snímků
- 3) Seznam snímku ke zpracování

Po vybrání složky určené ke zpracování je vytvořen seznam všech obrázků v dané složce. Následně je nutné vybrat ze seznamu v prvku 1 o jaký typ snímků se jedná. Po dokončení zpracování sekvence je automaticky vygenerován textový soubor s nalezenými body. Soubor má název „vystup.txt“ a je uložen ve veřejné složce s absolutní adresou „C:\Users\Public“. Z tohoto je možné vytvořit tabulku použitím symbolu „;“ jako oddělovače. V prvním sloupci tabulky je zapsáno o kolikátý snímek sekvence se jedná, v 3 a 4 sloupci pak souřadnice bodu odpovídajícího středu spojnice prvních dvou nalezených bodů. V dalších sloupcích jsou souřadnice všech nalezených bodů. Dvojice souřadnic je vždy oddělena prázdným sloupcem.

5.2. Preprocessing

Protože nemůžeme využít barevných vlastností, je výhodné nejprve obrázky převést do odstínů šedi. V závislosti na množství šumu v obraze je nutné provést vyhlazování pomocí dvourozměrné konvoluce s Gaussovým jádrem o rozměrech 3x3.

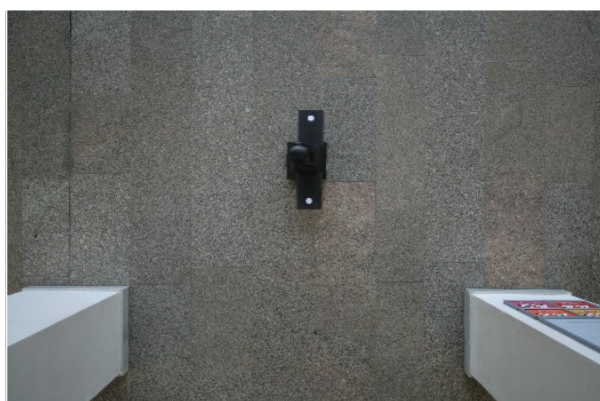
5.3. Zpracování samostatného snímku

Zpracování bitmap, pomocí základních funkcí obsažených v `System.Drawing`, jako je například funkce `GetPixel`, je poměrně pomalé a náročné na paměť. Proto je vhodné uzamknout obrázek do paměti a pracovat s ním pomocí ukazatelů (pointerů). K tomuto slouží `unsafe` metody. Jejich použitím je výrazně snížena časová složitost zpracování obrazu.

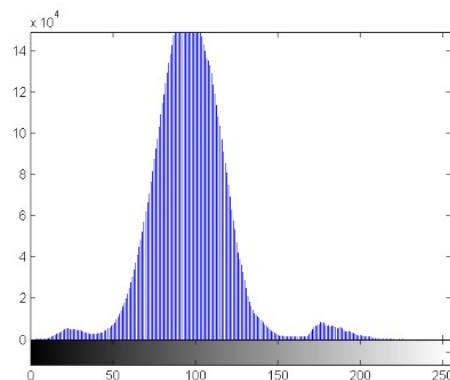
5.3.1. Snímek z fotoaparátu Canon

V případě kamery Canon je vhodné využít toho, že na záběru je černý povrch robotu nejtmavším objektem v záběru. Proto je vhodné použít poměrné prahování. Zatímco nalezení prahu pomocí Otsuho metody zanechá v obraze značné množství oblastí na okraji světlé oblasti uprostřed snímku, poměrné prahování odstraní pouze robota a oddělí tak bílé značky (viz Obrázek 15). Další možností, která by umožnila spolehlivé nalezení identifikačních značek, by bylo naopak dosažení dodatečně vysokého prahu tak, aby byla ze snímku úplně odstraněna textura podlahy. Identifikační značky by tak zůstaly společně s bílými sloupy jedinými neprahovanými objekty v obraze. Tato úroveň prahu se mi však nejevila tak vhodná, neboť se zvyšujícím se prahem klesá počet nenaprahovaných pixelů v identifikačních značkách. Také jsou tyto značky výrazně menší než ostatní světlé objekty v obraze.

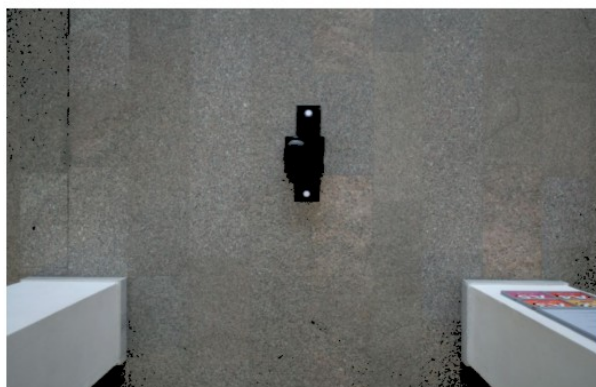
Zpracování snímku z fotoaparátu Canon v plném rozlišení zabere nejdelší čas v porovnání se záběry z kamery Pixelink. Doba potřebná ke zpracování snímku je nejvýrazněji závislá na počtu zpracovávaných obrazových bodů. Snímky z fotoaparátu Canon mají přibližně dvacetinásobný počet pixelů.



a) nezpracovaný snímek



b) histogram snímku



c) prahování pomocí poměrného prahování



d) prahování pomocí Otsuho metody

Obrázek 15: Zpracování snímku z fotoaparátu Canon.

Takto jsou na obrazu značky jediné oblasti, které vyhovují parametrům vyhledávacího algoritmu, a nemůže dojít k chybné identifikaci polohy.

Zpracování snímku z fotoaparátu s vysokým rozlišením je značně pomalé a neefektivní. Obě identifikační značky mají přibližně 1000 obrazových bodů. I malé rozdíly v osvětlení způsobí poměrně velký rozdíl mezi počtem pixelů v každé značce. Tím se zvyšuje pravděpodobnost výskytu oblastí, které mohou být chybně identifikované. Z tohoto důvodu jsem snížil rozlišení snímků na 1037x691 pixelů. Takto upravené fotky je možné rychleji zpracovat. Také u fotek s nižším rozlišením byly značky správně identifikované ve větším rozsahu prahů.

Při použití poměrného prahování s nastavenou hodnotou $1/40$ nejtmašších pixelů je nalezena hodnota prahu 52. Parametry nastavené pro hledání značek jsou v tabulce 1

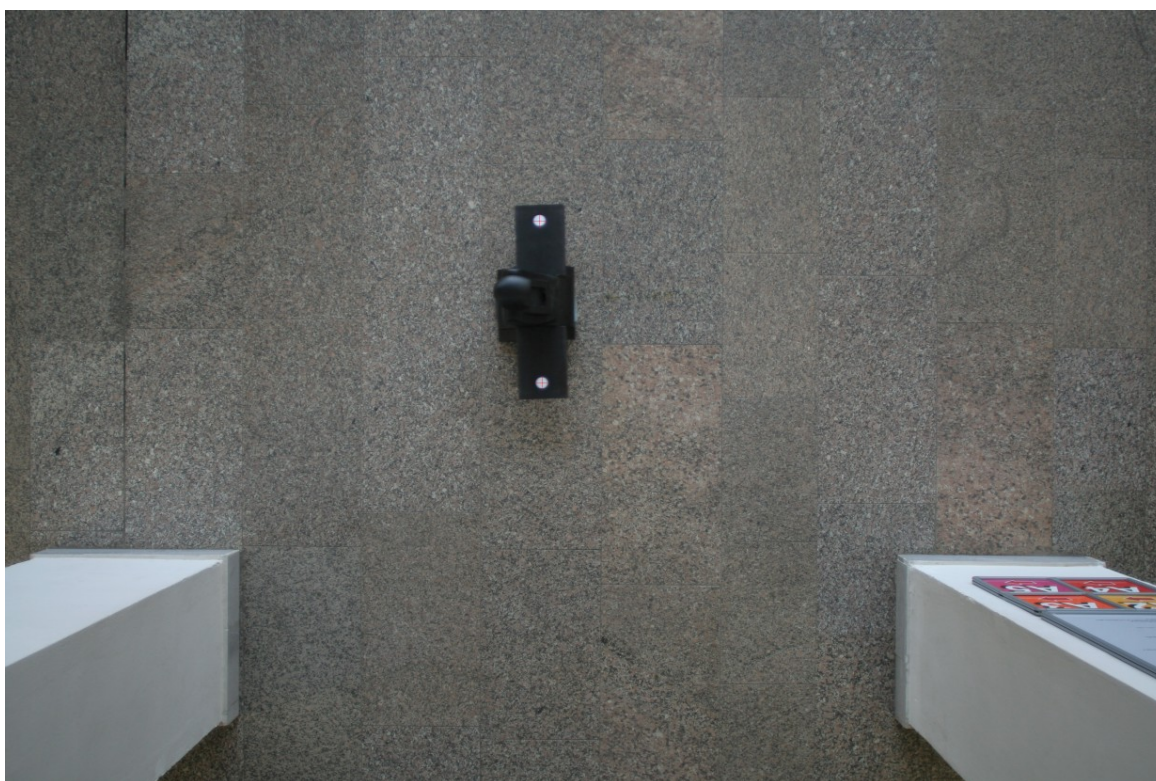
Parametr	Minimální hodnota	Maximální hodnota
Počet pixelů značky	800	3000
Vzdálenost bodů	400	500
Poměr pro prahování	1/40	
Hodnota prahu	52	
Velikost oblasti zájmu (ROI)	3456	2304

Tabulka 1: Parametry hledání

Rozsah hodnot použitých pro zpracování je poměrně velký, přesto jsou nalezeny pouze dvě správné oblasti. Identifikace je tak v tomto případě dostatečně robustní. Počet pixelů v nalezených značkách je 1457 a 1977. Výsledek zpracovaného obrazu s vyznačenými nalezenými body je na obrázku 16. Poloha nalezených bodů v obrazovém prostoru je v tabulce 2

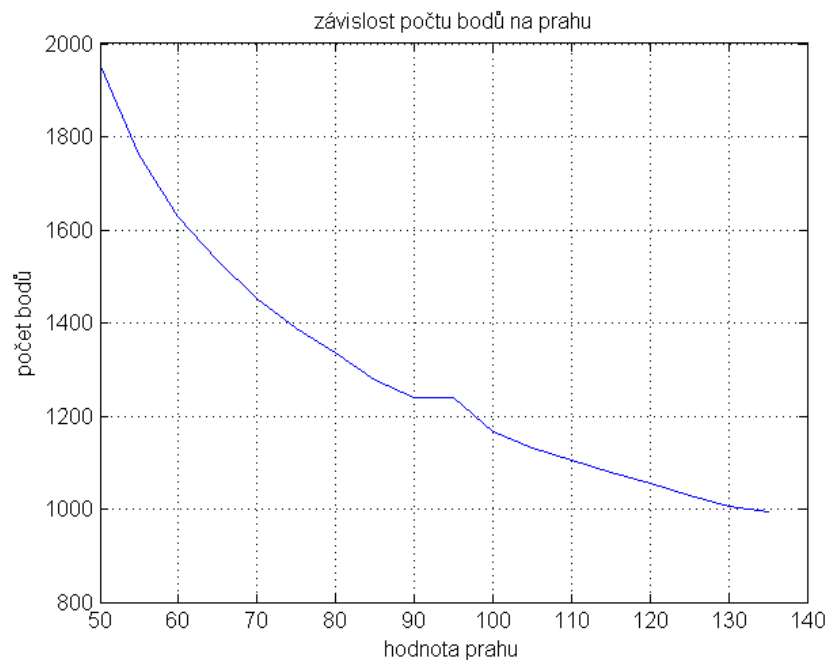
	x	y	Počet pixelů
Bod 1	1599,668	1106,648	1457
Bod 2	1592,59	625,895	1977

Tabulka 2: Zpracování snímku z fotoaparátu Canon



Obrázek 16: Zpracovaný snímek z fotoaparátu Canon

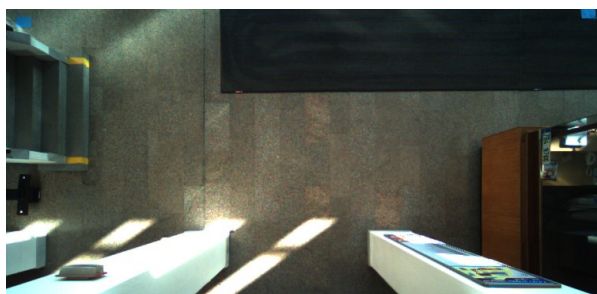
Počet pixelů je výrazně závislý na hodnotě prahu (viz obrázek 17) z toho důvodu je nutné zvolit poměrně velký rozsah počtu pixelů pro zpracování prvního snímku sekvence. To však způsobuje zvýšené riziko chybných identifikací.



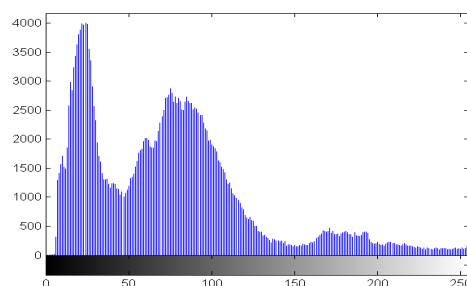
Obrázek 17: závislost počtu pixelů značky na hodnotě prahu

5.3.2. Snímek z kamery Pixelink

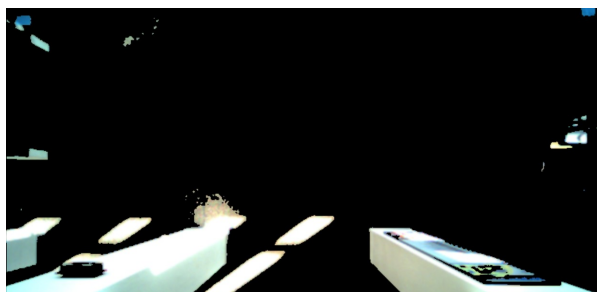
V případě snímků z kamery Pixelink je situace složitější a ani jedna z předchozích metod prahování nezaručí dobrý výsledek. Tyto snímky nemají dobré světelné vlastnosti a vyskytují se zde ostré přechody mezi světlem a stínem. V tomto případě je vhodné použít postupné iterativní prahování. Zde vycházíme z předpokladu, že značky na robotu budou první oblasti, které odpovídají zadaným parametrům. Toto však platí pouze v případě, kdy jsou obě značky přibližně stejně osvětleny.



a) původní snímek



b) histogram snímku



c) prahování otsuho metodou

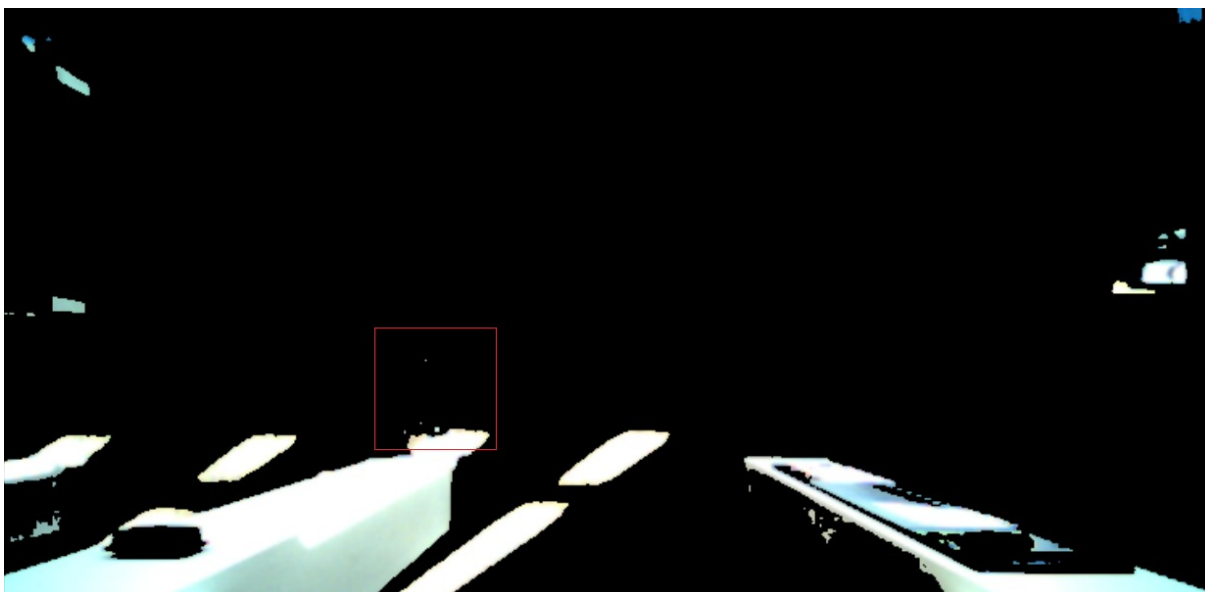


d) poměrné prahování

Obrázek 18: Prahování snímku z kamery Pixelink

Na obrázku 18 vlevo je zobrazeno prahování pomocí Otsuho metody. Z tohoto obrazu je patrné, že tato metoda je pro tento typ snímku zcela nevhodná, neboť zcela odstraní jednu z identifikačních značek. Poměrné prahování je na obrázku vpravo. Tento výsledek je již přijatelný, avšak stále se na obraze nachází velké množství matoucích míst na okraji stínů.

V sekvenci snímků z kamery Pixelink se nachází několik snímků, pro které je automatické nalezení značek pomocí použitých metod téměř nemožné.



Obrázek 19: Snímek průjezdu robota ostrým světlem po prahování

Na obrázku 19 je zobrazeno minimální prahování, při kterém je oddělena značka nacházející se v ostrém slunci od pozadí. Poloha robota je zvýrazněna červeným čtvercem. Jedná se o hodnotu prahu 169. Při této hodnotě prahu je druhá značka velká pouze jeden pixel.

V případě, kdy se robot pohybuje v oblasti s rovnoměrným osvětlením jsou již podmínky lepší. Pro zpracování takového snímku je nastaveno poměrné prahování, velikost značek a další parametry podle hodnot v tabulce 3.

Parametr	Minimální hodnota	Maximální hodnota
Počet pixelů značky	15	35
Vzdálenost bodů	42	50
Poměr pro prahování	1/2	
Hodnota prahu	53	
Velikost oblasti zájmu (ROI)	752	368

Tabulka 3: parametry pro zpracování snímku z kamery pixelink.

Jedná se o poměrně velký rozsah parametrů velikosti a vzdálenosti bodů, přesto jsou v daném snímku nalezeny pouze dva body. Tento výsledek je však velmi citlivý na správnou volbu prahu. V tomto případě je nutné odstranit celou polovinu nejtmačších pixelů ve snímku, aby nevznikly artefakty umožňující chybnou interpretaci v povrchu černého koberce v horní

části snímku. Zároveň však nesmí být hodnota prahu příliš vysoká, jinak začnou vznikat matoucí oblasti na povrchu podlahy.

	x	y	Počet pixelů
Bod 1	476,65	196,75	22
Bod 2	450,41	162,72	25

Tabulka 4: nalezené body ve snímku "image1165.bmp"

Přestože dochází v průběhu sekvence pouze k menším změnám v osvětlení scény, nemusí se vždy nalézt ideální práh pomocí poměrného prahování. Nalezení jiného prahu také může způsobit velké změny v počtu pixelů značek mezi jednotlivými snímky. Což značně stěžuje jejich nalezení pomocí jednotných parametrů pro všechny snímky.



Obrázek 20: Zpracovaný snímek z kamery Pixelink

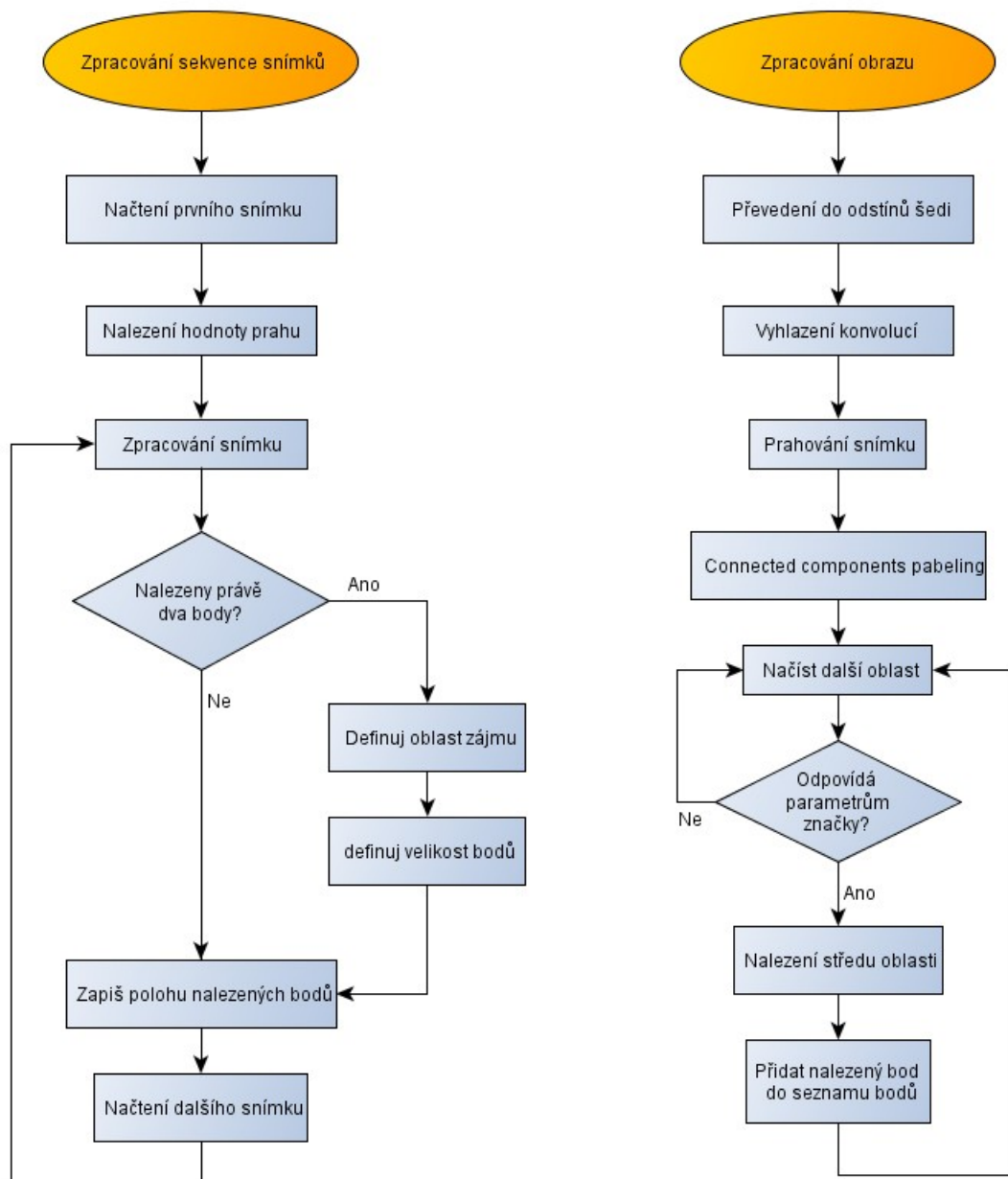
5.4. Zpracování sekvence snímků

Pro nalezení polohy bodů je nejprve prohledán celý snímek podle předem daných parametrů. Pokud jsou nalezeny pouze 2 body, definujeme polohu robota uprostřed spojnice těchto bodů, a kolem jeho polohy definujeme oblast zájmu pro zpracování následujícího snímku.

V dalším snímku ze sekvence tak prohledáváme pouze malou část obrazového prostoru. Tím je velmi snížena výpočetní doba. Schéma zpracování sekvence je na obrázku 21

V případě prvního snímku je za prohledávanou oblast označen celý snímek. V kroku nalezení polohy robota se situace komplikuje, pokud je nalezeno více oblastí vyhovujících podmínek. V takovém případě je nutné porovnat stejnou oblast na jiném snímku a vyřadit oblasti, které odpovídají nehybným předmětům. Jiný algoritmus by mohl určit správnou dvojici bodů pomocí predikce na základě předcházející trajektorie. Tento přístup je výhodný z hlediska náročnosti, neboť nepotřebuje do paměti ukládat další obrazový soubor. Je však nepoužitelný v případě prvních snímků, protože předcházející trajektorie není známa. V případě, že algoritmus neurčí jednoznačně pouze dva body, je pro následující snímek opět udána oblast zájmu jako celý snímek.

Pro správnou identifikaci polohy je nutné, aby pro první snímek sekvence byl dostatečně velký rozsah parametru při správně nastaveném prahu. Takovým způsobem lze nalézt poprvé polohu a velikost identifikačních značek. Při zpracování následujících snímků je pak omezena oblast zájmu a snížen rozsah parametrů. Tím je snížena pravděpodobnost, že při postupné změně intenzity osvětlení dojde k chybným identifikacím.



Obrázek 21: Postup prohledávání sekvence snímků

5.4.1. Zpracování sekvence z fotoaparátu Canon.

V případě fotoaparátu Canon je při zpracování sekvence snímků použito metody nalezení prahu pomocí poměrného zpracování kumulativního histogramu. Poměr pro tuto metodu byl zvolen jako $1/40$, což by mělo odpovídat velikosti robota v obraze. Tato hodnota je však nale-

zena experimentálně a nejedná se tedy o přesnou hodnotu. Počáteční rozsah povoleného počtu pixelů v hledané oblasti je nastaven na 1000 až 2300 obrazových bodů, tato hodnota se však automaticky upravuje pro další snímek podle přesného počtu pixelů v nalezených oblastech. Úprava počtu pixelů je na rozsah: spodní hranice počtu nalezených pixelů – 20% horní hranice počtu nalezených pixelů + 20%.

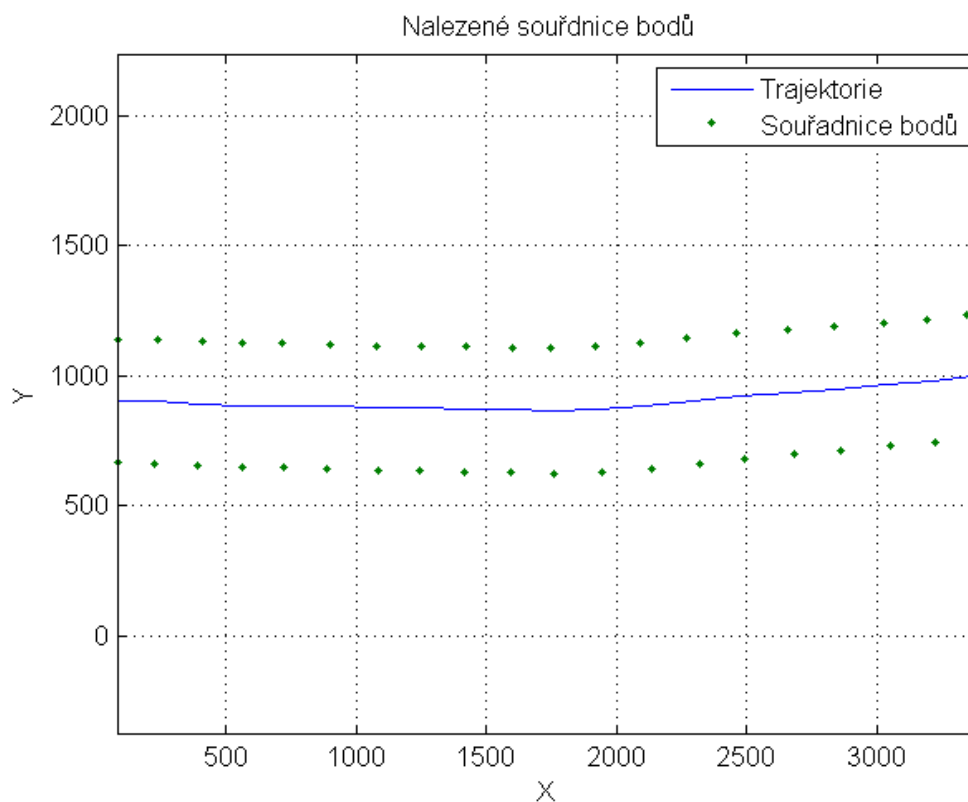
Parametr	Minimální hodnota	Maximální hodnota
Počet pixelů značky	1000	2300
Vzdálenost bodů	400	500
Poměr pro prahování	1/35	
Hodnota prahu	50	
Velikost oblasti zájmu (ROI)	800	800

Tabulka 5: Parametry při zpracování sekvence z fotoaparátu Canon

Nalezené body jsou vypsány v tabulce 6

Nalezené body						
číslo snímku	souřadnice robota X	souřadnice robota Y	souřadnice bodu 1 X	souřadnice bodu1 Y	Souřadnice bodu 2 X	souřadnice bodu 2 Y
Snímek 0	83,43	901,02	86,61	663,55	80,25	1138,49
Snímek 1	229,79	899,53	223,77	661,90	235,80	1137,15
Snímek 2	398,31	891,44	390,23	653,57	406,40	1129,31
Snímek 3	562,99	884,50	562,46	646,03	563,51	1122,97
Snímek 4	716,34	882,82	717,92	643,91	714,76	1121,74
Snímek 5	890,64	879,77	884,16	640,66	897,12	1118,88
Snímek 6	1080,29	874,53	1082,40	634,89	1078,19	1114,16
Snímek 7	1246,38	874,15	1243,33	634,25	1249,44	1114,06
Snímek 8	1418,07	869,32	1416,83	629,01	1419,30	1109,63
Snímek 9	1596,13	866,37	1592,59	626,09	1599,67	1106,65
Snímek 10	1753,27	865,00	1758,94	624,59	1747,60	1105,42
Snímek 11	1929,06	869,56	1941,68	628,70	1916,44	1110,43
Snímek 12	2110,56	882,13	2133,74	642,96	2087,38	1121,29
Snímek 13	2293,68	903,07	2316,36	664,15	2271,01	1142,00
Snímek 14	2474,40	920,26	2488,58	681,19	2460,22	1159,33
Snímek 15	2670,71	935,42	2685,08	696,91	2656,34	1173,93
Snímek 16	2846,22	949,10	2860,51	711,08	2831,92	1187,12
Snímek 17	3039,21	964,33	3053,01	726,96	3025,41	1201,69
Snímek 18	3208,52	977,05	3225,97	740,57	3191,08	1213,52
Snímek 19	3372,55	994,32	3398,65	759,25	3346,44	1229,39

Tabulka 6: Nalezené body v sekvenci snímků z fotoaparátu Canon



Obrázek 22: Zpracování sekvence snímků z fotoaparátu Canon

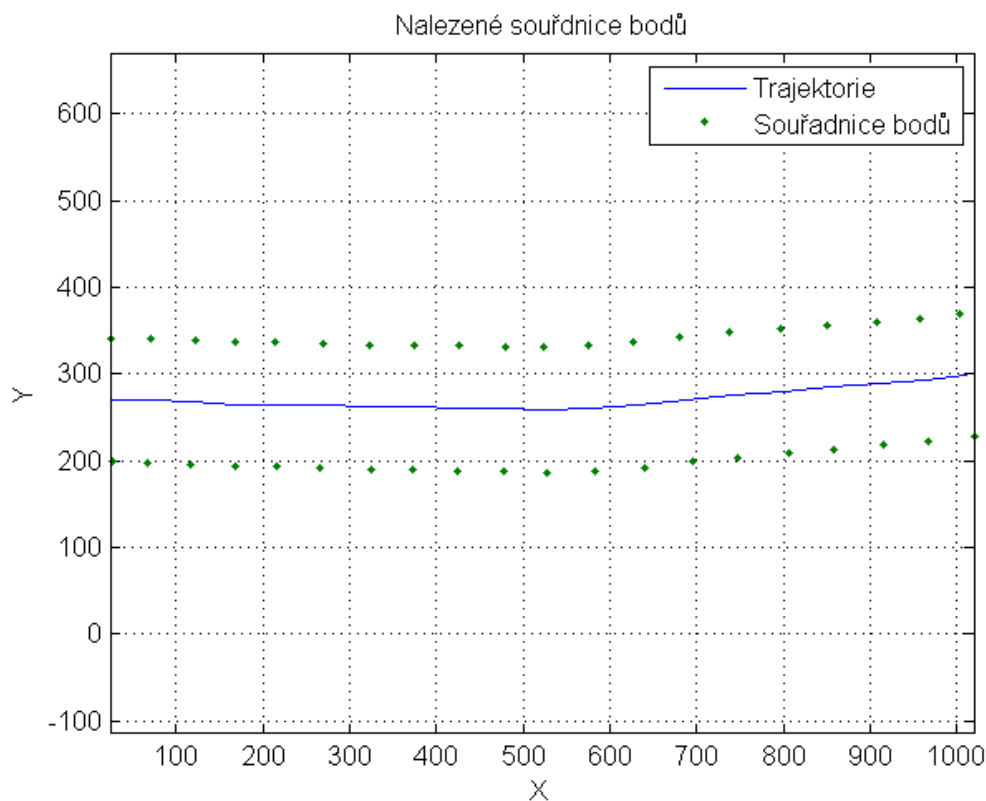
5.4.2. Zpracování sekvence z fotoaparátu Canon se sníženým rozlišením

Pro zpracování sekvence snímků se sníženým rozlišením jsem použil stejnou metodu zpracování jako v případě snímků v plném rozlišení. Pouze jsem upravil parametry tak, aby odpovídaly nižšímu rozlišení. (tabulka 7)

Parametr	Minimální hodnota	Maximální hodnota
Počet pixelů značky	100	250
Vzdálenost bodů	140	150
Poměr pro prahování.	1/35	
Hodnota prahu	51	
Velikost oblasti zájmu (ROI)	200	200

Tabulka 7: Parametry při zpracování sekvence se sníženým rozlišením

Po zpracování byla nalezena trajektorie pohybu. V každém snímku byly identifikované pouze dva body, které vždy odpovídaly identifikačním značkám.



Obrázek 23: Zpracování sekvence snímků se sníženým rozlišením

5.4.3. Zpracování sekvence z kamery Pixelink

Při zpracování sekvence natočené kamerou Pixelink dochází mnohem častěji k nesprávné identifikaci značek. Toto je způsobeno především výrazně horšími světelnými podmínkami, kde se potřebná hodnota prahu i nastavení parametrů potřebných pro správnou identifikaci může skokově měnit.

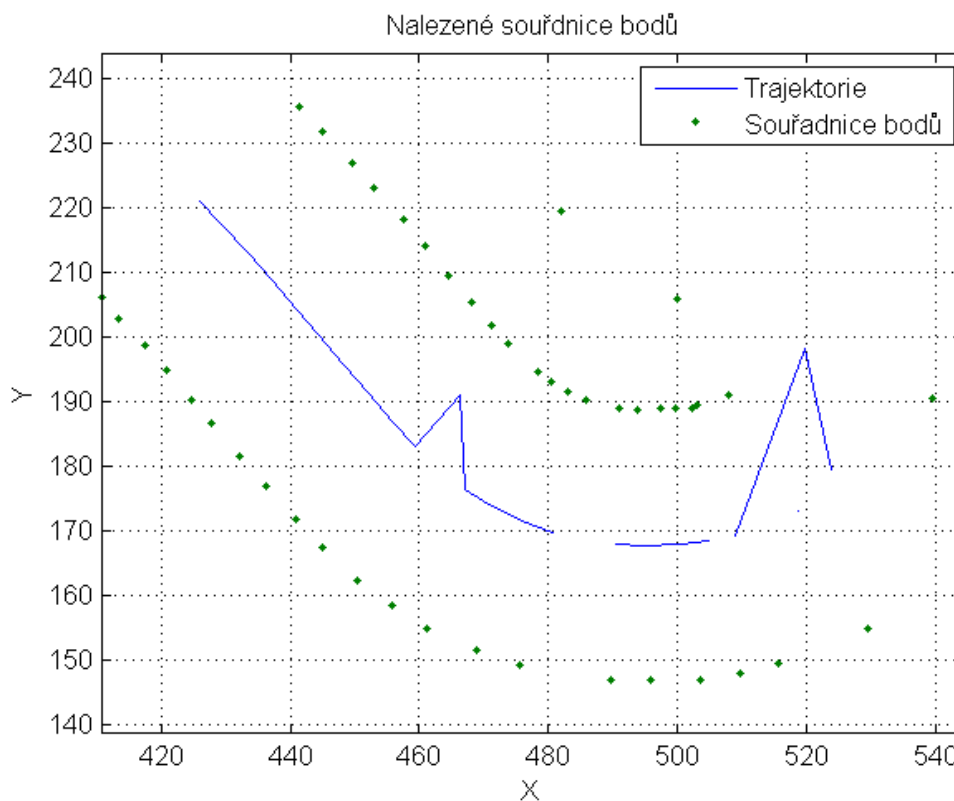
V případě zkrácené sekvence snímků, kde se robot pohybuje pouze mimo oblasti ostrého světla, funguje algoritmus téměř bezchybně.

Parametr	Minimální hodnota	Maximální hodnota
Počet pixelů značky	20	25
Vzdálenost bodů	43	45
Poměr pro prahování	1/40	
Velikost oblasti zájmu (ROI)	80	80

Tabulka 7: Parametry při zpracování sekvence z kamery Pixelink

Nalezené body						
číslo snímku	souřadnice robota X	souřadnice robota Y	souřadnice bodu 1 X	souřadnice bodu1 Y	Souřadnice bodu 2 X	souřadnice bodu 2 Y
Snímek 0	426,08	220,99	441,46	235,73	410,70	206,26
Snímek 1	429,28	217,33	445,05	231,78	413,51	202,87
Snímek 2	433,60	212,90	449,64	227,03	417,56	198,77
Snímek 3	436,91	208,94	453,06	223,04	420,76	194,84
Snímek 4	441,17	204,19	457,52	218,12	424,82	190,27
Snímek 5	444,32	200,47	460,83	214,20	427,82	186,75
Snímek 6	448,44	195,51	464,61	209,52	432,26	181,50
Snímek 7	452,28	191,15	468,20	205,49	436,35	176,81
Snímek 8	455,95	186,76	471,07	201,78	440,83	171,73
Snímek 9	459,36	183,17	473,84	199,02	444,88	167,31
Snímek 10	466,17	190,93	482,05	219,50	450,29	162,36
Snímek 11	467,06	176,49	478,34	194,62	455,77	158,36
Snímek 12	470,78	173,96	480,30	193,06	461,27	154,86
Snímek 13	475,90	171,53	483,03	191,64	468,77	151,42
Snímek 14	480,64	169,78	485,77	190,35	475,51	149,21
Snímek 15	1,00	1,00				
Snímek 16	490,36	167,92	491,07	189,02	489,64	146,82
Snímek 17	494,73	167,71	493,73	188,62	495,74	146,79
Snímek 18	500,55	167,95	497,46	189,00	503,63	146,90
Snímek 19	504,75	168,47	499,79	189,02	509,72	147,92
Snímek 20	1,00	1,00				
Snímek 21	508,92	169,21	502,26	189,07	515,58	149,36
Snímek 22	519,76	198,25	500,01	206,00	539,51	190,50
Snímek 23	523,75	179,54	503,02	189,57	544,49	169,50
Snímek 24	1,00	1,00				
Snímek 25	518,73	172,96	507,87	191,03	529,58	154,89

Tabulka 8: Nalezené body v sekvenci snímků z kamer Pixelink



Obrázek 24: zpracování sekvence z kamery Pixelink

Na sekvenci, kde se robot pohybuje pouze ve stínu, došlo pouze k dvěma chybným identifikacím a ve dvou případech nebyl nalezen žádný bod. Rozsah počtu pixelů v hledaných značkách se mění.

Pro zlepšení fungování tohoto algoritmu při použití této kamery, by bylo vhodné zvážit použití objektivu s větší ohniskovou vzdáleností. Tím se zvýší rozlišení identifikačních značek a algoritmus by tak měl být robustnější. Nevýhodou však bude snížení zorného úhlu.

6. Závěr

V této práci se zabývám zpracováním obrazu ze dvou různých kamer, které snímají z nadhledu robota pohybujícího se po rovině. Cílem je nalezení polohy robota v obrazovém prostoru pomocí dvou bílých značek umístěných na robotu. Podle charakteru prostředí jsem prozkoumal metody segmentace obrazu a zpracování segmentovaných oblastí. Následně jsem určil algoritmus vhodný pro nalezení značek a určení jejich polohy v obrazové rovině.

Pro zpracování snímků jsem vytvořil program v prostředí Microsoft Visual Studio 2010 Express za použití programovacího jazyka C# a .NET Framework library. Tento program má dvě hlavní části. Zpracování samostatného snímku a zpracování sekvence snímků. Dokáže pracovat se soubory typu bmp, jpg, png a CR2.

Jako nejvhodnější metodu segmentace jsem zvolil prahování a vyzkoušel jsem některé metody nalezení vhodného prahu. Nejprve jsem určil vhodný práh experimentálně, následně jsem jej porovnal s prahy nalezenými pomocí Otsuho metody a metody poměrného prahování. Jako nejvhodnější metoda se ukázala metoda poměrného prahování. V segmentovaném obraze jsem následně klasifikoval nalezené oblasti pomocí metody pojmenování spojených oblastí pomocí trasování kontur. U nalezených oblastí odpovídajících parametrům značek je pak nalezen jejich střed pomocí váženého těžiště. Za polohu robota je pak považován střed spojnice těchto těžišť.

Pro snímky, kde se robot pohybuje v rovnoměrně osvětlené oblasti, se použité metody ukázaly jako dostatečně robustní. Problém však nastává v případě kdy je velký rozdíl v intenzitě osvětlení každé značky. V takovém případě algoritmus selhává a není schopen spolehlivě nalézt pozici robota. Rychlost zpracování jednoho snímku při použití kamery Pixellink se pohybuje kolem 90 milisekund. Pokud je zpracována pouze oblast zájmu v okolí předchozí polohy robota, je výpočetní čas přibližně 70 milisekund. Pokud jsou zpracovávány snímky z kamery Canon v plném rozlišení, dosahuje výpočetní čas až 3000 milisekund respektive 1200 ms při omezení oblasti zájmu. Tento čas je možné výrazně snížit snížením rozlišení snímků přibližně na 30% původních rozměrů. V takovém případě je výpočetní čas přibližně 300 milisekund respektive 150 ms při omezení oblasti zájmu.

Další práce na programu vyžaduje optimalizaci a zpřehlednění zdrojového kódu, dále je nutné zlepšit uživatelskou přívětivost programu a dodat možnost přidávat a editovat profily pro zpracování sekvence snímků.

Možným rozšířením algoritmu může být implementace metod založených na edge detection které by mohly zlepšit kvalitu detekce za obtížných světelných podmínek. Dále by bylo možné využít funkcí obsažených v knihovně Egmu CV, což je převedení knihoven OpenCV

do formátu použitelného pro .NET. Tato knihovna umožňuje použít několikanásobně rychlejší procesory v grafických čípech.

Předmětem dalšího výzkumu je možnost použití znalostních metod, které využívají předem získané informace o pozorovaném objektu pro definování šablon nebo modelů objektů. Tyto šablony nebo modely jsou následně porovnány s novým obrazem, ve kterém se hledají shody. Další možností je použití umělých neuronových sítí, které pracují čistě na základě zpracování vstupních dat. Umělou neuronovou síť je nutné nejprve naučit pomocí trénovací množiny dat, kterou je nutné předem získat pomocí jiného segmentačního postupu.

Seznam použité literatury

- [1] TIŠNOVSKÝ P.; *PNG is Not GIF*, root.cz [online] 7. 9. 2006 [cit. 17. 5. 2012] dostupné z <<http://www.root.cz/clanky/png-is-not-gif/>>
- [2] *Understanding What is stored in a Canon RAW .CR2 file, How and Why*, [online] 6. 3. 2010 [cit. 17. 5. 2012] dostupné z <<http://lclevy.free.fr/cr2/>>
- [3] XIN LI, BAHADIR GUNTURK, LEI ZHANG, *Image Demosaicing: A Systematic Survey*, [online] 2008 [cit. 17. 5. 2012] dostupné z <http://www.csee.wvu.edu/~xinl/papers/demosaicing_survey.pdf>
- [4] PAVELEK, M.; JANOTKOVÁ, E.; ŠTĚTINA, J. *Hypertextová skripta Vizualizační a optické měřicí metody*. [online] 2007 [cit. 17. 5. 2012] Dostoevsky z <<http://ottp.fme.vutbr.cz/users/pavelek/optika>>
- [5] WEISSTEIN, ERIC W, *Convolution*, MathWorld-A Wolfram Web Resource, [online] [cit. 17. 5. 2012] dostupné z <<http://mathworld.wolfram.com/Convolution>>
- [6] FISHER, R. B., PERKINS, S., WALKER, A., WOLFART E.; *Hypermedia Image Processing Reference*, John Wiley and Sons, [online] 1996 [cit. 17. 5. 2012] dostupné z <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/>>
- [7] *Počítačové vidění – přednášky*, [online] Červen 2009 - Leden 2012 [cit. 17. 5. 2012] dostupné z <<http://midas.uamt.feec.vutbr.cz/>>
- [8] SHARMA N, AGGARWAL LM. *Automated medical image segmentation techniques*. J Med Phys [online] 2010 [cit. 17. 5 2012.];35:3-14.dostupné z: <<http://www.jmp.org.in/text.asp?2010/35/1/3/58777>>
- [9] THIERRY P., *A new method for grey-level picture thresholding using the entropy of the histogram*, Signal Processing, Volume 2, Issue 3, July 1980, Pages 223-237, ISSN 0165-1684, dostupné z <<http://www.sciencedirect.com/science/article/pii/0165168480900201>>
- [10] OTSU, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.
- [11] BIRDAL, T., *Famous Otsu Thresholding in C#*, The Code Project, [online] červenec 2009 [cit. 17. 5. 2012] dostupné z <<http://www.codeproject.com/Articles/38319/Famous-Otsu-Thresholding-in-C>>

- CHANG F., CHEN C. J., LU C. J., A linear-time component-labeling algorithm
[12] 4 using contour tracing technique, *Computer Vision and Image Understanding*,
Vol. 93, I. 2, str. 206-220, 2004, 1077-3142
- VAN ASSEN, H.C.; EGMONT-PETERSEN, M.; REIBER, J.H.C.; , "*Accurate object
[13] localization in gray level images using the center of gravity measure: accuracy versus
precision*," *Image Processing, IEEE Transactions on* , vol.11, no.12, pp. 1379- 1384,
Dec 2002

Seznam ilustrací

Obrázek 1: Pozorovaný robot.....	11
Obrázek 2: Ukázka záběru pozorované scény.....	13
Obrázek 3: Záběr vyfocený kamerou Canon.....	14
Obrázek 5: Bayerova maska.....	17
Obrázek 6: Histogram obrazu v odstínech šedi.....	18
Obrázek 7: Grafické znázornění funkce diskrétní 2D konvoluce.....	19
Obrázek 8: Gaussovo rozložení s průměrem 0 a rozptylem 1.....	20
Obrázek 9: Určení prahu z histogramu.....	23
Obrázek 10: Histogram snímku z fotoaparátu Canon.....	23
Obrázek 11: vyhledávání následujícího pixelu.....	28
Obrázek 12: Příklad labelingu.....	28
Obrázek 13: Obrazovka 1 - zpracování samostatného obrázku.....	31
Obrázek 14: Obrazovka 2 - zpracování sekvence snímků.....	32
Obrázek 15: Zpracování snímku z fotoaparátu Canon.....	34
Obrázek 16: Zpracovaný snímek z fotoaparátu Canon.....	35
Obrázek 17: Závislost počtu pixelů značky na hodnotě prahu.....	36
Obrázek 18: Prahování snímku z kamery Pixelink.....	37
Obrázek 19: Snímek průjezdu robota ostrým světlem po prahování.....	38
Obrázek 20: Zpracovaný snímek z kamery Pixelink.....	39
Obrázek 21: Postup prohledávání sekvence snímku.....	41
Obrázek 22: Zpracování sekvence snímků z fotoaparátu Canon.....	43
Obrázek 23: zpracování sekvence snímků se sníženým rozlišením.....	44
Obrázek 24: zpracování sekvence z kamery Pixelink.....	46

Seznam tabulek

Tabulka 1: Parametry hledání.....	39
Tabulka 2: Zpracování snímku z fotoaparátu Canon.....	39
Tabulka 3: parametry pro zpracování snímku z kamery Pixelink.....	45
Tabulka 4: nalezené body ve snímku "image1165.bmp".....	46
Tabulka 5: Parametry při zpracování sekvence z fotoaparátu Canon.....	50
Tabulka 6: Nalezené body v sekvenci snímků z fotoaparátu Canon.....	50
Tabulka 7: Parametry při zpracování sekvence se sníženým rozlišením.....	52
Tabulka 7: Parametry při zpracování sekvence z kamery ixelink.....	53
Tabulka 8: Nalezené body v sekvenci snímků z kamer Pixelink.....	54

Obsah přílohy

Přílohou této práce je disk CD-ROM, který obsahuje text práce ve formátu PDF, dále pak spustitelný program a veškerý zdrojový kód tohoto programu. Pro spuštění programu je nutný operační systém Windows XP, Windows Vista nebo Windows 7 s nainstalovaným Microsoft .NET Framework 4. CD dále obsahuje sekvence snímků zpracovávané v této práci.

Struktura Adresářů na disku je následující:

- /text – Vlastní text práce ve formátu PDF
- /exe – spustitelný soubor programu.
- /source – Zdrojový kód programu
- /bitmap – Obrazová data.