Mendelova univerzita v Brně
Provozně ekonomická fakulta

Manipulace s objekty pomocí mobilního robotu založená na analýze obrazu

Bakalářská práce

Vedoucí práce:
Ing. Vít Ondroušek, Ph.D.

Robin Antonič

Brno 2015

Rád bych poděkoval vedoucímu této práce Ing. Vítu Ondrouškovi, Ph. D. za jeho odborné vedení, rady a pomoc při psaní této práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Manipulace s objekty pomocí mobilního robotu založená na analýze obrazu**
vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*
Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.
Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 19.května 2015 .....

Abstract

ANTONIČ, Robin. *Objects Manipulation Using the Autonomous Robot Based on Image Processing*. Bachelor thesis. Brno, 2015.

Bachelor thesis is focused on design and implementation of a suitable algorithm that will be used for object recognition in known environment by autonomous mobile robot K3. Common algorithms of computer vision and selected libraries of image processing are described in the first part of this bachelor thesis. Design of the new method intended for the control unit of the K3 robot is designed and implemented in the second part. Practical tests of the designed method are performed to evaluate capability and usability of the method.

Keywords

C#, Image analysis, AForge.NET, Autonomous robot

Abstrakt

ANTONIČ, Robin. *Manipulace s objekty pomocí mobilního robotu založená na analýze obrazu*. Bakalářská práce. Brno, 2015.

Bakalářská práce se zabývá návrhem a implementací vhodného algoritmu, který je určen pro detekci objektů v předem známém prostředí za pomocí autonomního robotu K3. V teoretické části práce jsou popsány známé algoritmy analýzy obrazu a vybrané knihovny zajišťující analýzu obrazu. V praktické části práce je popsán návrh algoritmu a jeho implementace do řídicí jednotky robota K3. Funkčnost algoritmu bude otestována v prostředí Mendelovy univerzity.

Klíčová slova

C#, Analýza obrazu, AForge.NET, Autonomní robot

Obsah

1 Úvod a cíl práce

1.1 Úvod

Počítačové vidění je oblast výpočetní techniky, která se v současnosti využívá pro návrh inteligentních algoritmů řízení v mnoha rozličných oblastech. Lze uvést například analýzu snímků při diagnostice nemocí v medicíně, rozpoznávání objektů na dopravníkovém páse, kontrola kvality v potravinářství, autonomní navigace mobilních robotů ve vojenském průmyslu atd.

Autonomní robot je stroj, jehož činnost je řízena algoritmem z oblasti umělé inteligence na základě vstupů ze senzorické soustavy bez zásahu operátora. Tato bakalářská práce se zaměřuje na návrh a implementaci algoritmu počítačového vidění určeného pro konkrétní indoor autonomní kolový robot, který je vyvíjen týmem AiStorm na Ústavu informatiky PEF MENDELU. Řešená problematika by měla přispět k návrhu komplexní řídicí jednotky, která umožní robotu K3 účastnit se robotických soutěžích v disciplíně Bear Rescue.

1.2 Cíl práce

Cílem práce je tvorba algoritmu a jeho implementace do nejvyšší vrstvy autonomního indoor robotu K3. Algoritmus umožní na základě vhodných metod analýzy obrazu analyzovat a vyhodnocovat obraz prostoru pořízený kamerou robotu. Tyto získané informace budou sloužit pro navigaci autonomního robotu v prostoru, pro plánování trajektorie k cílovému objektu a také pro uchopení objektu robotem.

2 Analýza obrazu

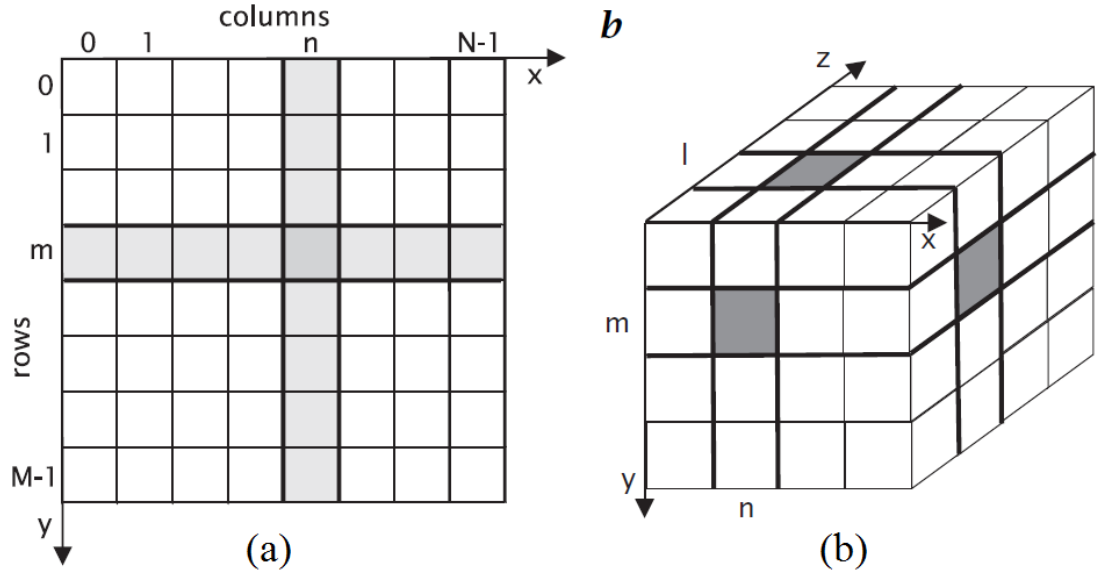
Metoda zpracování obrazu, která slouží například pro určování jakosti a kvality produktů, při které je nahrazeno vizuální hodnocení člověka. Je založena na vyhodnocování digitálního obrazu, který byl pořízen kamerou. V průmyslu a praxi se častěji používá název strojové vidění (Analýza obrazu, 2011-2015).

2.1 Reprezentace digitálního obrazu

Obraz představuje rozložení spektrálního záření v rovině. Obraz může být vyjádřen matematicky jako spojitá funkce dvou proměnných prostoru:

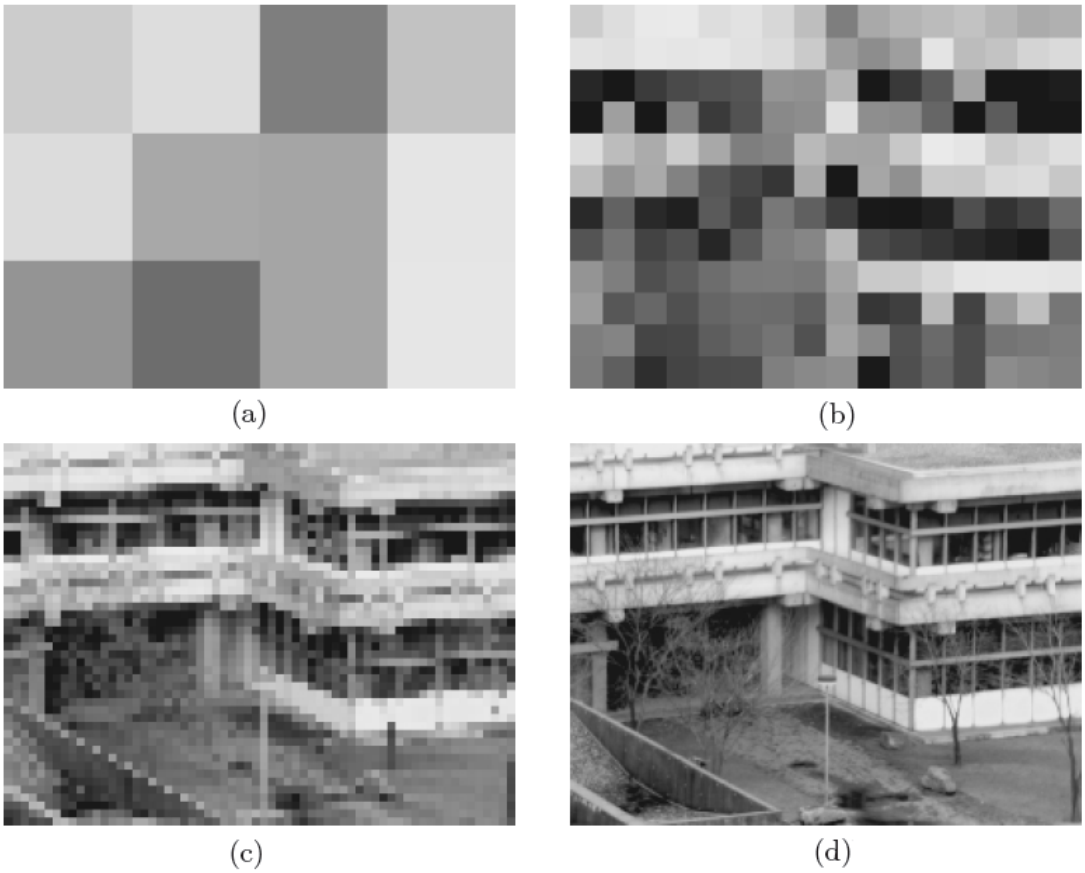
$$E(x_1, x_2) = E(x) \tag{1}$$

Počítače neumí zacházet se spojitými obrazy, proto musí být obraz reprezentován pomocí dvoudimenzionálního pole bodů. Bod je v 2D rovině pojmenován pixel nebo pel. Pixel vyjadřuje hodnotu obrazu v dané pozici. Pixely jsou v nejjednodušším případě reprezentovány pomocí obdélníkové mřížky. Pozice pixelů v mřížce je dána obecnou notací pro matice, takže index m určuje pozici řádku a n pozici sloupce. Index m musí být v rozsahu 0 až $M - 1$ a index n v rozsahu $N - 1$. Hodnota N udává počet sloupců a M počet řádků mřížky. Digitální obraz tedy obsahuje $N * M$ pixelů. Mřížka obsahuje osu y jdoucí od shora dolů a osu x jdoucí zleva doprava (Jähne, 2002).



Obrázek 1: Reprezentace digitálního obrazu pomocí pole bodů znázorněného v obdélníkové mřížce a) 2-D obraz b) 3-D obraz. Zdroj: (Jähne, 2002)

Každý pixel necharakterizuje jen bod, ale také základní buňku mřížky. Pokud se velikost pixelů snižuje, tak poté se zlepšuje prostorové rozlišení a snižují se nespojitosti hran pixelů, a tím postupně získáme dojem prostorově kontinuálního obrazu. Tento dojem nastane, pokud se velikost pixelů stane menším než prostorové rozlišení našeho vnímacího systému (Jähne, 2002).



Obrázek 2: Digitální obraz, který je reprezentován rozdílnou velikostí obdélníkové mřížky a) 3 × 4 b) 12 × 16 c) 48 × 64 d) 192 × 256 pixelů. Zdroj: (Jähne, 2002)

2.1.1 Obrazy dle barevné hloubky

Barevná hloubka je vlastnost udávající počet bitů, které byly použité pro popis pixelu v bitmapovém obrazu.

Binární obraz

Obraz, který obsahuje body reprezentované jedním bitem. Hodnoty bitu můžou ukládat informace pouze o dvou barvách. Kdy první hodnota reprezentuje pozadí a druhá popředí obrazu. Výhodou tohoto obrazu je snadnější vyhodnocení.

Obraz ve stupních šedi

Obraz, který obsahuje body, jejichž hodnota může být vyjádřena pomocí funkce $f(x, y)$, která daným souřadnicím stanovuje intenzitu šedi v daném bodě. Typicky je 256 úrovní jasu. Barevný obraz lze transformovat na tento typ obrazu na základě vzorce:

$$I = 0.33 \cdot R + 0.33 \cdot G + 0.33 \cdot B \tag{2}$$

$$I = 0.2125 \cdot R + 0.7154 \cdot G + 0.0721 \cdot B \tag{3}$$

Vzorec (2) je metoda průměrování. U tohoto vzorce jsou si všechny koeficienty rovny. Vzorec (3) přizpůsobuje své koeficienty lidskému vnímání, jelikož zrak je nejvíce citlivý na odstíny zelené barvy a nejméně citlivý na modré odstíny.

Barevný obraz

Obraz obsahující body reprezentované různými kombinacemi třech barevných složek, které mají 256 úrovní jasu. Pomocí modifikace obrazu lze dosáhnout lepších výsledků např. zpracovávat pouze s nejvýznamnější složkou barev, převedení na obraz ve stupních šedi, zpracovávat každou barevnou složku zvlášť, převést na jiný barevný model atd.

2.1.2 Barevné modely

Model, který využívá základní barvy a jejich kombinacemi se získává výsledná barva. Barva světla může být reprezentována jedinou vlnovou délkou záření až po směsi více záření různých vlnových délek, ale lidské oko je schopné vnímat pouze některé. Modely lze rozdělit na aditivní a subtraktivní modely. Aditivní modely využívají tři základní barvy (červená, modrá, zelená). Dané barvy se sčítají a výsledná barva má větší intenzitu. Základní barvy subtraktivního modelu jsou žlutá, purpurová a azurová. Pokud se přidá barva, tak se ubere část světla původní barvy.

RGB model

Aditivní barevný model se třemi základními barvami, který je používán zejména v barevných monitorech a projektorech. Zbylé spektrum barev modelu je dáno kombinacemi různých úrovní intenzity základních barev. Intenzita každé barvy může nabývat hodnot od 0 do 255.

HSL model

Barevný model sestávající ze tří složek, tj. odstín barvy, nasycení barvy a intenzita barvy. Tento model se nejvíce blíží k lidskému vnímání barev a je zejména používán v grafických aplikacích.

YCbCr model

Je používán při zápisu rastrových obrázků ve formátu JPEG. Je složen ze tří komponent (luminance, modrý chrominační komponent, červený chrominační komponent). Hodnota Y leží v intervalu $< 0.0, 1.0 >$ a hodnoty C_B,C_R v intervalu $< -0.5, 0.5 >$ (Moderní počítačová grafika, 1998).

YCbCr není absolutní barevný model, ale pouze určitý způsob kódování RGB informací. Převod RGB modelu do modelu YCbCr lze vyjádřit pomocí matematického vztahu:

$$C_B = 0.5643 \cdot (B - Y) \tag{4}$$

$$C_R = 0.7133 \cdot (R - Y) \tag{5}$$

$$\begin{bmatrix} Y \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & -0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6}$$

2.2 Hlavní úkoly analýzy

Předzpracování obrazu

Hlavním úkolem tohoto kroku je určení oblasti zájmu v daném obrazu. Vstupní obrazy mohou obsahovat určité množství šumu, který by měl být v tomto kroku odstraněn nebo redukován pomocí úpravy jasu, kontrastu, histogramu, velikosti a dalších vlastností obrazu.

Segmentace obrazu

Segmentace obrazu je jedním ze základních kroků analýzy obrazu. Je to proces během kterého je rozdělen obraz na nepřekrývající oblasti, které mají společné vlastnosti a na oblasti, které nemají souvislost s věcným obsahem obrazu. Výsledek segmentace je obraz, který obsahuje jednotlivé vyznačené oblasti. Výsledkem tohoto procesu je obraz se stejnými rozměry, jako původní obraz obsahující vyznačené oblasti. Binární reprezentace obrazu je využívána u obrazů, které obsahují pouze jednu význačnou oblast a pozadí. Tyto části jsou poté reprezentovány binárními hodnotami. Pokud je obraz rozdělen na více oblastí, tak jsou oblasti reprezentovány například pomocí barev nebo indexů (Analýza biomedicínských obrazů, 2013).

Následné zpracování

Pro zlepšení segmentovaného obrazu může být vhodné aplikovat další úpravy na obraz.

Získávání vlastností objektů

Metoda, která má za úkol získávat informace o typických vlastnostech objektů ve

zpracovaném obraze. Pomáhá snižovat složitost v klasifikaci objektů a zvyšuje úspěšnost klasifikace.

Klasifikace

Cílem toho kroku je klasifikace zpracovaného obrazu na základě získaných vlastností objektů. Využívají se statistické analýzy a algoritmy strojového učení (IJEAT, 2015).

2.3 Vybrané metody segmentace

2.3.1 Prahování

Je to jedna z nejstarších a nejjednodušších segmentačních metod, která se používá pro segmentaci objektů z obrazu. Použití metody na obraz je vhodné, pokud se objekty obrazu výrazně liší od pozadí (Katedra kybernetiky ZČU, 2015).

Prosté prahování

Princip prahování je založen na transformaci vstupního obrazu f na binární výstupní obraz g. Segmentace obrazu využívá práh P pro stanovení výstupní hodnoty obrazu, pokud je intenzita objektu větší než práh, poté je výstup 1 jinak nula. Prosté prahování lze matematicky vyjádřit pomocí vztahu:

$$g(x,y) = \begin{cases} 1 & \text{pro} f(x,y) \geq P \\ 0 & \text{pro} f(x,y) < P \end{cases} \tag{7}$$

Prahování s více prahy

Na obraz lze také aplikovat prahování s více prahy, při kterém bude mít výstupní obraz g počet barev rovný počtu prahů + 1. Víceprahování lze matematicky vyjádřit pomocí vztahu:

$$g(x,y) = \begin{cases} 1 & \text{pro } f(x,y) \in M_1 \\ 2 & \text{pro } f(x,y) \in M_2 \\ n & \text{pro } f(x,y) \in M_n \\ 0 & \text{jinak} \end{cases} \tag{8}$$

Poloprahování

Při poloprahování jsou ovlivněny jen hodnoty, které jsou menší než práh a zbytek obrazu zůstává neovlivněn. Výstupním obrazem g tedy nemusí být binární obraz. Poloprahování lze matematicky vyjádřit pomocí vztahu:

$$g(x,y) = \begin{cases} \text{f(x,y)} & \text{pro } f(x,y) \geq P \\ 0 & \text{pro } f(x,y) < P \end{cases} \tag{9}$$

Charakter výstupního obrazu je zásadně ovlivněn volbou hodnoty prahu, která může

být stanovena na základě pokusů, pomocí metod určování prahu např. hledání lokálního minima, či jinak (Katedra kybernetiky ZČU, 2015).

2.3.2 Regionově orientované metody

Jsou metody, které jsou založeny na detekci oblastí obrazu namísto hran a jsou účinnější pro zašuměné obrazy. Pokud je obraz hodně zašuměn, poté mají hranové operátory problém z detekcí hran. Důležitým kritériem detekce je homogenita oblasti, které může být charakterizována např. barvou, tvarem, modelem, úrovní šedi, texturou atd. (Obrazové segmentační techniky, 2005).

Narůstání oblasti

Metoda narůstání oblasti je založena na rozšiřování oblastí, na základě podobných vlastností bodů, a tím stanovuje oblasti v obraze. Oblast je tedy množina propojených pixelů, která má homogenní vlastnosti. Oblast je vytvořena iteracemi, které rozšiřují okolí semínkového bodu ve čtyřech nebo osmi směrech. Výhodou metody je schopnost eliminovat velký šum v obraze. Průběh algoritmu lze rozdělit do následujících fází:

1. Stanovení semínkových bodů
2. Analýza sousedních pixelů bodu
3. Rozhodnutí jestli má být pixel přidán
4. Ukončení algoritmu

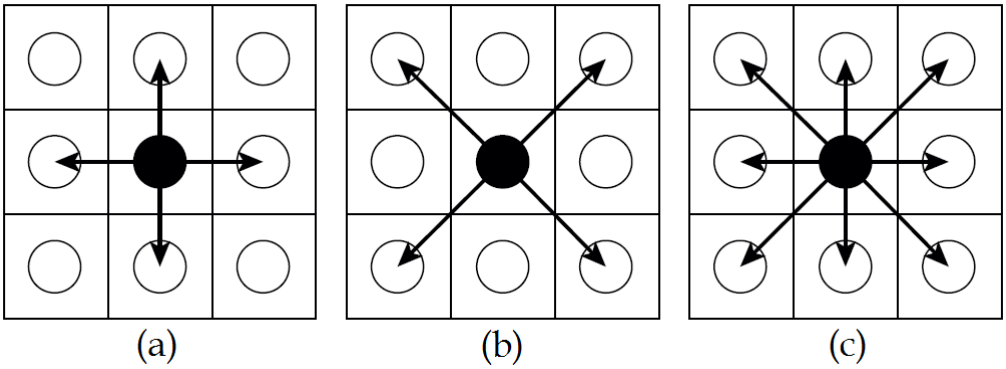
Inicializační semínkové body mohou být do obrazu vloženy interaktivně uživatelem, nebo automaticky na základě náhody, nebo předchozí analýzy. Okolní body výchozích bodů jsou prohlíženy a pomocí jednoho z výše stanovených kritérií se rozhoduje, zda budou přidány k nově rostoucímu objektu (Analýza biomedicínských obrazů, 2013).

Pixel může být přidán do oblasti podle statického nebo dynamického kritéria. Statické kritérium používá vlastnost semínkového bodu, která je porovnávána s vlastností testovaného pixelu podle matematického vzorce:

$$\|p_s - p_j\| \leq T \tag{10}$$

(10) p_s je vlastnost semínkového bodu, p_j je vlastnost testovaného pixelu a T je práh. Pokud rozdíl $\|p_s - p_j\|$ splňuje danou podmínku, tak je bod přidán do oblasti.

Pokud testovaný pixel splňuje danou podmínku, tak je přidán k rostoucí oblasti a v další iteraci algoritmu se stává výchozím bodem, u kterého se zjišťuje jeho okolí. Vlastnost testovaného parametru může být porovnávána podle dynamického kritéria



Obrázek 3: Způsoby prohledávání okolí. a) Horizontální a vertikální okolí b) Diagonální okolí c) Horizontální, diagonální a vertikální okolí. Zdroj: (Analýza biomedicínských obrazů, 2013)

, např. vlastnost naposledy přidaného pixelu, střední hodnotou pixelů v aktuální oblasti.

$$\|p_i - p_j\| \leq T \tag{11}$$
$$\|p_i - p_a\| \leq T \tag{12}$$

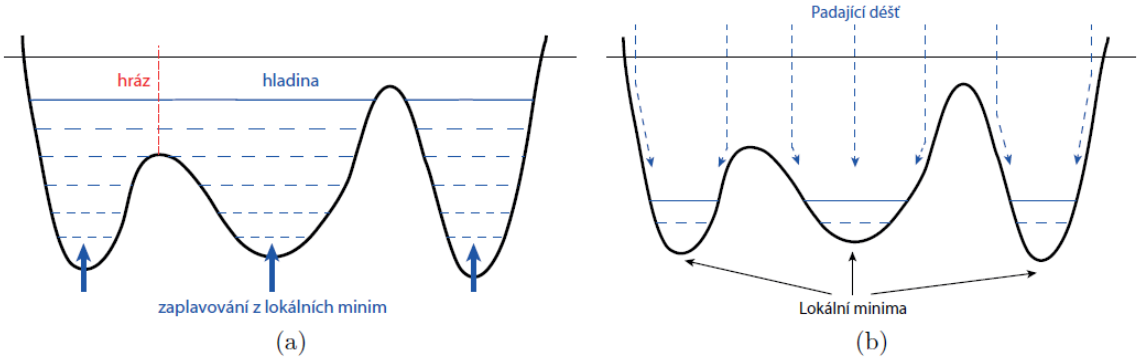
Proměnná p_i vyjadřuje vlastnost naposledy přidaného pixelu. Proměnná p_a reprezentuje střední hodnotu pixelů v aktuální oblasti. Pokud jejich rozdíl splňuje danou podmínku, tak je bod přidán k oblasti. Proměnná p_j reprezentuje vlastnost testovaného pixelu. Pokud rozdíl $\|p_i - p_j\|$ splňuje podmínku, tak je poté přidán k dané oblasti.

Použití dynamického kritéria může být výhodné u oblastí, u kterých není sledovaná vlastnost pixelů homogenní, ale postupně se mění určitým směrem např. u nerovnoměrně osvětleného objektu. Algoritmus končí, pokud žádný pixel nesplňuje danou podmínku (Analýza biomedicínských obrazů, 2013).

Transformace rozvodím

Morfologická metoda, která vychází z oblasti geografie. Na obraz je poté nahlíženo jako na topografický reliéf či terén. Terén je poté zaplavován vodou. Povodí v obraze jsou poté zaplňována vodou směrem od lokálních minim obrazu. V oblasti, kde se může voda slít dohromady z dvou různých povodí, jsou vytvořeny hráze. Proces zaplavování je ukončen, pokud je dosaženo absolutního maxima obrazu. Výsledkem metody je obraz, který je rozdělen do oblastí reprezentujících jednotlivé povodí, které jsou odděleny hrázemi.

Nevýhodou metody je nadměrná segmentace obrazu. Pro odstranění velkého počtu oblastí lze na výsledek metody aplikovat metodu spojování oblastí (Obrazové segmentační techniky, 2005).



Obrázek 4: Princip metody, která znázorňuje zaplavování z lokálních minim obrazu. Zdroj: (Analýza biomedicínských obrazů, 2013)

Dělení a slučování oblasti

Je metoda, která pracuje s celým obrazem. Určitá vlastnost obrazu je zvolena jako kritérium, které určí, jestli je oblast jednoznačná či ne. Kritérium je založeno na analýze určité vlastnosti regionu, např. na statistické analýze histogramu jasu, histogramu barev atd. Pokud histogram vykazuje určitou nejednoznačnost např. vysokou směrodatnou odchylkou, tak je region rozdělen do čtyř kvadrantů. Každý kvadrant je poté otestován, a pokud nevyhovuje kritériu, tak je znovu rozdělen. Celý proces pokračuje do té doby, než dosáhne na úroveň jednoho pixelu. Vztah mezi regionem a kvadranty je často reprezentován pomocí stromové struktury, která obsahuje uzly se čtyřmi potomky (Russ, 2011).

Obrázek 5: (a) Obraz v odstínech šedi (b) Dělení a slučování oblasti po čtyřech iteracích (c) Dělení a slučování oblasti po šesti iteracích. Zdroj: (Russ, 2011)

2.3.3 Hranově orientované metody

Detekce hran je velmi důležitá oblast zpracování obrazu na nižší úrovni. Hrany jsou obrazové body s významnou změnou hodnoty jasu. Hrana může být určena velikostí a směrem, jestliže je chápána, jako vlastnost bodu obrazu, který je započten, jako funkce obrazu v okolí daného bodu. Rozmístění objektů obrazu je určováno na základě změny nebo přerušení hodnot jasu v obraze, které jsou jedny ze základních obrazových charakteristik. Jasové hrany jsou místní změny hodnot jasu v obraze z dané úrovně na jinou a globální změny jsou nazývány hraniční segmenty obrazu. Ideální hrana může být vyjádřena pomocí skokové funkce, ale typicky je změna hodnoty jasu v obraze postupná, proto je lepší využít šikmou funkci.

Průběh detekce lze rozdělit do tří fází. První fází je filtrování, které je určeno pro aplikaci filtrů, které částečně eliminují šum vzniklý při kvantování, rozmazání či vzorkování obrazu nebo nesprávně nastavenou kamerou. Další fází je diferenciace, která zvýrazňuje oblasti obrazu na základě významné změny intenzity jasu. Poslední fází je detekce, která je určena pro lokalizaci a detekci bodů v obraze s nejvýznamnější změnou intenzity jasu (Obrazové segmentační techniky, 2005).

Sledování hranice

Metoda, která je používána pro nalezení ohraničení objektů. Metoda využívá určitou vlastnost pixelů, která umožňuje určit, zda daný pixel patří do daného objektu či ne. Digitální obraz je poté procházen po řádcích, a pokud je nalezen pixel s danou vlastností, tak se prochází pixely v okolí v protisměru hodinových ručiček, dokud není nalezeno původní místo. Metoda není příliš účinná u obrazů, které obsahují velký šum nebo u objektů, které mají příliš složitý tvar.

Aktivní kontura

Metoda založená na postupném tvarování kontur k hraně objektu v digitálním obraze. Metoda používá model aktivní kontury, který je řízen uzavřenou konturou, která je deformována vnitřními, obrazovými nebo vnějšími silami. Obrazové síly slouží pro tvarování kontury k hraně objektu. Vnější síly vyplívají z počáteční pozice kontury. Vnitřní síly slouží pro kontrolu hladkosti průběhu (Obrazové segmentační techniky, 2005).

Hranové detektory

Hranové detektory jsou pokročilé metody, které jsou používány pro detekci hran v obraze. Detektory lze rozdělit do dvou kategorií podle řádu derivace. První kategorií jsou detektory, které používají derivaci prvního řádu pro určení hran v obraze. Do této kategorie patří např. Kirschův detektor, Sobelův detektor a Laplaceův detektor. Druhou kategorií jsou detektory využívající derivaci druhého řádu např. Marr-Hildrethův filtr. Mezi nejpoužívanější detektory pro detekci hran v digitálním obraze je používán Cannyho hranový detektor. Algoritmus detektoru se skládá z několika

dílčích kroků, které slouží pro získání co nejlepšího výsledku v dvourozměrném obraze. Postup lze rozdělit do následujících kroků (Analýza biomedicínských obrazů, 2013):

- Aplikace Gaussova filtru pro eliminaci šumu v obraze.
- Zjištění derivací ve směru gradientů daného obrazu.
- Nalezení lokální maxima derivací.
- Aplikace prahování pro detekci potencionální hran.
- Potlačení pixelů, které nejsou hranové pomocí sledovaní hrany
- Prahování s hysterezí

(a) (b)

Obrázek 6: (a) Výchozí snímek. (b) Snímek po aplikaci Cannyho hranového detektoru. Zdroj: (Berkeley Robotics and Intelligent Machines Lab, 1996)

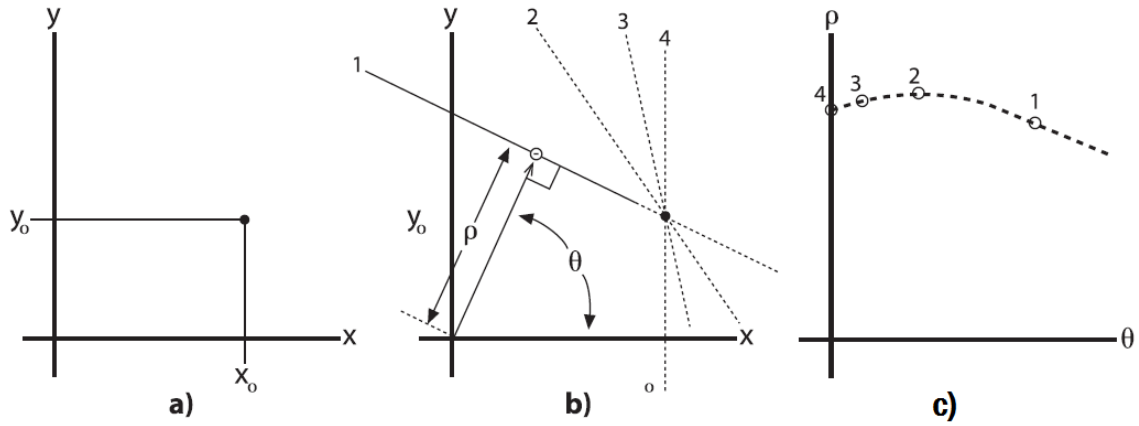
Houghova transformace

Metoda je určena pro detekci přímek, kružnic nebo jiných jednoduchých tvarů v obraze. Původní transformace byla používána pouze pro určování přímek na základě poměrně rychlého prohledání binárního obrazu (OpenCV, Hough Line Transform, 2015).

Houghova transformace pro vyhledání přímek transformuje přímky z kartézského systému souřadnic do systému polárních souřadnic. Přímka je převedena pomocí vzorce:

$$\rho = x \cos\theta + y \sin\theta \tag{13}$$

Proměnná ρ vyjadřuje velikost normály, která spojuje přímku a počátek souřadnicového systému. Proměnná θ reprezentuje úhel sklonu normály vzhledem k ose X. Přímka je poté reprezentována polárními souřadnicemi (ρ, θ). Souřadnice musí splňovat podmínku $\rho > 0$ a také $0 < \theta < 2\pi$.



Obrázek 7: a) Bod se souřadnicemi x_0, y_0 zobrazený v kartézském systému. b) Skupiny přímek, které prochází daným bodem, určené polárními souřadnicemi. c) Část sinusoidy, která zobrazuje procházející přímky, vyjádřené polárními souřadnicemi. Zdroj: (Bradski, 2008)

Obrazovým bodem prochází přímky, které mají společný průsečík viz Obr. 7.b. Každá tato přímka je převedena do systému polárních souřadnic. Poté jsou přímky zobrazeny pomocí bodů v parametrickém prostoru. Tyto body následně vytvářejí část sinusoidy viz Obr. 7.c.

Tato část je provedena pro každý obrazový bod. V parametrickém prostoru tedy vznikne určitá množina sinusoid. Pokud se sinusoidy reprezentující body v parametrickém prostoru protnou ve stejných souřadnicích, tak souřadnice určují přímku na které leží tyto body. Lze také stanovit práh, který stanoví minimální počet bodů ležících na přímce, a tím ovlivnit výslednou detekci. Také lze stanovit konkrétní θ úhel a detekovat pouze přímky s daným úhlem (OpenCV, Hough Line Transform, 2015).

2.3.4 Znalostní metody

Kategorie obsahující metody, které jsou založeny na dříve získaných znalostí a informací (např. tvar, struktura, barva apod.) o objektech v obraze. Znalosti jsou určeny modely nebo šablonami objektů, které jsou používány pro srovnání s novým obrazem

na základě nalezených shod. Tyto metody je vhodné použít, pokud jsou si struktury objektů v obraze vzájemně podobné, naopak pro složité či variabilní struktury je segmentace složitá (Obrazové segmentační techniky, 2005).

2.3.5 Hybridní metody

Jsou metody, které nezapadají do žádné z předchozích kategorií. Mezi tyto metody patří například použití neuronových sítí nebo matematické morfologie. Rozbor těchto metod přesahuje rozsah této práce.

3 Autonomní robot

Je robotické zařízení, které je řízené programem a pracuje zcela samostatně. Robot získává informace o prostředí pomocí senzorického systému, poté tyto informace analyzuje a zpracovává pomocí vhodných metod a následně tyto informace vyhodnotí.

3.1 Základní řešené problémy

Mapování

Mapa reprezentuje určitý model prostředí, který je používán autonomním robotem pro lokalizaci a plánování cesty. Mapa prostředí může být buď předem známa, nebo může být robotem dynamicky vytvářena podle pohybu robotu v prostředí. Druhá možnost je používána mnohem častěji, protože mapy ve vhodném formátu nebo s aktuálním stavem prostředí nejsou často dostupné. Mapy jsou rozděleny do tří kategorií (Siegwart, 2004):

- **Senzorická mapa**
  Mapa, která reprezentuje prostředí pomocí získaných senzorických dat. Senzorickou mapou může být např. mřížka obsazenosti, která reprezentuje prostředí dílky, které jsou uspořádány do mřížky.
- **Topologická mapa**
  Nejabstraktnější model prostředí, který reprezentuje mapu pomocí grafu. Uzly grafu znázorňují důležitá místa pro robota. Hrany grafu určují, jak se robot dostane z jednoho uzlu do druhého. Tento typ mapy nemusí obsahovat geometrické informace o prostředí.
- **Geometrická mapa**
  Tato mapa používá geometrické objekty např. polygony, kružnice atd. Výhodou mapy je její abstrakce od detailů prostředí.

Lokalizace

Tato úloha představuje jeden z klíčových problémů robotiky, který je podmínkou pro autonomní chování robotů a jeho vyřešení je využíváno po další robotické úlohy např. plánování cesty či navigace. Podle charakteru měření lze lokalizaci rozdělit na (Siegwart, 2004):

- **Relativní**
  Relativní lokalizace odhaduje změnu polohy robotu vzhledem k předcházející poloze robotu. Celková změna od počáteční až poslední polohy je určena jednotlivými dílčími změnami. Tento typ lokalizace je určen pro krátkodobý odhad polohy. Hlavním prostředkem relativní lokalizace je zejména odometrie.
- **Absolutní**

Tento typ lokalizace odhaduje absolutní polohu robotu v prostředí pomocí jednorázového měření a bez informací o událostech, které předcházeli dosažení dané polohy.

Plánování cesty

Úloha, která umožňuje robotu se dostat z bodu A do bodu B. Schopnost efektivně naplánovat cestu je závislá na přesnosti, z které robot získává informace o prostředí, na počtu překážek, které se vyskytují v prostředí a také na schopnosti lokalizace robotu v mapě. Plánování cesty je spojeno s problémem nalezení nejkratší cesty z bodu A do bodu B.

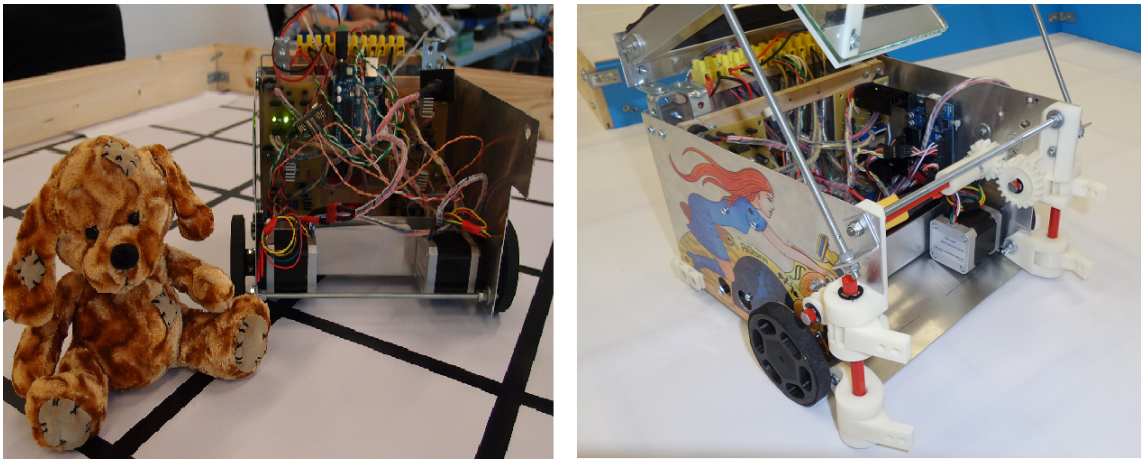
Synchronizovaná lokalizace a mapování (též SLAM)

Úloha při které si autonomní robot vytváří mapu prostředí a zároveň tuto mapu využívá pro svoji lokalizaci v prostředí.

3.2 Robot K3

Robot K3 je cílová platforma, na které bude testován algoritmus pro detekci a uchopení objektů ve scéně. Nejnižší vrstva pro řízení robotu je napsána v jazyce C#, proto byla zvolena technologie .NET pro vývoj tohoto algoritmu. V konstrukci robotu je umístěn tablet HP ElitePad s operačním systémem Windows 8.1, který je propojen pomocí USB portu s řídicí jednotkou robotu. Součástí tabletu jsou dvě integrované kamery.

Kamera, která bude používána robotem, je umístěna v zadní části tabletu a je schopna pořizovat snímky v rozlišení 1280 x 720. Tato kamera je používána robotem pro získávání snímků, které jsou poté analyzovány a vyhodnocovány. Na konstrukci robotu je umístěno zrcátko, které je skloněno pod určitým úhlem a slouží pro rozšiřování zorného pole robotu.



Obrázek 8: Autonomní robot K3

Konstrukce robotu je kovová a její součástí jsou díly z materiálu PVC. Pohyb robotu je zajištěn dvěmi krokovými motory. Součástí robotu je platforma Arduino, která se stará o řízení motoru a prácí ze senzorickým systémem robotu.

4 Vývojové prostředky

4.1 .NET Framework

Platforma, která je určena pro vývoj aplikací pod systémy Windows, Windows Phone, Windows Server a také pod platformou Microsoft Azure. Součástí .NET Frameworku je běhové prostředí CLR ("Common Language Runtime"), které má za úkol poskytovat služby usnadňující vývojový proces. A také CLS součást("Common Language Specification"), která slouží pro usnadnění spolupráce mezi frameworkem a dalšími programovacími jazyky a tím poskytuje částečnou jazykovou nezávislost frameworku (MSDN, 2015).

4.2 Jazyk C#

C# je objektově orientovaný programovací jazyk, který slouží pro tvorbu aplikací běžících pod platformou .NET Framework (Sharp, 2010).

4.3 Framework AForge.NET

Framework s otevřeným obsahem, který je používán pro vývoj a výzkum v oblastech počítačového vidění, umělé inteligence a robotiky pro platformu .NET (AForge.NET Framework, 2008-2012). Mezi hlavní výhody frameworku patří jeho jednoduché použití a také jeho snadná implementace do projektu. Další výhodou je dostupná dokumentace frameworku, která je velmi dobře zpracována. Největším záporem daného frameworku je jeho nízký výkon při zpracování obrazu, jak vyplívá z dané tabulky 1. Zde jsou uvedeny základní informace o třídách a metodách frameworku AForge.NET, které budou používány v praktické částí práce:

Třída Substract

Třída Substract je používána pro filtrování. Třída zpracovává zdrojový obraz a obraz, který je poté překrýván. Tyto obrazy musí mít stejnou velikost a také formát pixelů. Výsledkem filtrace je obraz, který je reprezentován pixely, kde každý pixel je roven rozdílu hodnot odpovídajících pixelů z původních dvou obrazů. Pokud je rozdíl menší než minimální povolená hodnota, tak je rozdíl nastaven na tuto hodnotu. Třída umí zpracovávat obrazy ve stupních šedi a barevné obrazy (AForge.NET Framework, 2008-2012).

Třída Merge

Třída zpracovává zdrojový obraz a obraz, který je poté překrýván. Tyto obrazy musí mít stejnou velikost a také formát pixelů. Výsledkem filtrace je obraz, který je reprezentován pixely, kde každý pixel je roven větší hodnotě odpovídajících pixelů z původních dvou obrazů. Třída umí zpracovávat obrazy ve stupních šedi a barevné obrazy (AForge.NET Framework, 2008-2012).

Třída YCbCrFiltering

Třída, která slouží pro filtraci pixelů pomocí barevného modelu YCbCr. Tři základní prvky modelu jsou specifikovány pomocí intervalů. Pokud pixel leží v daném intervalu, tak poté je zobrazen beze změn. Ale pokud pixel v daném intervalu neleží, tak poté je jeho barva nahrazena předem nastavenou barvou. Třída umí zpracovávat barevné obrazy, jejichž formát pixelů je buď 24 nebo 32 bitů (AForge.NET Framework, 2008-2012).

Třída HSLFiltering

Třída slouží pro filtrování pixelů v barevném modelu HSL. Základní prvky modelu (Barevný tón, sytost barvy, hodnota jasu) jsou specifikovány pomocí intervalů, které určují, jestli dané pixely budou uvnitř nebo vně intervalu. Pokud pixely neleží uvnitř intervalu, tak jsou poté vyplněny předem nastavenou barvou. Pixely, které leží uvnitř intervalu, jsou zobrazeny beze změn. Třída umí zpracovávat barevné obrazy jejichž formát pixelů je buď 24 nebo 32 bitů (AForge.NET Framework, 2008-2012).

Extrakce objektů BLOB

Metoda, která je používána v analýze obrazu pro detekci regionů v binárním obraze. Pixely, které mají hodnotu 1 a jsou v navzájem propojené oblasti, tak jsou poté označeny stejným indexem. BLOB objekt poté reprezentuje spojitou oblast tvořenou pixely, které mají stejný index. S BLOB objekty lze poté jednoduše manipulovat např. filtrovat, určovat jejich počet, extrahovat pouze největší BLOB objekt atd. BLOB objekty také poskytují informace o své hustotě, svých rozměrech, těžišti atd.

4.4 Knihovna Emgu CV

Otevřená multiplatformní knihovna, která slouží pro zpracování obrazu. Emgu CV poskytuje funkcionalitu knihovny OpenCV pro platformu .NET (Shi, 2013). Největší výhodou knihovny Emgu CV je její velmi vysoký výkon, který vyplívá z testů znázorněných pomocí tabulky 1. Další výhodou je přehledná dokumentace knihovny. Nevýhodou této knihovny je její poměrně složitá implementace do projektu. Největším problémem, který se vyskytl během vývoje aplikace byla práce knihovny s ovládáním kamer tabletu a zpracováním snímků z kamery. Knihovna dokázala pracovat pouze s přední kamerou tabletu, proto byl pro vývoj aplikace použit framework AForge.NET.

Srovnání výkonu daných knihoven

K testování výkonu jednotlivých knihoven jsou vybrány základní funkce zpracování obrazu. Testovanými algoritmy jsou binarizace a převod obrazu do stupňů šedi. Tyto algoritmy jsou vybrány, jako zástupci operací zpracování obrazu, protože velká část zpracování obrazu je založena na čtení paměti, zápisu do paměti a operaci s maticemi (Shi, 2013). Algoritmy byly testovány převážně v programovacím jazyce C#. Pro

tento test byly vybrány metody z knihoven OpenCV, Emgu CV a také metody z frameworku AForge.NET. Rychlost jednotlivých algoritmů je reprezentována v následující tabulce:

Tabulka 1: Porovnání výkonosti jednotlivých knihoven. Zdroj: (Shi, 2013)

Kód	Knihovna	Převod do stupňů šedi [ms]	Binarizace [ms]
C	OpenCV	9.3432	6.9332
C#	AForge.NET	32.6548	19.8743
C#	Emgu CV	11.2369	9.6853
C#	OpenCV (P/Invoke)	10.2355	8.0332
C#	Custom method	33.6742	17.2009

5 Metodika

Cíle, které jsou stanoveny v sekci cíl práce, bude dosaženo pomocí realizace těchto kroků:

1. Pořízení testovacích snímků
2. Návrh algoritmu pro detekci objektu
3. Výpočet těžiště hledaného objektu
4. Experimentální odhad referenčního bodu
5. Návrh algoritmu pro otočení robotu k objektu
6. Návrh trajektorie robotu
7. Testování navrženého algoritmu
8. Implementace vylepšení algoritmu na základě provedených testů

V prvním kroku budou pomocí Webcamery tabletu, umístěného na robotu, pořízeny různé sady testovacích snímků v rozlišení 1280 x 720. Tyto snímky budou odlišeny rozdílným sklonem zrcátka na robotu, pozicí objektu v daném prostředí a také rozdílnými barvami prostředí.

V následujícím kroku budou aplikovány vhodné algoritmy analýzy obrazu na testovací snímky a bude stanovena jejich úspěšnost pro detekování jednotlivých částí obrazu, tj. ohraničení, vnější plocha a vnitřní plocha. Podle těchto získaných informací bude stanoven optimální sklon zrcátka robotu. Poté bude navržen vhodný algoritmus, který bude sestaven z dílčích algoritmů pro detekci jednotlivých částí, který bude detekovat cílový objekt v prostředí. Pak bude navržen algoritmus, který bude sloužit pro výpočet těžiště objektu a určí tento bod v obraze. V příštím kroku bude stanoven referenční bod na základě experimentálního odhadu. Tento bod bude určovat optimální pozici těžiště objektu.

V dalším kroku bude stanoven vztah mezi rovinným úhlem otočení robotu a vzdáleností v pixelech mezi nalezeným těžištěm hledaného objektu a referenčním bodem pomocí experimentálního zjišťování. Tento vztah bude využíván pro otočení robotu a stanovení vzdálenosti robota od hledaného objektu.

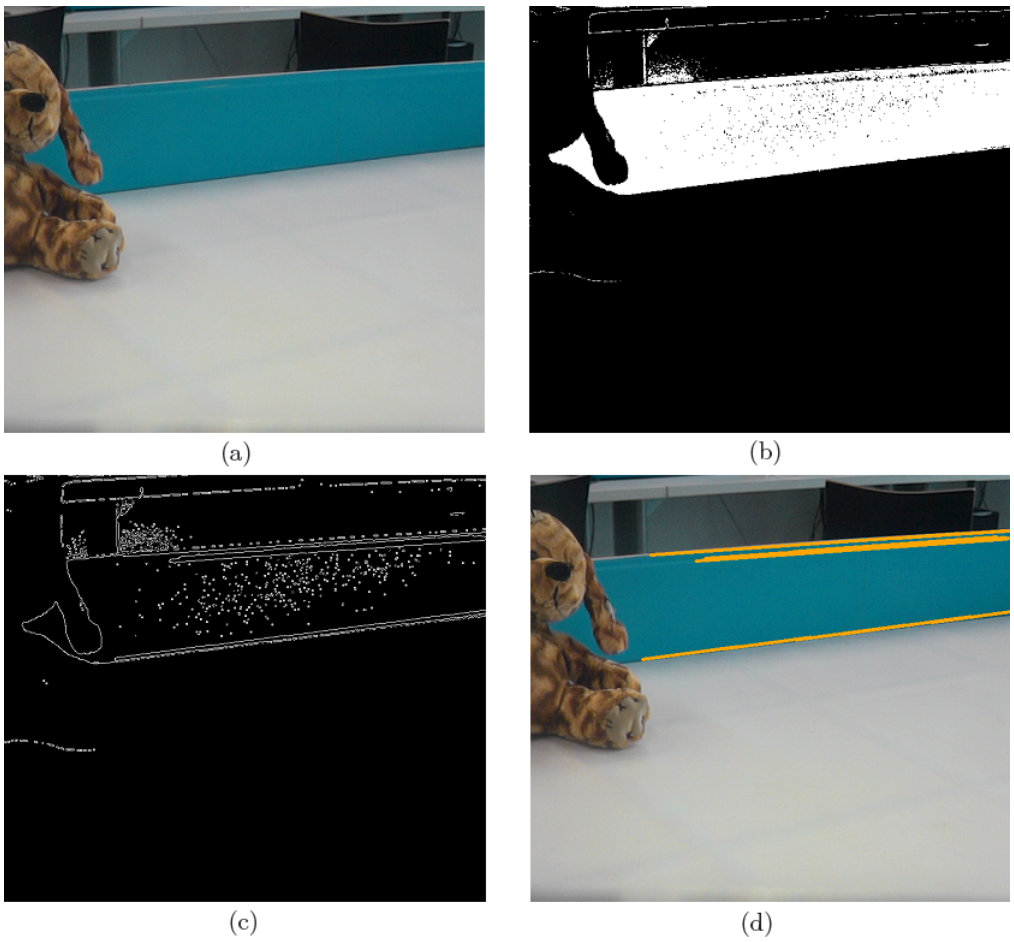
V následujícím kroku bude implementován navržený algoritmus do řídicí jednotky robotu a bude provedeno větší počet testů, které budou ověřovat schopnost nalézt a uchopit různé druhy objektů v daném prostředí za určitý čas. Na základě těchto výsledků bude provedeno vyhodnocení a bude stanovena úspěšnost algoritmu.

6 Praktická část

6.1 Návrh algoritmu pro detekci ohraničení

6.1.1 Řešení založené na Houghově transformaci

Obraz pořízený kamerou robotu je převeden do binárního obrazu pomocí metody prahování s více prahy viz Obr. 9.a. Poté je na obraz aplikován Cannyho hranový detektor, jehož výsledkem je binární obraz, který obsahuje pouze obrysy objektů. Tento obraz je vhodný pro aplikaci metody Houghovi transformace, která v obrazu detekuje přímky. Výsledkem prvního řešení jsou přímky, které reprezentují hrany ohraničení v původním obraze viz Obr.9.d.



Obrázek 9: (a) Výchozí snímek pořízený robotem. (b) Obraz získaný aplikováním metody prahování s více prahy na obraz (a). (c) Obraz, který vzniknul aplikací Cannyho hranového detektoru na obraz (b). (d) Výsledný obraz byl získán aplikací metody Houghovi transformace na obraz (d).

Tabulka 2 reprezentuje výsledky, které byly získány testováním algoritmu založeného na houghově transformaci. První sloupec tabulky vyjadřuje pořadové číslo

Tabulka 2: Hodnoty získané na základě testování řešení podle Houghovi transformace.

Experiment	Ohraničení	Barva ohraničení	Úspěšnost detekce
1	Obě části	Fialová	40%
2	Pouze část	Modrá	85%
3	Celé	Žlutá	80%
4	Obě části	Modrá	45%
5	Část rohu	Žlutá	20%
6	Celé	Fialová	100%
7	Pouze část	Žlutá	90%
8	Obě části	Fialová	60%
9	Část rohu	Modrá	15%
10	Celé	Žlutá	85%

experimentu. V každém experimentu bylo provedeno 20 testů algoritmu podle Houghovi transformace. Výsledkem algoritmu jsou úsečky, které ohraničují dané ohraničení. Pokud jsou úsečky dobře stanoveny a jejich počet je velmi malý, tak poté je test úspěšný. Pokud je detekováno větší množství úseček, úsečky neleží v ohraničení nebo jejich velikost je zásadně menší než délka ohraničení, tak poté je test neúspěšný.

V každém experimentu byl použit rozdílný obraz. Obraz, který obsahuje dvě části, značí ohraničení, které je rozdělené cílovým objektem. Obraz obsahující část ohraničení značí, že robot je velmi blízko k ohraničení nebo ohraničení je přerušeno pozicí cílového objektu. Celé ohraničení vyjadřuje ohraničení, které nemá porušenou svoji plochu. Barva ohraničení v obraze mohla být buď fialová, modrá nebo žlutá. Sloupec úspěšnost detekce zachycuje procentuální hodnotu, která vyjadřuje průměr z výsledků daných testů.

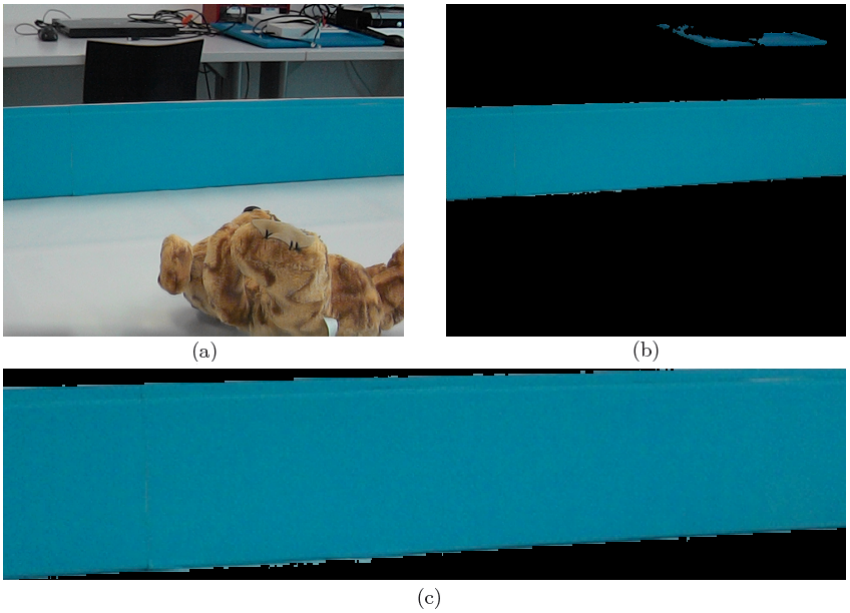
Výsledek řešení

Řešení je vhodné zejména pro nezašuměný a zaostřený obraz. Řešení dosahuje dobrých výsledků u snímků, které obsahují celé ohraničení nebo jeho určitou část. U snímků, které obsahovali ohraničení rozdělené na dvě části dosahovalo řešení průměrných výsledků. U snímků s rohovým ohraničením či jeho částí dosahovala detekce špatných výsledků viz Tab. 2.

6.1.2 Řešení využívající extrakci objektů BLOB

Prvním krokem je aplikace filtru z třídy YCbCrFiltering na snímek, který byl pořízen kamerou robotu. Na základě testování byly určeny optimální parametry filtru, při kterých filtr dosahoval nejlepších výsledků. Obraz s ohraničením, který byl získán aplikací filtru na původní snímek, je převeden pomocí metody prahování na binární obraz viz Obr. 10.b. Na binární obraz je poté aplikována metoda extrahování BLOB objektů s minimální výškou a šířkou 50 pixelů. Výsledek metody je pole spojitých

objektů, které byly nalezeny v obraze s ohraničením viz Obr. 10.c. Nalezené spojité objekty jsou poté vykresleny do bitmapového obrazu ,který představuje výsledné ohraničení.



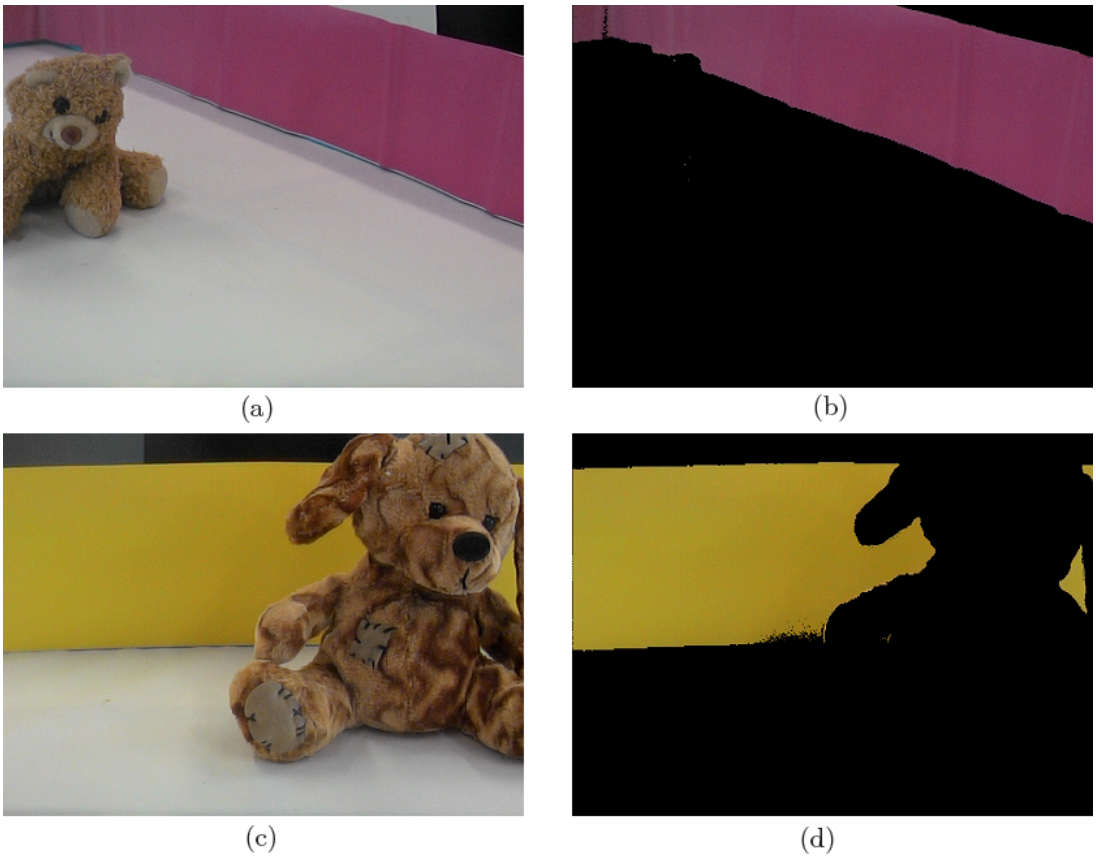
Obrázek 10: (a) Výchozí snímek pořízený robotem. (b) Obraz získaný aplikováním YCbCr filtru na obraz (a). (c) Obraz, který zobrazuje objekt typu BLOB získaný pomocí extrakce objektu typu BLOB z obrazu (c).

Tabulka 3: Hodnoty získané na základě testování řešení podle extrakce objektů BLOB.

Experiment	**Ohraničení**	**Barva ohraničení**	**Úspěšnost detekce**
1	Obě části	Fialová	100%
2	Pouze část	Modrá	100%
3	Celé	Žlutá	100%
4	Obě části	Modrá	100%
5	Část rohu	Žlutá	100%
6	Celé	Fialová	100%
7	Pouze část	Žlutá	100%
8	Obě části	Fialová	100%
9	Část rohu	Modrá	100%
10	Celé	Žlutá	100%

Tabulka 3 reprezentuje výsledky, které byly získány testováním algoritmu založeného na extrakci objektů BLOB. Sloupec experiment vyjadřuje pořadové číslo experimentu. Pro každý experiment bylo provedeno 20 testů daného algoritmu. Výsledkem testu může být buď správně detekované ohraničení nebo ohraničení, které obsahovalo velké množství šumu a nebylo by možné jej použít pro další zpracování.

Pokud je ohraničení správně detekováno, tak poté je ohodnoceno hodnotou jedna, pokud ne, tak je ohodnoceno nulou. V každém experimentu byl testován rozdílný obraz, který byl pořízen kamerou robotu, v různých pozicích robotu a s různou barvou ohraničení. Pro testování byly vybrány tři různorodé barvy ohraničení, tj. fialová, modrá a žlutá barva. Sloupec úspěšnost detekce zachycuje procentuální hodnotu, která vyjadřuje průměr z výsledků daných testů.



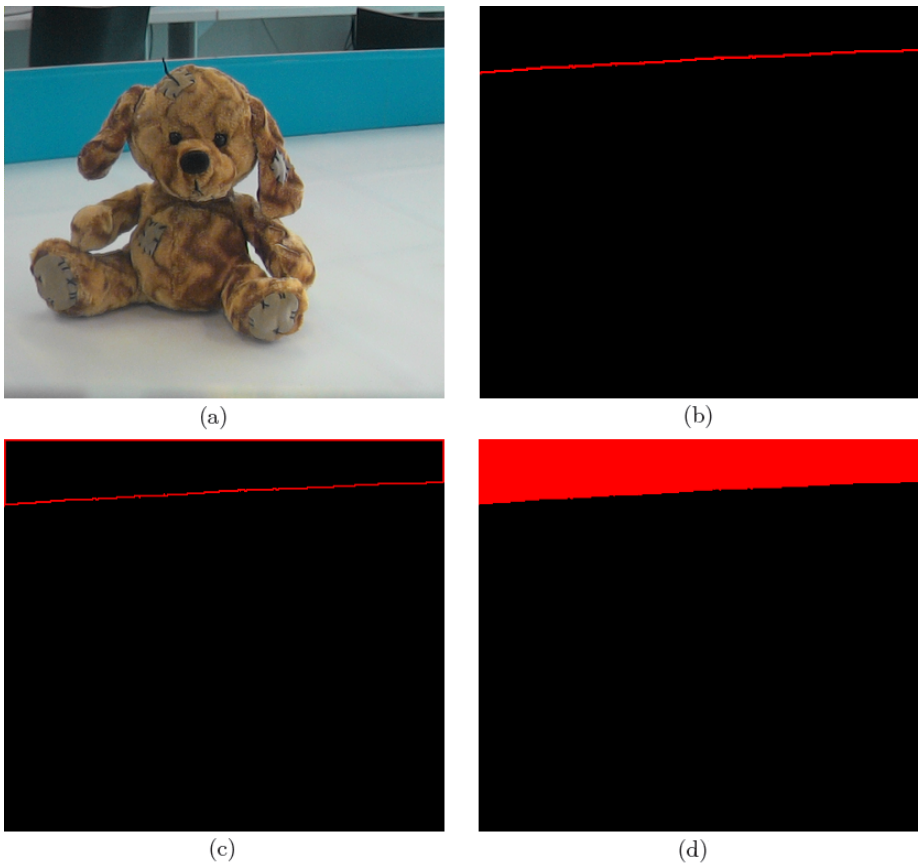
Obrázek 11: (a),(c) Výchozí snímek pořízený robotem. (b),(d) Obraz s ohraničením, který je výsledek druhého řešení algoritmu pro detekci ohraničení.

Výsledek řešení

Výhodou tohoto řešení je extrakce pouze těch objektů, které mají relevantní obsah. Řešení dosahuje velmi dobrých výsledků i u nezaostřených a zašuměných snímků. Ve všech testovaných případech bylo ohraničení vždy přesně detekováno viz Tab. 3. Další výhodou spojenou s použitím objektů BLOB jsou dostupné vlastnosti a funkcionalita, které objekty typu BLOB poskytují např. hustota objektu, plocha objektu, Směrodatná odchylka barev pixelů, Průměr barev pixelů a těžiště objektu.

6.2 Návrh algoritmu pro detekci vnější plochy

Algoritmus pro detekci vnější plochy využívá algoritmus pro detekci ohraničení založený na extrakci objektů BLOB. Výsledkem algoritmu pro detekci ohraničení může být objekt typu BLOB, pokud není ohraničení rozděleno na více částí. Výsledkem může být více objektů BLOB, pokud je ohraničení rozděleno na více částí např. pokud je cílový objekt položen na ohraničení. Výsledkem algoritmu může také být prázdný obraz, v případě, pokud ohraničení není nalezeno v pořízeném snímku.

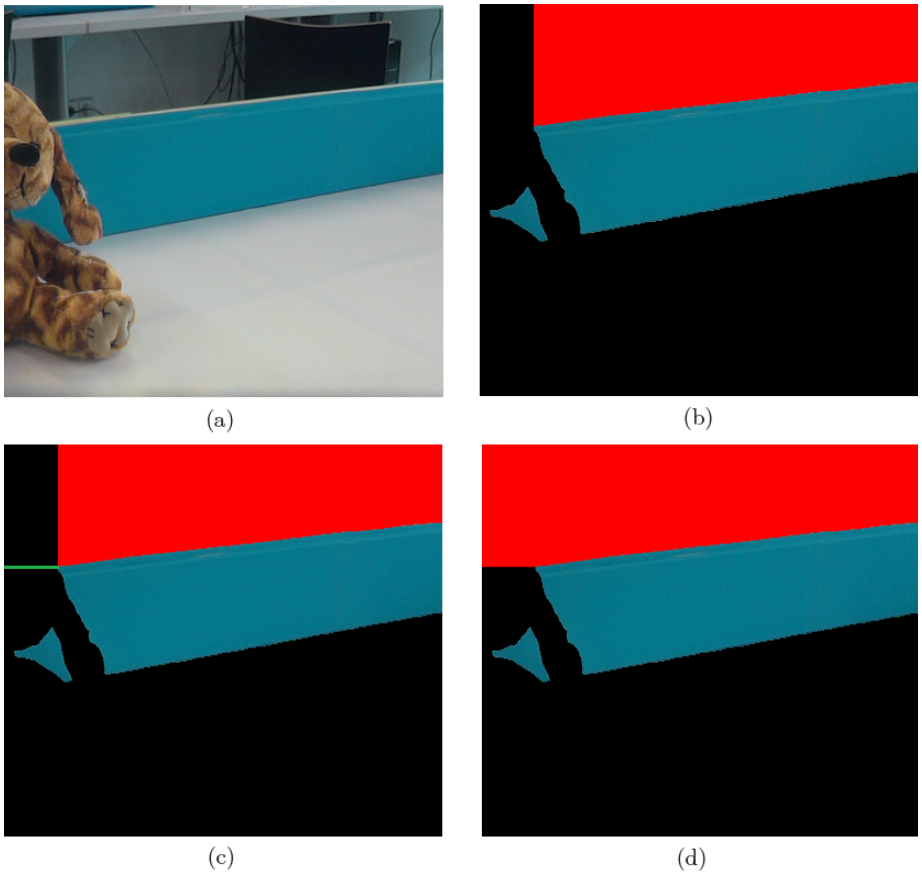


Obrázek 12: (a) Výchozí snímek pořízený robotem. (b) Horní hrana ohraničení. (c) Křivka, která ohraničuje vnější oblast. (d) Oblast reprezentující vnější plochu.

Pokud je detekován alespoň jeden objekt BLOB, tak poté je určena horní hrana objektu či objektů viz Obr. 12.b. Pokud je nalezená hrana spojitá a její délka je stejná jako šířka pořízeného snímku, tak mohou být dopočítány ostatní body vnější oblasti a tím stanovena křivka, která ohraničuje vnější oblast viz Obr. 12.c. Tato křivka je následně vyplněna barvou a tím je získána oblast reprezentující vnější oblast viz Obr. 12.d. V případě, pokud hrana není spojitá nebo její délka nedosahuje šířky pořízeného snímku, tak musí být aproximována.

Aproximace hrany ohraničení

Horní hrana ohraničení musí být aproximována pomocí přímek, pokud hrana ohraničení není spojitá nebo její délka nedosahuje šířky pořízeného snímku viz Obr. 13.b. V prvním kroku je nalezeno přerušení horní hrany ohraničení, které je způsobeno např. umístěním objektu před ohraničení. Objekt může přerušovat ohraničení třemi způsoby.



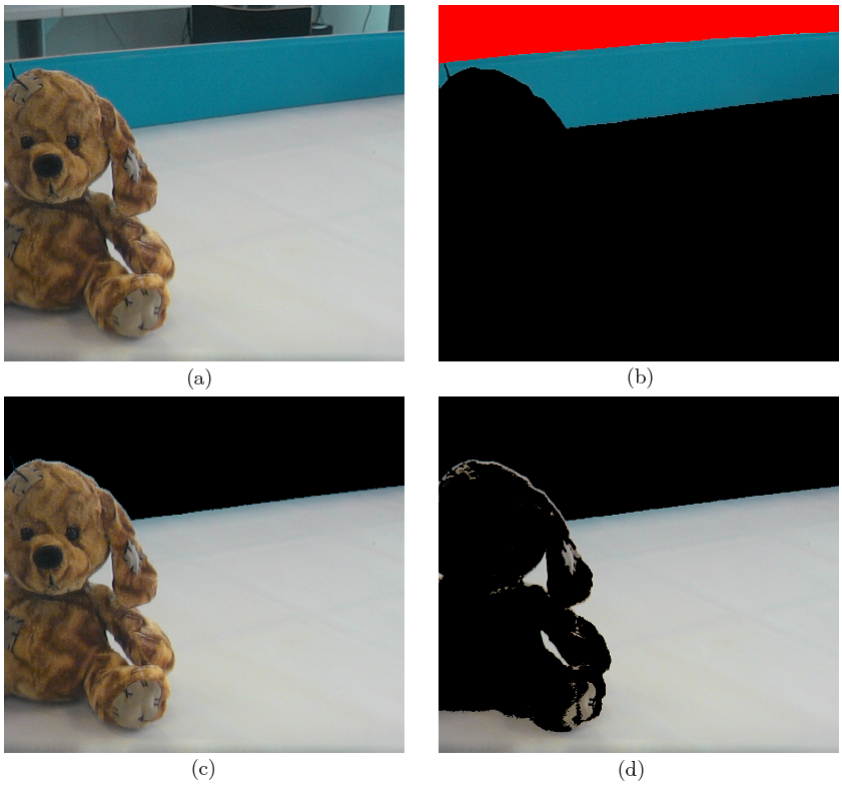
Obrázek 13: (a) Výchozí snímek pořízený robotem. (b) Obraz znázorňující pouze část vnější plochy. (c) Obraz s aproximovanou horní hranou ohraničení. (d) Obraz, který obsahuje výslednou vnější plochu.

Pokud je objekt umístěn na levé straně snímku, tak poté musí být vytvořena přímka mezi levou stranou snímku a prvním bodem částečného ohraničení na ose X viz Obr. 13.c. Je li objekt umístěn takovým způsobem, že je ohraničení rozděleno na dvě části, tak poté musí být vytvořena přímka mezi posledním bodem prvního objektu na ose X a prvním bodem druhého objektu na ose X. Pokud je objekt umístěn napravo, tak musí být vytvořena přímka mezi posledním bodem částečného ohraničení na ose X a pravou stranou snímku. Pokud je částečná hrana ohraničení spojena s novou přímkou, tak vzniká nové ohraničení, které reprezentuje spodní hranu vnější plochy. Poté můžou být dopočteny ostatní body křivky, která ohraničuje vnější oblast. Tato oblast je poté vyplněna barvou viz Obr. 13.d.

Úspěšnost algoritmu pro detekci vnější plochy je závislá na úspěšnosti algoritmu pro detekci ohraničení. Pokud je ohraničení správně detekováno, tak je vnější plocha vždy správně určena.

6.3 Návrh algoritmu pro detekci vnitřní plochy

Na snímek, který byl pořízen robotem pomocí kamery, jsou aplikovány algoritmy pro detekci ohraničení a vnější plochy v obraze. Výsledkem algoritmů mohou být dva nové obrazy obsahující ohraničení a vnější plochu v případě, kdy je robot dostatečně blízko ohraničení a pořízený snímek tedy obsahuje ohraničení a vnější plochu. Dalším výsledkem algoritmu může být jeden obraz s ohraničením, pokud je robot relativně vzdálený od ohraničení a pořízený snímek obsahuje pouze část ohraničení. Třetím výsledkem algoritmů nemusí být žádný obraz, v případě, pokud snímek obsahuje pouze vnitřní plochu.



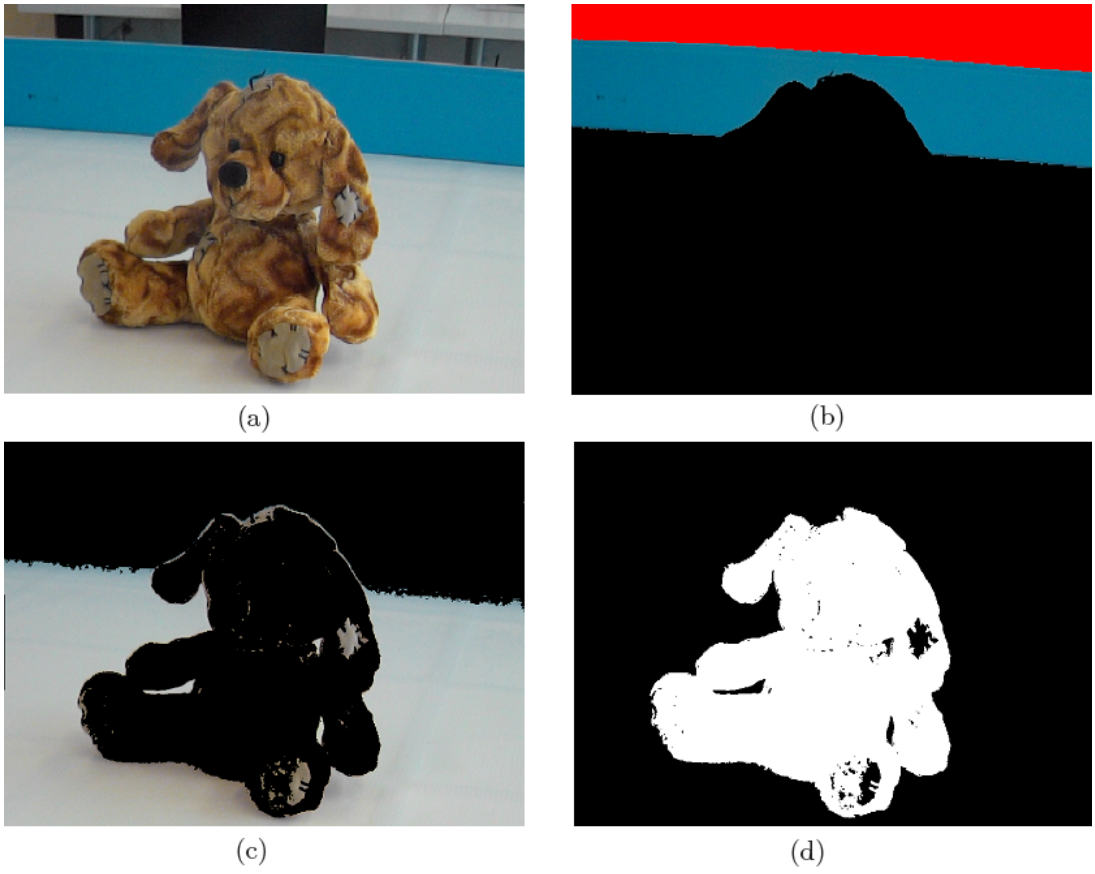
Obrázek 14: (a) Výchozí snímek pořízený robotem. (b) Obraz získaný pomocí filtru Merge spojující obrazy s ohraničením a vnější plochou. (c) Obraz získaný aplikací filtru Substract na obrazy (a), (b). (d) Výsledný obraz, který vzniknul aplikací HSL filtru na obraz (c).

V prvním případě jsou obrazy následně sloučeny do jednoho obrazu pomocí třídy Merge. Výsledkem této operace je tedy obraz s ohraničením, vnější plochou a pozadím, které je reprezentováno černou barvou viz Obr. 14.b. Poté je na nově získaný obraz a původní obraz, který byl pořízen robotem, aplikován filtr ze třídy

Substract, který odečítá hodnoty pixelů daných obrazů a jejímž výsledkem je obraz, který obsahuje vnitřní plochu a hledaný objekt viz Obr. 14.c. Pro oddělení hledaného objektu a vnitřní plochy je použit filtr ze třídy HSLFiltering. viz Obr. 14.d. Výsledkem detekce je tedy bitmapový obraz reprezentující vnitřní plochu snímku.

6.4 Návrh algoritmu pro detekci cílového objektu

Pro detekci objektu v obraze, jsou na pořízený snímek postupně aplikovány algoritmy pro detekci ohraničení, vnější plochy a vnitřní plochy viz Obr. 15.c. Výsledkem těchto algoritmů jsou tedy tři obrazy, které obsahují ohraničení, vnější plochu a vnitřní plochu (obrazy, které obsahují ohraničení a vnější plochu mohou být prázdné v případě, že snímek neobsahoval ohraničení respektive vnější plochu).



Obrázek 15: (a) Výchozí snímek pořízený robotem. (b) Obraz získaný algoritmem pro detekci ohraničení a vnější plochy. (c) Obraz získaný algoritmem pro detekci vnější plochy. (d) Výsledný obraz, který obsahuje cílový objekt.

Na tyto získané obrazy je aplikován filtr ze třídy Merge, který spojí ohraničení a vnější plochu do jednoho obrazu (viz Obr.15.b) a poté spojí vnitřní plochu s obrazem obsahujícím ohraničení a vnější plochu. Takto získaný obraz je převeden pomocí metody prahování na binární obraz, který je poté pomocí filtru invertován.

Výsledný obraz tedy obsahuje cílový objekt, který je reprezentován bílou barvou a pozadí, které je znázorněno černou barvou viz Obr. 15.d. Výhodou tohoto řešení je jeho univerzálnost protože algoritmus je schopen detekovat objekt s jakýmkoli tvarem či barevnou skladbou viz Obr. 16.

Fáze algoritmu:

1. Pořízení snímku kamerou robotu
2. Detekce ohraničení
3. Detekce vnější plochy
4. Detekce vnitřní plochy
5. Spojení obrazů do jednoho
6. Převod na binární obraz
7. Invertace obrazu

Tabulka 4: Hodnoty získané na základě testování algoritmu pro detekci cílového objektu.

	Objekt	Pozice	Osvětlení	Úspěšnost
1	Robot	V rohu	Osvětlené	85%
2	Velký medvěd	Ve středu plochy	Osvětlené	100%
3	Malý medvěd	Před ohraničením	Neosvětlené	90%
4	Velký medvěd	V rohu	Neosvětlené	80%
5	Robot	Před ohraničením	Osvětlené	95%
6	Velký medvěd	U robotu	Neosvětlené	95%

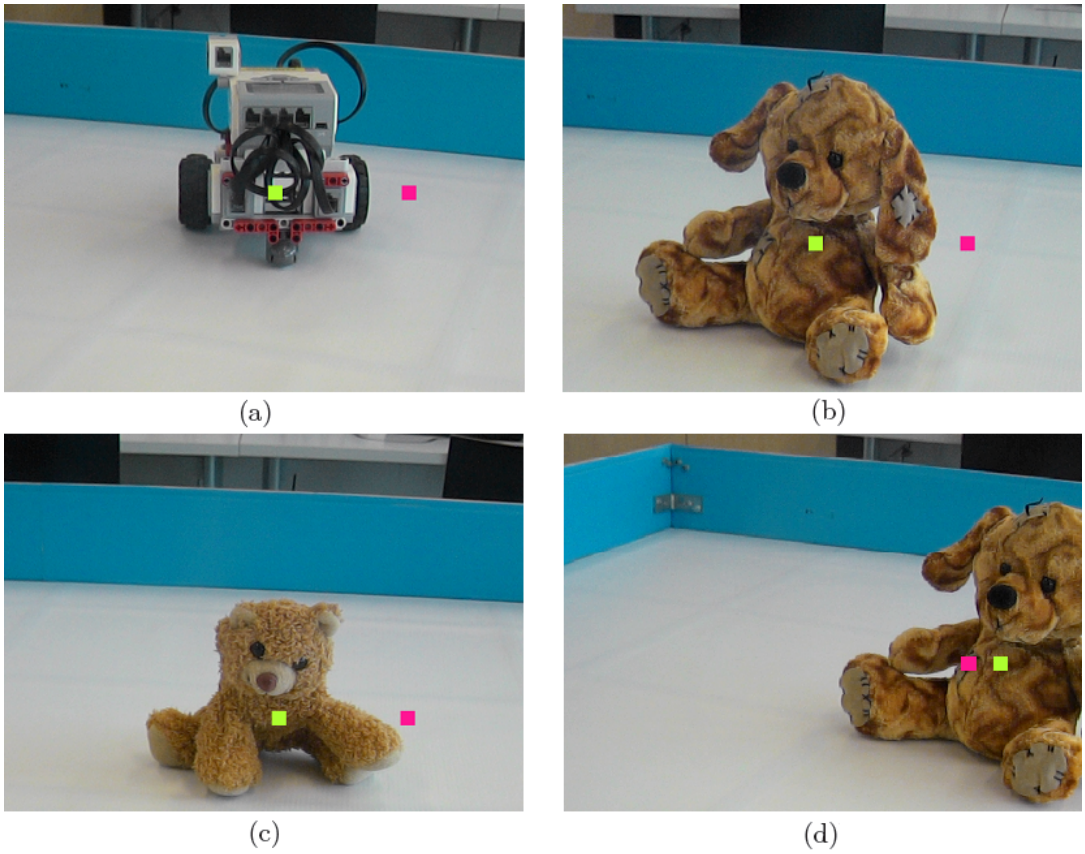
Tabulka 4 reprezentuje výsledky, které byly získány testováním algoritmu pro detekci cílového objektu. První sloupec udává pořadové číslo experimentu. Pro každý experiment bylo provedeno 15 testů. Pokud byl cílový objekt algoritmem správně detekován, tak výsledkem testu je hodnota 1. Pokud výsledný obraz obsahoval velké množství šumu, tak poté objekt nebyl správně detekován a výsledná hodnota je nula.

V experimentech byly testovány tři různorodé objekty, tj. velký medvěd, robot, malý medvěd. Pozice objektů v prostředí se měnila podle daného experimentu. Sloupec osvětlení udává zda bylo prostředí, ve kterém se nachází cílový objekt, osvětleno či ne. Výsledek experimentu je uveden ve sloupci úspěšnost, jehož hodnota udává průměr z jednotlivých testů algoritmu.

6.4.1 Nalezení objektu a stanovení těžiště

V prvním kroku je na pořízený snímek aplikován algoritmus pro detekci cílového objektu, jehož výsledkem je obraz, který obsahuje cílový objekt. Na tento obraz je

poté aplikována metoda extrahování největšího objektu typu BLOB, která v obraze vyhledává největší spojitý objekt, který má alespoň šířku 50 pixelů a výšku 50 pixelů. Pokud objekt není nalezen nebo objekt není dostatečně velký, tak algoritmus vyhodnotí obraz jako prázdný, tj. nulový počet nalezených objektů.



Obrázek 16: Stanovení těžiště a referenčního bodu u různých objektů v rozdílných pozicích.

Pokud je objekt nalezen, tak poté je porovnána jeho hustota, což je vlastnost objektu typu BLOB, která je spočítána podle matematického vzorce:

$$\text{hustota objektu} = \frac{\text{počet pixelů objektu}}{\text{šířka objektu} \cdot \text{výška objektu}} \tag{14}$$

Hustota může nabývat hodnoty od 0 do 1, která udává procentuální hodnotu vyplnění objektu. Pokud je hustota menší než 20%, tak algoritmus stanoví, že obraz je prázdný, tj. nulový počet nalezených objektů. Pokud je hustota větší než 20%, tak poté je stanoveno těžiště objektu BLOB pomocí vlastnosti CenterOfGravity viz Obr. 16.

6.4.2 Stanovení referenčního bodu

Referenční bod je bod, jehož souřadnice X odpovídá optimální pozici těžiště objektu. Pokud se X souřadnice těžiště nachází na levé straně od X souřadnice referenčního bodu, tak poté se robot musí otočit směrem doleva, aby se těžiště objektu dostalo na hranici referenčního bodu, a naopak pokud se X souřadnice těžiště nachází na pravé straně od referenčního bodu, tak poté se musí robot otočit doprava.

Referenční bod byl stanoven na základě testování, při kterém byla sledována X souřadnice těžiště a optimální natočení robotu směrem k objektu.

6.5 Návrh algoritmu pro otočení robotu k objektu

Otočení robotu k těžišti cílového objektu je založeno na přesné detekci objektu ve snímku. Pokud je objekt v obraze detekován, tak poté je stanoveno jeho těžiště v obraze. Vzdálenost mezi těžištěm a referenčním bodem na ose X udává počet pixelů, o kolik se musí robot otočit, aby byl namířen přímo na cílový objekt.

6.5.1 Řešení využívající analytický výpočet

Toto řešení stanovuje vztah mezi pixely a úhlem otočení. Poté lze převést vzdálenost mezi těžištěm a referenčním bodem, které jsou v jednotkách pixelů, na jednotky rovinného úhlu. V prvním kroku je provedena detekce objektu a určení jeho těžiště. Poté je robot otočen o 5° směrem k referenčnímu bodu a následně je provedena druhá detekce objektu a určení jeho těžiště. Vzdálenost mezi dvěma těžišti vznikla otočením o 5° a je v jednotkách pixelů. Tímto lze stanovit počet pixelů na 1° a tím určit výsledný úhel otočení. Vztah je matematicky vyjádřen pomocí:

$$\alpha = |x_R - X_{M2}| \cdot \frac{\Delta}{|x_{M2} - x_{M1}|} \tag{15}$$

Kde α udává výsledný úhel otočení. Proměnná x_{M2} udává souřadnici X těžiště objektu, který byl detekován po otočení robotu o 5°. Proměnná x_{M1} udává souřadnici X těžiště objektu před otočením robotu o 5°. Otočení robotu o 5° je v rovnici vyjádřeno pomocí hodnoty Δ. Proměnná x_R udává souřadnici X referenčního bodu, který určuje optimální pozici těžiště objektu.

Tabulka 5: Hodnoty získané na základě testování řešení využívající analytický výpočet

Experiment	1	2	3	4	5	6	7	8	9	10
Skutečný úhel	21°	18°	11°	27°	7°	28°	8°	21°	9°	24°
Optimální úhel	35°	21°	19°	21°	15°	14°	17°	12°	4°	18°
Rozdíl úhlů	14°	3°	8°	6°	8°	14°	9°	9°	5°	6°

Tabulka 5 reprezentuje výsledky, které byly získány testováním algoritmu založeného na principu analytického výpočtu. Řešení využívá vztah mezi pixely a úhlem otočení. Pro každý experiment byl proveden jeden test, který testuje úspěšnost daného algoritmu.

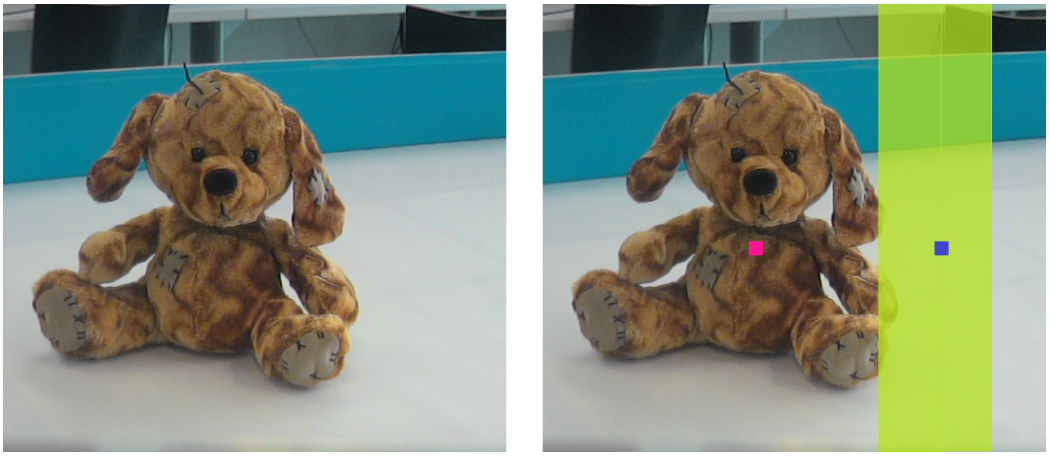
Pro každý test byla zaznamenána hodnota úhlu otočení, tj. o kolik se robot otočil směrem k objektu. Tato hodnota je v tabulce pojmenována skutečný úhel. Řádek s hodnotami optimálních úhlů určuje úhel, o kolik se měl robot otočit, aby byl přesně natočen na daný objekt. Pokud je skutečný úhel roven optimálnímu, tak je úspěšnost algoritmu 100%. Čím větší je absolutní hodnota rozdílu optimálního a skutečného úhlu, tak tím klesá úspěšnost daného algoritmu. Přesnost naměřených hodnot je $\pm 1°$. Vzhledem k rozdílům mezi skutečným a optimálním úhlem je přesnost měření dostatečná.

Výsledek řešení

Výhodou tohoto řešení je velmi nízký počet detekcí objektu. Stačí pouze jedna detekce objektu k určení velikosti jednoho stupně úhlu v pixelech a následného otočení robotu k objektu. Nevýhodou řešení je nepřesnost stanoveného úhlu vzhledem k optimálnímu úhlu, které je způsobeno deformací obrazu.

6.5.2 Iterativní numerické řešení

Druhé řešení je založeno na postupném otáčení robotu o konstantní velikost úhlu 5° a stanovení dvou konstantních intervalů, jejichž počátkem je referenční bod. První interval stanovuje vzdálenost, jdoucí směrem doprava od referenčního bodu po ose X a druhý interval určuje vzdálenost, jdoucí směrem doleva od daného bodu po ose X. Tímto je stanovena oblast okolo referenčního bodu, která je použita pro ověření, jestli je nalezené těžiště detekovaného objektu dostatečně blízko k referenčního bodu.



Obrázek 17: Pořízený snímek pomocí kamery a snímek po detekci těžiště, určení referenčního bodu a stanovení oblasti okolo referenčního bodu.

Algoritmus 1: Natočení robotu k těžišti cílového objektu

Input: Úhel otočení , pravý interval, levý interval, referenční bod
Output: Robot je natočen k těžišti objektu

```
1  objekt ← DetekceObjektu()
2  teziste ← VratTezisteObjektu(objekt)
3  if teziste.X > referencniBod.X then
4      while teziste.X > (referencniBod.X + pravyInterval) do
5          OtocitRobota(uhelOtoceni, DOPRAVA)
6          objekt ← DetekceObjektu()
7          teziste ← VratTezisteObjektu(objekt)
8  else
9      while teziste.X < (referencniBod.X + levyInterval) do
10         OtocitRobota(uhelOtoceni, DOLEVA)
11         objekt ← DetekceObjektu()
12         teziste ← VratTezisteObjektu(objekt)
13     PresunRobotu(10, DOPREDU)
```

V prvním kroku se provede detekce objektu, následně se stanoví těžiště objektu a poté vzdálenost mezi těžištěm a referenčním bodem. Pokud se těžiště nachází ve stanovené oblasti okolo referenčního bodu, tak se robot neotáčí, jelikož je těžiště dostatečně blízko k referenčnímu bodu. Ale pokud se těžiště nenachází v dané oblasti, tak je provedeno otočení robotu o 5°.

Směr otočení je určen podle polohy těžiště. Pokud je X souřadnice těžiště větší, než X souřadnice referenčního bodu a jejich rozdíl je větší než pravý konstantní interval, poté je stanoveno otočení směrem doprava. Pokud je souřadnice X těžiště menší než X souřadnice referenčního bodu a jejich rozdíl v absolutní hodnotě je větší, než levý konstantní interval, poté je směr otočení doleva. Tento krok je prováděn, dokud se nenachází těžiště objektu v oblasti referenčního bodu.

Tabulka 6: Hodnoty získané testováním postupu iterativního numerického řešení.

Experiment	1	2	3	4	5	6	7	8	9	10
Skutečný úhel	33°	25°	8°	12°	31°	12°	16°	12°	32°	29°
Optimální úhel	35°	27°	8°	14°	29°	11°	16°	13°	31°	30°
Rozdíl úhlů	2°	2°	0°	2°	2°	1°	0°	1°	1°	1°

Tabulka 6 reprezentuje výsledky, které byly získány testováním algoritmu založeného na principu iterativního numerického řešení. Řešení využívá přípustnou oblast okolo referenčního bodu. Levý interval od referenčního bodu má velikost 90 pixelů a velikost pravého intervalu je 70 pixelů. Pokud se tedy nachází těžiště v daných intervalech, tak je řešení stále přípustné.

Pro každý experiment byl proveden jeden test, který testuje úspěšnost daného algoritmu. Pro každý test byla zaznamenána hodnota úhlu otočení tj. o kolik se robot otočil směrem k objektu. Tato hodnota je v tabulce pojmenována skutečný úhel. Přesnost naměřených hodnot je $\pm 1°$. Experimentální testy tohoto řešení prokázali jeho funkčnost viz kapitola Shrnutí. Řádek s hodnotami optimálních úhlů určuje úhel, o kolik se měl robot otočit, aby byl přesně natočen na daný objekt. Pokud je skutečný úhel roven optimálnímu, tak je úspěšnost algoritmu 100%. Čím větší je absolutní hodnota rozdílu optimálního a skutečného úhlu, tak tím klesá úspěšnost daného algoritmu.

Výsledek řešení

Nevýhodou je větší počet detekcí objektu a tím větší pravděpodobnost nesprávně detekovaného objektu. Výhodou je přesnější stanovení úhlu vzhledem k optimálnímu úhlu a také už není potřeba stanovit závislost mezi vzdáleností v pixelech a úhlem otočení.

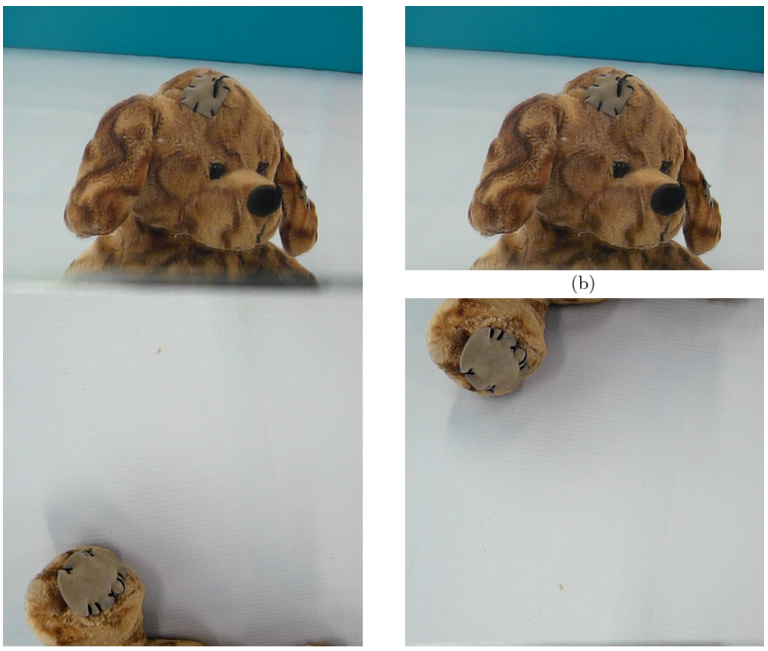
6.6 Plánování trajektorie

První krok plánování je pořízení snímku, který je rozdělen na dvě části z důvodu upevněného zrcátka pod kamerou robotu, které slouží pro zvětšení zorného pole kamery robotu viz Obr. 18.a. Tyto dvě části snímku jsou odděleny podle hrany zrcátka v pořízeném snímku a jsou poté reprezentovány pomocí bitmapových obrazů viz Obr. 18.b a 18.c.

Části snímku jsou mimo jiné používány pro odhad vzdálenosti objektu od robotu. Pokud je objekt dostatečně blízko robotu, tak poté je podstatná část objektu zachycena ve spodní části snímku. Ale pokud je objekt vzdálený od robotu, tak je jeho značná část zobrazena v horní části snímku.

V druhém kroku je provedena detekce objektu ve spodní a horní části snímku. Pokud je v horní části snímku detekován objekt a zároveň ve spodní části není objekt detekován, nebo jeho těžiště nedosahuje určité výšky, tak poté je použito iterativní numerické řešení pro otočení robotu směrem k objektu. Robot je tedy správně natočen k cílovému objektu a je provedeno popojetí robotu k cílovému objektu o konstantní vzdálenost. Tento krok je opakován, dokud není ve spodní části snímku detekován objekt, jehož těžiště dosahuje určité výšky vzhledem k výšce snímku viz Obr. 19.c. Pokud výška těžiště dosahuje této výšky, tak je zřejmé, že pozice robotu je velmi blízko k hledanému objektu a robot je částečně natočen na cílový objekt.

V třetím kroku je provedeno konečné otočení robotu k objektu pomocí iterativního numerického řešení. Tímto je robot téměř přesně natočen k objektu a může být provedeno popojetí robotu o konstantní vzdálenost směrem k objektu a také aktivování funkce pro uchopení objektu. Algoritmus je znázorněn pomocí vývojového diagramu viz Obr. 21.



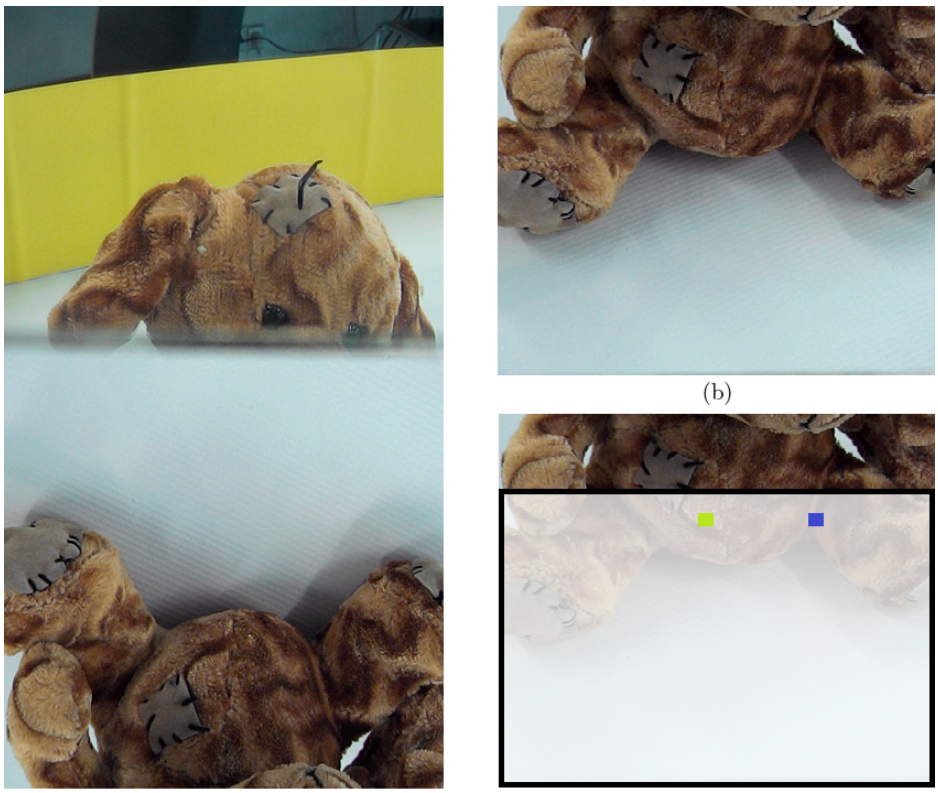
Obrázek 18: (a) Výchozí snímek pořízený robotem. (b) Obraz reprezentující horní část obrazu. (c) Obraz reprezentující horní spodní obrazu. Obraz je otočen o 180°.

Tabulka 7 reprezentuje výsledky, které byly získány testováním algoritmu pro plánování trajektorie. První sloupec udává číslo experimentu. Pro každý experiment bylo provedeno 10 testů. Výsledkem testu může být hodnota 1, pokud byl objekt nalezen a uchopen, nebo nula v případě nenalezení objektu nebo špatného uchopení.

Sloupec objekt určuje typ objektu, který byl vyhledáván robotem v dané scéně. Další sloupec udává pozici objektu v rámci daného experimentu. Sloupec osvětlení určuje, zda byla scéna v rámci experimentu osvětlena či ne. Sloupec úspěšnost obsahuje průměr z provedených testů v rámci experimentu. Poslední sloupec udává průměrný čas, v kterém byl objekt nalezen a uchopen. Sloupec čas obsahuje také směrodatnou odchylku, která určuje variabilitujednotlivých testů.

Tabulka 7: Hodnoty získané na základě testování algoritmu pro naplánování trajektorie.

	Objekt	Pozice	Osvětlení	Úspěšnost	Čas [s]
1	Robot	V rohu	Ano	80%	11.5 ±4.9
2	Velký medvěd	Ve středu plochy	Ano	100%	12.7 ±5.2
3	Malý medvěd	Před ohraničením	Ne	90%	22.3 ±2.7
4	Velký medvěd	V rohu	Ne	90%	15.7 ±4.8
5	Robot	Před ohraničením	Ano	90%	18.5 ±2.8
6	Velký medvěd	U robotu	Ne	100%	14.2 ±3.7



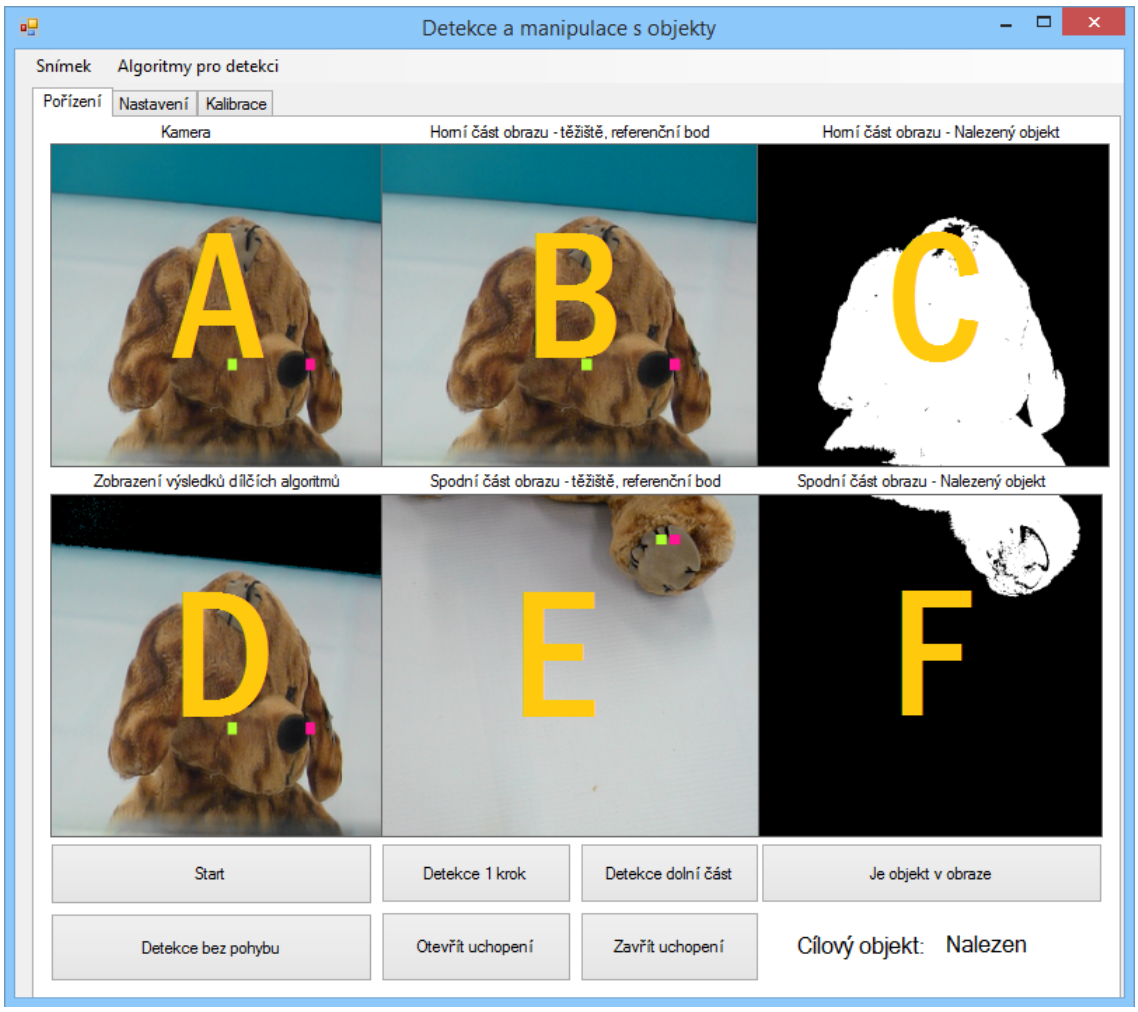
(a) (b) (c)

Obrázek 19: (a) Výchozí snímek pořízený robotem. (b) Obraz reprezentující spodní část obrazu otočenou o 180°. (c) Obraz znázorňující těžiště objektu, ref.bod a vyznačenou oblast, která reprezentuje přípustnou polohu těžiště.

6.7 Testování

Pro testování algoritmu byla vytvořena aplikace s grafickým rozhraním viz Obr. 20. Aplikace obsahuje šest grafických prvků, které zachycují výsledky dílčích algoritmů. První prvek slouží pro zobrazení snímku, který byl pořízen kamerou robotu viz Obr. 20.A. Další prvek je určen pro zobrazení horní části snímku, těžiště nalezeného objektu a referenčního bodu viz Obr. 20.B. Pomocí daného prvku lze ověřit, jestli bylo těžiště určeno správně vzhledem k objektu. Třetí prvek zachycuje nalezený cílový objekt pomocí binárního obrazu viz Obr. 20.C. Z třetího prvku lze jednoduše zjistit, zda byl cílový objekt detekován správně. Čtvrtý prvek je určen pro testování dílčích algoritmů detekce např. detekce ohraničení, detekce vnější plochy a aproximace horní hrany ohraničení. Pátý prvek obsahuje spodní část snímku, který byl pořízen robotem a také obsahuje těžiště objektu a referenční bod viz Obr. 20.E. Poslední prvek obsahuje binární obraz, který zobrazuje nalezený objekt ve spodní částí snímku viz Obr. 20.F.

Aplikace také poskytuje nastavení, které umožňuje nastavovat vstupní parametry daných algoritmů a tím experimentálně testovat optimální parametry daných algoritmů. Aplikace poskytuje informace o počtu nalezených objektů v daném snímku,



Obrázek 20: Aplikace s grafickým uživatelským rozhraním, která slouží pro testování dílčích algoritmů pro detekci objektu.

hustotě objektů, těžišti cílového objektu a vzdálenosti těžiště od referenčního bodu v jednotkách pixelů. Aplikace poskytuje tyto funkce:

- **Spuštění algoritmu pro nalezení a uchopení objektu**
- **Detekce objektu v horní části snímku bez pohybu robotu**
- **Detekce objektu v horní části snímku pouze s jedním otočením k objektu**
- **Detekce objektu ve spodní části bez pohybu robotu**
- **Ověření, zda je objekt ve scéně**
- **Otevření uchopení**
- **Zavření uchopení**

- **Kalibrace robotu**
- **Ukládání a načítání obsahu jednotlivých prvků**
- **Nastavení kamer daného zařízení**

7 Zhodnocení

7.1 Shrnutí

Cílem práce byl návrh algoritmu pro detekci objektu ve scéně. Algoritmus je implementován do řídicí jednotky robotu K3. Robot K3 je schopný najít a uchopit určitý objekt v předem známém prostředí. Práce je rozdělena do sedmi kapitol.

První kapitola definuje cíl práce a její úvod. Druhá kapitola se soustřeďuje na reprezentaci digitálního obrazu v počítači, základní úlohy analýzy obrazu a také na vybrané metody segmentace obrazu, které jsou poté použity v praktické části práce.

Třetí kapitola se zaměřuje na problematiku autonomních robotů a na jejich základní úlohy. V této části jsou také uvedeny základní informace o robotu K3 a jeho dílčích částech. Další kapitola popisuje použité vývojové prostředky po teoretické stránce. Tato část také zhodnocuje dostupné knihovny analýzy obrazu pro programovací jazyk C#.

Pátá kapitola obsahuje postup práce, který vedl ke splnění cíle práce.

Šestá kapitola obsahuje praktickou část práce a je rozdělena do několika podkapitol. První podkapitola obsahuje návrh dvou řešení algoritmů pro detekci ohraničení. V této části je také provedeno testování daných řešení a jejich následné vyhodnocení. Následující podkapitola je zaměřena na návrh algoritmu pro detekci vnější části obrazu a také navrhuje řešení pro aproximaci hrany ohraničení. Následně je proveden test daného řešení a jeho vyhodnocení. Třetí podkapitola popisuje návrh algoritmu pro detekci vnitřní plochy.

Další podkapitola je zaměřena na návrh algoritmu pro detekci cílového objektu v prostředí. Návrh využívá algoritmy z předchozích podkapitol. V této podkapitole je provedeno otestování daného řešení za různých podmínek a poté je řešení vyhodnoceno. Podkapitola se také zabývá stanovením těžiště nalezeného objektu a určením referenčního bodu, tj. optimální pozice těžiště v obraze.

Další podkapitola řeší problematiku otočení robotu směrem k objektu. Jsou zde uvedeny dva návrhy řešení, které jsou otestovány a poté je stanovena jejich úspěšnost. Následující podkapitola je nejdůležitější částí práce, která je zaměřena na návrh algoritmu pro plánování trajektorie od robotu k cílovému objektu v daném prostředí. Poté je provedeno testování daného algoritmu za různých podmínek a jeho vyhodnocení.

V kapitole testování jsou popsány jednotlivé části aplikace s grafickým rozhraním, která byla vytvořena za účelem testování algoritmu.

7.2 Diskuze

Algoritmus pro detekci objektu ve scéně je rozdělen do několika dílčích algoritmů. Pro detekci ohraničení je použit algoritmus založený na extrakci objektů typu BLOB. Tento algoritmus detekuje ohraničení se stoprocentní úspěšností. Pro detekci vnější plochy je použit algoritmus, který využívá algoritmus pro detekci ohraničení. Úspěš-

nost algoritmu je téměř stoprocentní. Algoritmus pro detekci vnitřní plochy využívá předchozích algoritmů pro detekci ohraničení a vnější plochy. Na výsledky těchto algoritmů jsou aplikovány filtry, jejichž výsledkem je binární obraz, který reprezentuje vnitřní plochu.

Algoritmus pro detekci objektu tedy postupně detekuje ohraničení, vnější a vnitřní plochu. Tyto tři části obrazu jsou poté spojeny do jednoho obrazu a následně je na nový obraz a původní obraz pořízený robotem aplikován filtr Substract. Výsledkem algoritmu je binární obraz, který reprezentuje cílový objekt. Úspěšnost algoritmu je závislá na podmínkách, zda je scéna osvětlena, jaká je pozice objektu, jaký objekt je hledán. Úspěšnost se pohybuje v rozmezí 80% až 100%.

Pro otočení robotu směrem k těžišti objektu byl vybrán algoritmus, který je založen na iterativním numerickém řešení. Tento algoritmus určuje přípustnou oblast okolo referenční bodu, v kterém se dané těžiště musí nacházet. Navržený algoritmus je schopen zajistit otočení s přesností $\pm 2°$, což je dostačující, jak prokázalo testování.

Algoritmus pro plánování trajektorie k cílovému objektu je založen na předchozích algoritmech. V první fázi je detekován objekt a jeho těžiště. V druhém kroku je provedeno otočení robotu podle iterativního numerického řešení a popojetí robotu směrem k objektu. Tento krok je opakován, dokud není robot dostatečně blízko objektu. Poté robot popojede o konstantní vzdálenost směrem k objektu a aktivuje funkci pro uchopení objektu.

Úspěšnost algoritmu se pohybuje od 80% do 100%. Algoritmus dosahuje úspěšnosti 80% při nepříznivých podmínkách. Úspěšnosti 100% je dosáhnuto pouze za optimálních podmínek.

7.3 Závěr

Cílem práce byl návrh algoritmu pro detekci cílového objektu v prostředí a jeho implementace do řídicí jednotky autonomního robotu K3. Úspěšnost algoritmu pro nalezení daného objektu v prostředí se pohybuje od 80% do 100% podle daných podmínek. Řešení je schopné nalézt objekty, které se mohou lišit určitou vlastností např. (barvou, velikostí, tvarem), v prostředí s proměnlivou barvou ohraničení a plochou. Z daného výsledku vyplývá, že cíl práce byl splněn.

8 Literatura

AForge.NET Framework. 2008-2012. *AForge.NET :: Computer Vision, Artificial Intelligence, Robotics* [online]. [cit. 2015-03-10]. Dostupné z: `http://www.aforgenet.com/framework/`.

SIEGWART, Roland. *Introduction to autonomous mobile robots. 1. vyd.* Massachusetts: MIT Press, 2004, 321 s. ISBN 02-621-9502-X.

SHI, Shin. *Emgu CV essentials: develop your own computer vision application using the power of Emgu CV*. Birmingham, UK: Packt Publishing, 2013. ISBN 978-1-78355-952-7.

Canny edge detector demos. 1996. *Berkeley Robotics and Intelligent Machines Lab* [online]. [cit. 2015-05-17]. Dostupné z: `http://robotics.eecs.berkeley.edu/~sastry/ee20/` .

ŽÁRA, Jiří, Bedřich BENEŠ a Petr FELKEL. *Moderní počítačová grafika. Vyd. 1.* Praha: Computer Press, 1998, xvi, 448 s. ISBN 80-722-6049-9.

.NET Framework 4.5. 2015. *MSDN* [online]. [cit. 2015-03-30]. Dostupné z: `https://msdn.microsoft.com/en-us/library/zw4w595w.aspx`.

SHARP, John. *Microsoft Visual C# 2010: krok za krokem*. Vyd. 1. Brno: Computer Press, 2010, 696 s. Krok za krokem (Computer Press). ISBN 978-80-251-3147-3.

BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, 2008, xvii, 555 s. ISBN 978-0-596-51613-0.

Docs OpenCV. 2015. *Hough Line Transform* [online]. [cit. 2015-03-31]. Dostupné z: `http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html`.

WALEK, Petr, Martin LAMOŠ a Jiří JAN. *Analýza biomedicínských obrazů: Počítačová cvičení* [online]. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2013 [cit. 2015-04-10]. ISBN 978-80-214-4792-9. Dostupné z: `www.dbme.feec.vutbr.cz/sites/default/files/news/fabo.pdf`.

PRATT, William K. *Digital Image Processing: PIKS Inside, 3rd Edition*. New York: John Wiley & Sons, Inc., 2001. ISBN 0-471-37407-5.

ŠPANĚL, Michal a Vítězslav BERAN. *Obrazové segmentační techniky: Přehled existujících metod* [online]. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav počítačové grafiky a multimédií, 2005 [cit. 2015-04-10]. Dostupné z: `http://www.fit.vutbr.cz/~spanel/segmentace/`.

Bodové jasové transformace. 2015. *Katedra kybernetiky ZČU* [online]. [cit. 2015-04-10]. Dostupné z: http://www.kky.zcu.cz/cs/courses/zdo/lesson2.

JÄHNE, Bernd. *Digital image processing: 5th revised and extended edition*. Vyd. 1. Berlin: Springer, 2002, 585 s. ISBN 35-406-7754-2.

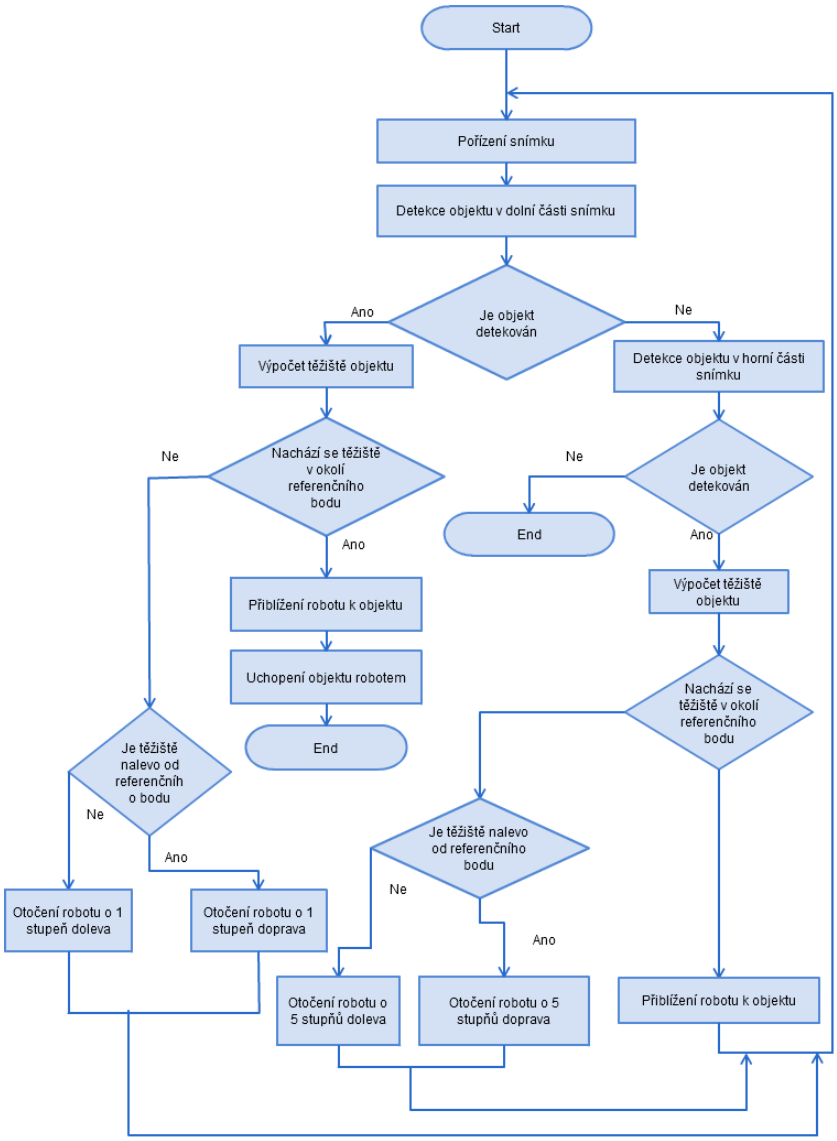
Review of Image Segmentation Techniques Including Pre, Post Processing Operation. 2015. *International Journal of Engineering and Advanced Technology (IJEAT)* [online]. [cit. 2015-04-14]. ISSN 2249–8958. Dostupné z: http://www.ijeat.org/attachments/File/v4i3/C3782024315.pdf .

Analýza obrazu. 2011-2015. *Analýza obrazu* [online]. [cit. 2015-04-14]. Dostupné z: http://www.analyza-obrazu.cz/ .

RUSS, John C. *The image processing handbook. 6th ed.* Boca Raton, FL: CRC Press, 2011. ISBN 978-143-9840-634.

Přílohy

A Vývojový diagram pro plánování trajektorie



Obrázek 21: Diagram