



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

BIN PICKING PRO 2D OBJEKTY

BIN PICKING FOR 2D OBJECTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jakub Robota

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Radomil Matoušek, Ph.D.

BRNO 2019

Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	Jakub Robota
Studijní program:	Strojírenství
Studijní obor:	Základy strojního inženýrství
Vedoucí práce:	doc. Ing. Radomil Matoušek, Ph.D.
Akademický rok:	2018/19

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 bakalářské práce:

Bin Picking pro 2D objekty

Stručná charakteristika problematiky úkolu:

V úloze bin picking jde o řešení velmi praktické úlohy výběru objektů z kontejneru. V uvedeném zadání bude řešena podstatně zjednodušená úloha, kdy bude objekt zájmu položen v rovině s nedefinovanou souřadnicí a rotací. Úkolem bude lokalizace a identifikace objektu pro následné uchopení objektu.

Cíle bakalářské práce:

1. Rešerše přístupů robotického vidění pro bin picking.
2. Tvorba aplikace pro 2D robotické vidění.
3. Aplikace pro rozpoznávání objektu z definované množiny objektů.
4. Praktická realizace s využitím řízené osy (SMC track).

Seznam doporučené literatury:

PAJDLA, Tomáš. Global optimization in camera and robot calibration: Globální optimalizace v kalibraci kamer a robotů. V Praze: České vysoké učení technické, [2017]. ISBN 978-80-01-06114-5.

KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Cieľmi tejto bakalárskej práce bolo spracovanie problematiky bin pickingu a návrh aplikácie pre 2D robotické videnie na identifikáciu objektu zo známej množiny s praktickou aplikáciou pri pohybe objektu vo vozíku po ose. Teoretická časť práce obsahuje rešerš bin pickingu a detailný prieskum využívaných metód v počítačovom videní. Praktická časť práce popisuje program vytvorený v prostredí MATLAB a algoritmy, ktoré boli v tomto programe používané. V závere praktickej časti je popísané reálne využitie tohto programu na detekciu objektu v pohybe po ose.

ABSTRACT

The goal of this bachelor thesis was to process issues of bin picking and to design an application for 2D robot vision used for object identification with practical application with the use of moving object on axis. Theoretical part of thesis contains research of bin picking and detailed research of methods used in computer vision. Practical part of thesis describes the program which was developed in MATLAB and algorithms, which are used in this program. End of practical part of thesis also describes real use of this program for detecting an object which is moving on axis.

KLÚČOVÉ SLOVÁ

Bin picking, počítačové videnie, rozpoznávanie obrazu, spracovanie obrazu, detekcia významných bodov, významné body SURF

KEYWORDS

Bin picking, computer vision, image recognition, image processing, feature detection, SURF features

BIBLIOGRAFICKÁ CITÁCIA

ROBOTA, Jakub. *Bin Picking pro 2D objekty*. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/117266>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce: Radomil Matoušek.

POĎAKOVANIE

Týmto by som chcel poďakovať doc. Ing. Radomilu Matouškovi, Ph.D. za pripomienky a rady týkajúce sa spracovania tejto bakalárskej práce, tiež by som chcel poďakovať mojim rodičom, sestre a v poslednom rade aj môjmu kamarátovi Sebastiánovi za poskytnutie kvalitného osvetlenia potrebného pri experimentálnej časti práce.

ČESTNÉ PREHLÁSENIE

Prehlasujem, že táto práca je mojím pôvodným dielom, spracoval som ju samostatne pod vedením doc. Ing. Radomila Matouška, Ph.D. a s použitím literatúry uvedenej v zozname literatúry.

OBSAH

1	ÚVOD.....	15
2	BIN PICKING.....	17
2.1	Druhy bin pickingu.....	17
2.1.1	Štruktúrovaný bin picking.....	17
2.1.2	Semi-štruktúrovaný bin picking.....	18
2.1.3	Náhodný bin picking.....	19
2.2	Používané technológie v bin pickingu.....	19
2.2.1	Senzory.....	19
2.2.2	Software.....	21
2.2.3	Koncové efekty.....	21
2.3	Rešerš komerčných riešení.....	22
2.4	Použitia bin pickingu.....	22
3	POČÍTAČOVÉ VIDENIE.....	23
3.1	Algoritmy používané pri počítačovom videní.....	23
3.1.1	Snímanie.....	23
3.1.2	Predspracovanie.....	23
3.1.3	Lokálne predspracovanie obrazu.....	25
3.1.4	Hranové detektory.....	27
3.1.5	Morfologické transformácie.....	30
3.1.6	Segmentácia.....	32
3.1.7	Popis objektov.....	35
3.1.8	Detekcia významných bodov.....	36
3.2	Neurónové siete.....	41
4	APLIKÁCIA PRE 2D ROBOTICKÉ VIDENIE.....	43
4.1	Popis problematiky problému a zvoleného programu.....	43
4.2	Špecifikácie použitej hardwarovej techniky na spracovanie obrazu.....	43
4.3	Spracovanie obrazu pomocou programu Matlab.....	44
4.3.1	Získanie obrazu pomocou kamery.....	44
4.3.2	Predspracovanie obrazu.....	44
4.3.3	Významné body SURF.....	48
4.3.4	Overovací experiment.....	52
5	ZÁVER.....	57
	ZOZNAM POUŽITEJ LITERATÚRY.....	59
	ZOZNAM SKRATIEK.....	63
	ZOZNAM OBRÁZKOV.....	65
	ZOZNAM PRÍLOH.....	67

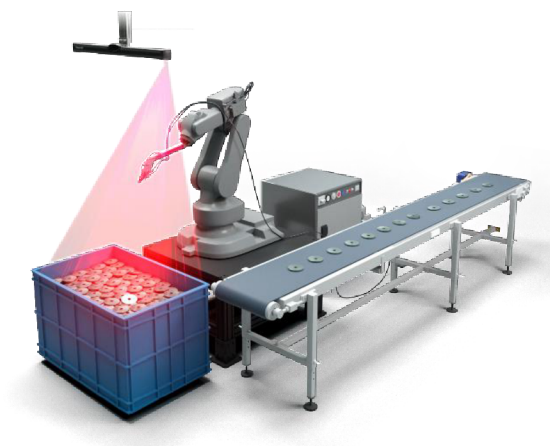
1 ÚVOD

V současnej dobe sa výrobné linky po celom svete stávajú viac robotizované. Implementácia riešení niektorých úloh sa však preukázala byť zložitejšia. Jednou z týchto úloh je vyberanie neznámych objektov z nádoby. Táto úloha sa nazýva bin picking. Bin picking je mnohými ľuďmi zaoberajúcimi sa robotizáciou často považovaný za „Svätý grál“ aplikovanej robotiky. Zložitosť a problematika bin pickingu je zviazaná ako s počítačovým videním tak aj s ovládaním robotického ramena.

V práci je postupne opísaný samotný bin picking, jeho rôzne druhy a technológie potrebné na realizáciu bin pickingu v praxi. Po rešerši bin pickingu nasleduje detailný rozbor algoritmov používaných v počítačovom videní, rozoberajúci fázy spracovania obrazu. Súčasťou kapitoly o počítačovom videní je aj popis umelej inteligencie využívanej v aplikáciách počítačového videnia. Ďalšou úlohou práce je vytvorenie aplikácie pre 2D robotické videnie. Na realizáciu tejto úlohy bol vybraný program MATLAB. V aplikácii sa využívajú kalibračné vzorky a pomocou viacerých algoritmov na spracovanie obrazu je možné tieto vzorky porovnať s obrazmi získanými kamerou. Táto aplikácia je následne využitá na riešenie problému identifikácie objektu a jeho polohy za pohybu po ose.

2 BIN PICKING

Bin picking je metoda, ktorá využíva robotov a počítačové videnie na úlohy týkajúce sa premiestnenia objektu z nádoby na predom určené miesto. Táto metóda má širokú škálu využitia pri automatizácii vo výrobe. Jej princíp spočíva vo využití rôznych senzorov na získanie dát o objektoch, tie sú ďalej spracované softwarom, ktorý je schopný navigovať robotickú ruku a pomocou nej daný objekt premiestniť na koncové miesto. Počas tohto premiestňovania objektu je možné použiť skener a CAD model na potvrdenie správnosti identifikovaného dielu a teda zaručenie kvality a presnosti výroby aj použitej metódy bin pickingu. Názorná ukážka bin pickingu je na obr. 1.



Obr. 1: Bin picking. [1]

Jednoduchšou úlohou, ktorá zdieľa časť problematiky s bin pickingom, je pick and place. Pick and place spočíva v rozpoznaní osamostatneného dielu a jeho uchopení nasledované presunom a položením dielu na stanovené miesto. V tejto práci je riešený zjednodušený problém 2D rozpoznávania obrazu podobný ako sa využíva pri pick and place aplikáciách.

2.1 Druhy bin pickingu

Bin picking je možné rozdeliť podľa úrovne usporiadania objektov, ktoré je potrebné z nádoby vyberať. Od danej úrovne usporiadania sa odvíja aj náročnosť riešenia tejto úlohy. Typy bin pickingu podľa usporiadania poznáme: štruktúrovaný, semi-štruktúrovaný a náhodný.

2.1.1 Štruktúrovaný bin picking

Štruktúrovaný bin picking je typ, pri ktorom sú objekty (diely) uložené v nádobe v predvídateľnom usporiadaní ako je možné vidieť na obrázku 2.



Obr. 2: Štruktúrovaný bin picking. [2]

Predvídateľné usporiadanie zjednodušuje rozpoznávanie polohy jednotlivých dielov a taktiež zjednodušuje ich vyberanie pomocou robotického ramena. Toto uľahčuje automatizáciu procesu, urýchľuje čas vyberania dielov z nádoby (part to part cycle time) a tiež znižuje nároky na komplexnosť použitého softwaru na rozpoznávanie objektov. Problémom tejto metódy je však nutnosť usporiadať objekty v nádobe pred procesom bin pickingu. Štruktúrovaný bin picking je metóda, ktorá sa už dlho používa vo výrobných fabrikách. [2, 3]

2.1.2 Semi-štruktúrovaný bin picking

Semi-štruktúrovaný bin picking je prechod medzi štruktúrovaným a náhodným bin pickingom. Diely sú menej usporiadané a je možné tu využiť len určitú predvídateľnosť. Toto usporiadanie znázorňuje obr. 3.



Obr. 3: Semi-štruktúrovaný bin picking. [2]

Menšia usporiadanosť súvisí s nárastom zložitosti tohto problému a teda aj jeho riešenia. Triedenie takéhoto usporiadania je softwarovo zložitejšie a teda aj finančne náročnejšie oproti štruktúrovanému bin pickingu. Tiež narastajú problémy s uchopením dielov pomocou robota a s tým aj čas vyberania dielov z nádoby. Nakoľko treba určitý stupeň predvídateľnosti usporiadania je opäť nutná príprava dielov v nádobe pred samotným procesom bin pickingu. Stupeň zložitosti týchto problémov narastá so stupňom náhodnosti rozmiestnenia dielov. [2, 3]

2.1.3 Náhodný bin picking

Pri náhodnom bin pickingu sú diely neusporiadané, môžu byť do nádoby vysypané alebo inak hromadne naložené vid' obr. 4 .



Obr. 4: Náhodný bin picking. [2]

Každá nádoba má diely inak rozmiestnené a diely majú úplne iné rozmiestnenie a orientáciu v porovnaní s minulou nádobou, nie je tu žiadna predvídateľnosť polohy a orientácie daných dielov. Tento typ bin pickingu je najnáročnejší. Môže sa tu vyskytnúť viac rôznych dielov, ktoré sa môžu prekrývať, dokonca môžu byť prepletené jeden cez druhý. Toto výrazne komplikuje rozpoznávanie dielov a ich zberanie pomocou ramena. [2 ,3]

2.2 Používané technológie v bin pickingu

Pri úlohách bin pickingu je využívané prepojenie medzi viacerými technológiami, každá z nich je určená na vykonanie potrebnej dielčej časti. Pomocou prepojení týchto procesov je možné vytvoriť komplexnú aplikáciu na bin picking. Pri bin pickingu je dôležité práve prepojenie senzorov, softwaru a koncového efektora. Bin picking a pick and place aplikácie využívajú rôzne druhy robotov podľa potreby danej úlohy, často používané roboty sú delta roboty, SCARA a tiež klasické angulárne roboty. [4]

2.2.1 Sensory

Pomocou senzorov získavame potrebné dáta o objektoch v pracovnom priestore na ďalšie spracovanie pomocou softwaru. Je možné použiť rozdielne druhy senzorov a tiež ich kombinácie. Podľa použitého druhu senzorov nadobúdame taktiež rôzne druhy dát o rôznej presnosti. Presnosť dát je veľmi podstatná a pri použití jednej kamery ju môžeme vylepšiť napríklad pridaním viacerých kamier a trianguláciou polohy pomocou dát z týchto kamier. Polohu pri pohybe môžeme taktiež upresniť pomocou zvýšenia snímok za jednotku času danej kamery.

Senzory na 2D videnie ako napríklad obyčajná video kamera získavajú dáta v podobe obrázkov, ktoré môžeme použiť na detekciu pohybov a lokáciu dielov. Senzory umožňujúce 2D videnie nie sú žiaden „nováčik“ na trhu avšak ich využitie je obsiahle a preto aj naďalej zostávajú relevantné pre mnoho aplikácií.

Senzory na 3D videnie sú čím ďalej viac používané a dostupné. 3D videnie je možné dosiahnuť po použití 2 a viac kamier v konfigurácii pre stereo videnie taktiež je možné použiť laserový skener alebo projekciu štruktúrovaného osvetlenia. Týmito senzormi a metódami sa teda dá detegovať 3. rozmer dielov, ktorý je ďalej podstatný pri určovaní ich tvarov a aj spôsobu, ktorým budú diely uchopované.

Pri použití kamier na stereo videnie získame 3. rozmer vďaka triangulácii, táto metóda využíva znalosti uhlov a natočení medzi kamerami, prípadne aj projektorom pomocou, ktorých je možné vypočítať polohu v priestore použitím znalostí uhlov v trojuholníku.

Laserový skener využíva buď jeden laserový lúč alebo laserový prúžok, ktoré zosnímajú povrch telesa a podľa uhlu odrazu a doby letu laserového lúča vyhodnotia polohu objektu.

Metóda štruktúrovaného osvetlenia (viď obr. 5) využíva preddefinované vzory premietané na daný objekt, objekt je snímaný kamerou, ktorá vďaka pretvoreniu daných vzorov na povrchu objektu vyhodnotí tvar tohto telesa. Fanuc 3D Area Sensor používa túto metódu štruktúrovaného osvetlenia a pomocou získaných dát sú vytvorené 3D mapy, ktoré sú ďalej využité systémom pri hľadaní dielov a určovaní ich tvarov. [5]



Obr. 5: Metóda štruktúrovaného osvetlenia. [5]

Existujú aj iné alternatívy kombinácie senzorov ako je napríklad pohybový senzor Kinect vyvinutý firmou Microsoft, tento senzor využíva štruktúrované infračervené svetlo na zistenie hĺbky v obraze v spojení s klasickou kamerou. [6] Jednou z ďalších alternatív je Intel Real Sense, tento produkt má rôzne varianty, ktoré využívajú buď dvojicu kamier alebo kombináciu infračerveného svetla s kamerou alebo v poslednom rade kamera Intel RealSense depth camera SR300 využíva štruktúrované svetlo. [7]

V bin pickingu použitie 3D videnia umožňuje detekciu objektu v 3D, vytvorenie 3D modelu a jeho nasledovnú analýzu na vybratie najlepšej možnej uchopovacej metódy a prípadné porovnanie s CAD modelom vzorového objektu. [5, 8]

2.2.2 Software

Software pri bin pickingu spracúva dáta zo senzorov. Spracovanie týchto dát a získanie vhodných informácií potrebných pre nasledovné uchopenie objektu je jednou z najpodstatnejších častí pri bin pickingu. Spracovanie dát je veľmi komplexné a je pri nej často využívaná umelá inteligencia. Z dát získaných senzormi sa využitím rôznych algoritmov segmentujú, detegujú a klasifikujú jednotlivé objekty. Z dát zo senzorov je potrebné získať aj informácie o polohe, rozmeroch a natočení jednotlivých objektov. Ďalej software volí poradie vybraných objektov a taktiež rozhoduje o optimálnej ceste robotického ramena. Pri výbere optimálnej cesty (path planning) je potrebné aby bola zvolená bezkolízna cesta, pri ktorej rameno robota dosiahne na objekt. Pri vyberaní dielov z nádoby sú prepočítané trasy medzi zistenou polohou súčasti, ktorá má byť uchopená, a výslednou vopred stanovenou polohou. Výpočty zvolenej trasy často prebiehajú priamo v systéme robota a použitými inverznými rovnicami dynamiky je možné vypočítať trasu do miesta kde sa nachádza diel. Pri použití rovníc je potrebné vedieť hmotnosti a momenty zotrvačnosti častí robota a daného dielu. Software býva optimalizovaný pre neustále zlepšovanie presnosti a skracovanie doby vyberania dielu. Čas spracovania dát závisí na faktoroch ako úroveň usporiadania dielov, počte senzorov, počte druhov dielov v nádobe, atď.

Pri klasifikácii objektov je možné využiť 2D alebo 3D model daného dielu, software je schopný hľadať podobnosti medzi objektami pri klasifikácii a využiť tento model na urýchlenie a tiež upresnenie procesu. [9, 10, 11]

2.2.3 Koncové efekторы

Po zosnímaní nádoby a spracovaní polohy dielov je potrebné dané diely z nádoby vytiahnuť a presunúť na požadované miesto pomocou koncového efektoru. Proces uchopovania súčasti a jeho automatizácie môže byť však problémový. Pri voľbe tvaru a typu efektoru sa totiž musí brať do úvahy charakter súčasti a tiež to, aké informácie o objekte, ktorý bude uchopovaný máme, napríklad presnosť jeho polohy a orientácie v priestore. Pri použití rôznych efektorov sa mení aj úspešnosť uchopovania a presúvania dielov pri vyššej rýchlosti a prípadnej zmene orientácie dielu počas manipulácie. Pri aplikáciách bin pickingu sa preto používajú rôzne efekторы, ktoré sú v niektorých prípadoch vyrábané na mieru pre zákazníka.

Jedným s klasických druhov koncového efektoru využívaných pri aplikáciách bin pickingu a pick and place je efektor s čel'usťami na uchytenie súčasti, počet čel'ustí môže byť rôzny a závisí od potreby aktuálnej aplikácie. Iný druh efektoru, ktorý sa využíva pre väčšinu prípadov, kde je vybraná súčasť z kovu je efektor s magnetom. Vďaka tomuto efektoru sme schopný uchopiť veľké množstvo rôznych dielov, avšak umiestnenie súčasti v presne stanovenej orientácii je pri použití tohto efektoru problémové. Ďalší často používaný druh efektoru je prísavka, ktorá sa používa pri dieloch s hladkým povrchom, jej nevýhodou je potreba pravidelnej údržby na docielenie správneho prisávania.

Koncové efektoři sú často schopné poskytovať spätnú väzbu o uchytení dielu pomocou rôznych vstavaných senzorov na detekciu prítomnosti súčasti alebo senzormi snímajúcich pôsobiace sily a momenty. Pomocou informácií zo senzorov sme schopný spätne zistiť správnosť uchopenia dielu alebo aj jeho orientáciu a rozmery. [12, 13]

2.3 Rešerš komerčných riešení

V súčasnosti existuje viacero firiem ponúkajúcich prakticky využiteľné riešenia pre bin picking. Tieto firmy sa obvykle zaoberajú vývinom senzorov spojených so softwarom, ktorý spracúva získané dáta.

Príkladom jednej z takýchto firiem je Photoneo. Photoneo ponúka riešenie pomocou nimi vytvorených senzorov a bin picking štúdia. Hlavnou prednosťou tejto firmy je ich kamera MotionCam-3D a skener PhoXi 3D Scanner, ktoré majú vysoké rozlíšenie, rýchly čas spracovania a taktiež boli viacnásobne ocenené a vďaka týmto oceneniam sú považované za najlepšie na trhu. Photoneo navrhlo a patentovalo vlastnú metódu štruktúrovaného svetla, ktorú využíva ich kamera MotionCam-3D. [1]

Pick-it ponúka kompletne riešenie obsahujúce 3D kameru, software a procesor na spracovanie dát a taktiež pomôcky na kalibráciu polohy kamery. 3D kamera s vysokým rozlíšením od Pick-it využíva štruktúrované svetlo na detekciu dielov. Software od Pick-it nevyžaduje programovanie, všetky potrebné nastavenia sa uskutočňujú pomocou rozhrania vo webovom prehliadači po prepojení počítača s procesorom Pick-it. [14]

IDS Imaging a ich Ensenso 3D kamery v spolupráci so softwarom EnsensoSDK sú schopné lokalizovať a klasifikovať diely. Podobne ako vyššie spomínané kamery využíva projekciu štruktúrovaného svetla na získanie dát o oblasti záujmu. [15]

2.4 Použitia bin pickingu

Existuje mnoho prípadov, kedy je pri výrobnom procese diel náhodne umiestnený v nádobe, riešením takého problému je často práve bin picking. Aj napriek tomu, že je bin picking relatívne nová technológia a je stále v procese rozvoja, je možné nájsť viacero úspešne zrealizovaných projektov využívajúcich túto technológiu. Vďaka možnosti použitia bin pickingu na širokú škálu dielov v rôznych prostrediach od automobilového priemyslu až linku, kde je zostavované záhradné náradie, vid' [16], má obrovský potenciál automatizovať v budúcnosti mnohé procesy.

3 POČÍTAČOVÉ VIDENIE

Počítačové videnie je vedecká disciplína, ktorej úlohou je naučiť počítače „vidieť“ alebo inak napodobniť funkciu ľudského videnia a vnímania. Videnie ľudí je ovplyvnené ich vlastnou inteligenciou a skúsenosťami. Realizácia úlohy videnia pre počítače je preto zložitá a komplexnosť tejto úlohy spočíva aj v nejasnostiach o tom ako presne fungujú biologické procesy pri videní ľudí. Pri počítačovom videní musí počítač po získaní obrazu spracovať a správne interpretovať informácie v tomto obraze. Počítačové videnie je stále vyvíjajúca sa disciplína, ktorá našla svoje uplatnenia v mnohých sférach. Počítačové videnie je využívané napríklad v medicíne, v autonómnych vozidlách alebo v už spomínanom bin pickingu. [17, 18]

3.1 Algoritmy používané pri počítačovom videní

Pri počítačovom videní je možné postup spracovávania a rozpoznávania zosnímaného obrazu rozložiť do viacerých krokov. V týchto krokoch je potrebné využiť mnoho výpočtových a aproximačných algoritmov za účelom uľahčiť a urýchliť výpočty vyhodnocovania a spracovania obrazu. [18]

3.1.1 Snímanie

Obraz je obvykle získaný zo vstupných informácií ako je napríklad jas, intenzita röntgenového žiarenia, ultrazvuk, tepelné žiarenie. Tieto informácie sú snímané pomocou rôznych kamier a senzorov. Pomocou digitalizácie je prevedený spojité signál na vstupe do diskretného tvaru. Ako výsledok digitalizácie je získaná matica zložená z prirodzených čísel, ktoré popisujú daný obraz, tzv. digitálnu obrazovú funkciu. Prvky matice sa nazývajú obrazové elementy (pixely), každý tento jeden prvok, element je ďalej nedeliteľná jednotka. Iný častý prípad reprezentácie obrazu je pomocou koeficientov dvojrozmernej Fourierovej transformácie, výhoda tohto spôsobu spočíva v možnosti použitia optických prostriedkov, ktoré sú schopné vykonať Fourierovu transformáciu ešte pred samotnou digitalizáciou obrazu. [18]

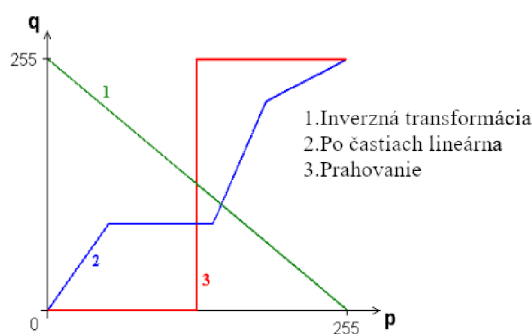
3.1.2 Predspracovanie

Predspracovanie obrazu je úprava obrázku odstránením a potlačením nechcených, nepotrebných informácií, ako je napríklad šum a skreslenie spôsobené vlastnosťami snímacieho zariadenia, ale aj úprava zvýraznením dôležitých znakov pre neskoršie operácie. Vstupmi aj výstupmi pri predspracovaní sú matice predstavujúce digitálnu obrazovú funkciu. Vstupná obrazová funkcia má veľký nadbytok informácií a toto je využité pri predspracovaní. Tento nadbytok je možné použiť napríklad pri úprave jednotlivých obrazových elementov skreslených šumom, nechcený šum je možné totiž upraviť pomocou priemeru susedných elementov. Pri predspracovaní obrazu nie je možné získať žiadne nové informácie, určité informácie sa dajú len potlačiť alebo naopak zvýrazniť, za účelom získania viac informácií je potrebné využiť lepšiu

metódu pri získavaní daného obrazu. Pri predspracovaní sa využíva viacero metód, ktoré sú volené podľa toho akú úpravu je potrebné vykonať. [18]

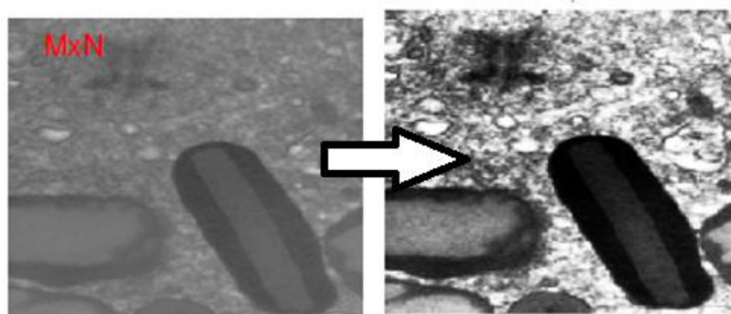
Bodové jasové transformácie

Pri týchto úpravách obrazu je snaha odstrániť chyby jas spôsobené nevhodným osvetlením alebo tiež chybami vznikajúcimi pri používaní hardwaru. Využívané sú dve rôzne skupiny metód: jasové korekcie a modifikácie jasovej stupnice. Pomocou jasových korekcií sa potlačujú chyby, ktoré vyplývajú zo skutočnosti toho, že snímacie zariadenia majú rôzne citlivosti podľa umiestnenia bodu v obraze. Pri transformácii jasovej stupnice nemá poloha bodu v obraze žiaden efekt, pôsobenie tejto transformácie je rovnaké na všetky body v obraze. Pôvodná stupnica jas označená ako p je transformovaná na novú stupnicu q , jasovú transformáciu zobrazuje obr. 6.



Obr. 6: Transformácia jasovej stupnice. [19]

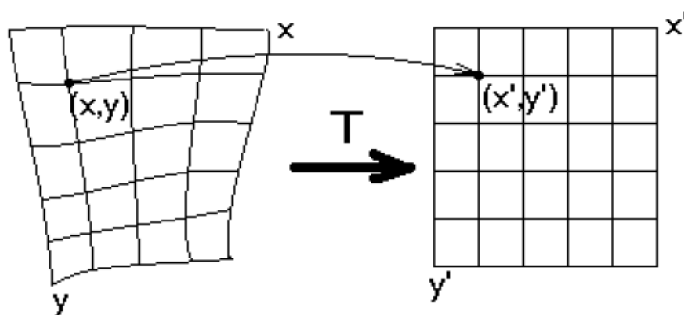
Transformácie jasovej stupnice sú dôležité na úpravy obrazu, po ktorých je ľahšia interpretácia daného obrazu, napr. zvýšenie kontrastu. Pri zvyšovaní kontrastu sa často používa vyrovnávanie histogramov. Po vyrovnaní histogramu obrazu sú jednotlivé hodnoty jas takmer rovnomerne zastúpené. Táto úprava zvyšuje kontrast pri maximálnych hodnotách histogramu a znižuje ho pri minimálnych, výsledok využitia vyrovnania histogramu je zobrazený na obr. 7. [18, 19]



Obr. 7: Vyrovnanie histogramu. [19]

Geometrické transformácie

Geometrická transformácia sa používa na úpravu obrazu, ktorý je skreslený. Toto skreslenie vzniká pri snímaní plochy pod uhlom iným ako je pravý alebo pri snímaní útvarov, ktoré nie sú ploché. Využité metódy nápravy obrazu pri geometrickom skreslení sú v princípe rovnaké ako pri geometrických transformáciách, ako sú napríklad rotácia a translácia. Rovnice transformácie sú pri niektorých transformáciách presne dané a známe, napr. pri rotácii, translácii, zmene mierky. Pri niektorých transformáciách je však nutné rovnice zistiť. Použité rovnice bývajú väčšinou aproximované pomocou polynómu. Pri geometrickej transformácii sa využívajú dva kroky, prvým z nich je plošná transformácia. Úlohou pri plošnej transformácii je nájsť bod vo výstupnom obraze, ktorý zodpovedá bodu vo vstupnom obraze. Druhým krokom je nájdenie správnej úrovne jasu, ktorá po transformácii bude odpovedať jednotlivým obrazovým elementom. Na hľadanie správnej úrovne jasu je používaná interpolácia, najčastejšou použitou metódou je v tomto prípade metóda najbližších susedov. Metóda najbližších susedov sa využíva aj pri transformácii. rozlíšenia. Ukážku jedného z možných použití geometrickej transformácie je možné vidieť na obr. 8. [18, 19]



Obr. 8: Geometrická transformácia. [19]

3.1.3 Lokálne predspracovanie obrazu

Metódy lokálneho predspracovania sa používajú na výpočet jasu bodu vo výstupnom obraze za použitia hodnôt z lokálneho okolia daného bodu vo vstupnom obraze. Podľa cieľu predspracovania môžeme tieto metódy deliť na dve skupiny, vyhladzovanie obrazu a gradientné operácie. Druhé delenie je podľa funkčného vzťahu, a to na lineárny a nelineárny. Hlavné zdroje tejto podkapitoly sú [18, 19, 20].

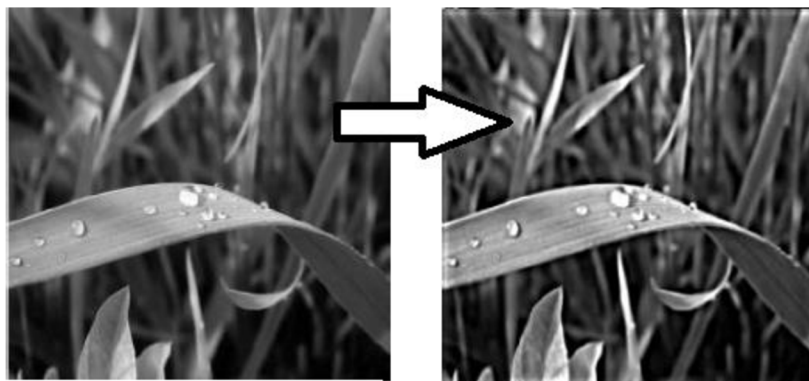
Vyhladzovanie obrazu je používané na potlačenie vyšších frekvencií obrazovej funkcie, týmto dochádza k potlačeniu nechceného náhodného šumu v obraze, žiaľ dochádza aj k potlačeniu ďalších náhlych zmien jasu ako napríklad ostré hrany, ktoré sa dajú považovať ako podstatná informácia obrazu.

Gradientné operácie, na rozdiel od vyhladzovania obrazu, vyššie frekvencie obrazovej funkcie posilnia. Pri posilnení vyšších frekvencií zvýrazníme ostré hrany v obraze, ale tiež zvýrazníme aj šumové body. Vyhladzovanie obrazu a gradientné operácie sú teda protiklady, existujú však algoritmy, ktoré sú schopné vyhladiť šum a pritom zaostriť hrany pomocou kombinácie vyhladzovania a gradientných operácií.

Na vyhladzovanie obrazu alebo na zvýraznenie hrán v obraze sa používa filtrácia. Obraz je možné filtrovať buď pomocou priestorovej filtrácie alebo je možné filtrovať vo frekvenčnej oblasti. Pri filtrácii vo frekvenčnej oblasti je potrebné obraz previesť do frekvenčnej oblasti pomocou Fourierovej transformácie po tomto prevode nasleduje použitie filtra a spätná Fourierová transformácia.

Vo frekvenčnej oblasti sa používajú:

- Dolnofrekvenčné filtre – zoslabujú vysoké frekvencie a ponechávajú hodnoty nízkych frekvencií nezmenené, týmto dochádza k vyhladeniu obrazu v priestorovej doméne.
- Hornofrekvenčné filtre – zoslabujú nízke frekvencie a ponechávajú vysoké, tým získame zosilnenie hrán po spätnej transformácii do priestorovej domény.
- Pásmové filtre – zoslabia veľmi nízke a veľmi vysoké frekvencie pri zachovaní stredného pásma frekvencií a vďaka tomu je možné vyhladiť obraz a zároveň zvýrazniť detaily. Príklad použitia tohto filtra je znázornený na obr. 9.



Obr. 9: Použitie pásmového frekvenčného filtra. [21]

Pri filtrácii v priestorovej doméne sa zvykne používať diskrétna konvolúcia obrazu s konvolučnou maskou, táto operácia sa dá zapísať rovnicou (1). Konvolúcia sa použije aj pri filtrácii vo frekvenčnej doméne pomocou použitia vety o konvolúcii.

$$f(i, j) = \sum_{(m, n) \in O} h(i - m, j - n) \cdot g(m, n), \quad (1)$$

kde h vyjadruje konvolučnú masku, O okolie o rozmeroch m, n a i, j súradnice stredového elementu. V priestorovej doméne sa často používajú:

- Lokálny aritmetický priemer – Priradí bodu hodnotu jasu, ktorá odpovedá aritmetickému priemeru bodov v jeho okolí vrátane tohto bodu samotného. Tento filter potlačí škvrny šumu, ktoré sú menšie ako veľkosť použitého okolia. Na získanie lokálneho aritmetického priemeru pri okolí 3x3 sa využíva napríklad konvolučná maska h z rovnice (2).

$$\mathbf{h} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

- Metóda mediánu – Hodnoty okolitých bodov sú zoradené podľa veľkosti a z týchto hodnôt sa vyberie medián, hodnota mediánu je priradená ako nová hodnota bodu, ktorý je preskúmaný. Podobne ako pri metóde s priemerom sa používa na potlačenie šumu v obraze. Narozdiel od priemeru je medián menej náchylný na silné pôsobenie od jedného pixelu s veľmi odlišnou hodnotou od ostatných pixelov v okolí.
- Konvolučný filter s Gaussovým rozdelením – Používa konvolučnú masku s hodnotami, ktoré sú založené na Gaussovom rozdelení vid' obr. 10.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Obr. 10: Konvolučný filter 5x5 s Gaussovým rozdelením. [19]

- Konzervatívny filter – Tento filter odstraňuje izolované pixely s mimoriadne vysokými alebo mimoriadne nízkymi hodnotami intenzity. Z pixelov v okolí sú zistené hodnoty minima a maxima. V prípade že hodnota tohto pixelu nie je medzi hodnotami minima a maxima, tak je hodnota tohto pixelu nahradená najbližšou hodnotou intenzity z okolia. Ak je hodnota pixelu medzi minimom a maximom tak je pixelu ponechaná jeho pôvodná hodnota.
- Filtrácia metódou rotujúcej masky – Podľa homogenity jasu sa hľadá časť okolia filtrovaného bodu, ktorá podľa podobnosti patrí k danému bodu. Nájdená homogénna časť okolia je ďalej použitá na výpočet výslednej hodnoty. Pri použití tejto metódy nie sú hrany rozmazané.

3.1.4 Hranové detektory

Detekcia hrán spočíva vo využití detektora na nájdenie a zvýraznenie miest, kde je náhla zmena jasu. Hrany preto určíme podľa veľkosti gradientu a jeho smeru. Veľkosť gradientu vychádza z rovnice (3) a jej smer z rovnice (4), kde $\mathbf{g}(x, y)$ vyjadruje obrazovú funkciu.

$$|\nabla \mathbf{g}(x, y)| = \sqrt{\left(\frac{\partial \mathbf{g}}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{g}}{\partial y}\right)^2}, \quad (3)$$

$$\psi = \tan^{-1} \left(\frac{\frac{\partial g}{\partial y}}{\frac{\partial g}{\partial x}} \right), \text{ pre } \frac{\partial g}{\partial x} \neq 0 \quad (4)$$

Pre diskretný obraz sa parciálne derivácie aproximujú diferenciami Δ s krokom n (obvykle $n = 1$):

$$\Delta_i g(i, j) = g(i, j) - g(i - n, j), \quad (5)$$

$$\Delta_j g(i, j) = g(i, j) - g(i, j - n). \quad (6)$$

Jedna z možností ako aproximovať deriváciu je pomocou diskkrétnej konvolúcie, kde využijeme maticové operátory, a to konvolučné masky (kernely). Takéto operátory aproximujú 1. deriváciu obrazovej funkcie a často sa nazývajú kompasové operátory nakoľko nie sú invariantné voči rotácii a je teda potrebné používať rôzne natočené konvolučné masky, ktorých celkovú odozvu následne sčítame. [18,24]

Robertsov operátor (známe aj ako Robertsov križ) je výpočtovo nenáročný operátor využívajúci okolie 2×2 . Používa konvolučné masky:

$$\mathbf{h}_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{h}_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (7)$$

Hrany sa detegujú medzi diagonálnymi susedmi a výslednú odozvu $\mathbf{g}(x, y)$ tohto operátoru je možné vyjadriť pomocou konvolúcie masiek s obrazom $\mathbf{f}(x, y)$ podľa vzťahu (8), často sa využíva odhad na zistenie veľkosti pomocou vzťahu (9), ktorý je možné vypočítať rýchlejšie. Nevýhodou tohto operátora je vyššia citlivosť na šum, kvôli kontrole iba malého 2×2 okolia. [18, 23, 24]

$$\mathbf{g}(x, y) = \sqrt{[\mathbf{f}(x, y) * \mathbf{h}_x]^2 + [\mathbf{f}(x, y) * \mathbf{h}_y]^2}, \quad (8)$$

$$|\mathbf{g}(x, y)| = |\mathbf{f}(x, y) * \mathbf{h}_x| + |\mathbf{f}(x, y) * \mathbf{h}_y| \quad (9)$$

Sobelov operátor obvykle používa symetrický kernel o rozmeroch 3×3 , nakoľko využíva väčšiu oblasť tak je pomalší na vyhodnotenie. Pri používaní kernelu 3×3 sa obraz viac vyhladí ako pri použití Robertsovho operátora a teda Sobelov operátor je menej náchylný na vplyv šumu. Konvolučná maska tohto operátora je definovaná v 4 rôznych pootočeníach zobrazených v (10). [18, 23, 24]

$$\begin{aligned} \mathbf{S}_x &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, & \mathbf{S}_{xy} &= \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \\ \mathbf{S}_y &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, & \mathbf{S}_{yx} &= \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}. \end{aligned} \quad (10)$$

Ďalšie operátory fungujúce na podobnom princípe ako Sobelov sú napríklad Prewittovej operátor využívajúci symetrický kernel, Kirschov operátor s nesymetrickým kernelom,

Robinsonov operátor, atď. Hlavné rozdiely týchto operátorov sú v koeficientoch konvolučných masiek. [18, 24]

Operátory, ktoré aproximujú 2. deriváciu sú založené na tom, že hrany sa nachádzajú v miestach kde má 1. derivácia svoje minimum alebo maximum a namiesto zisťovania týchto maxím a miním zistia miesta, kde 2. derivácia prechádza nulou. Tieto operátory sú invariantné od rotácie. Na výpočet 2. derivácie sa využíva Laplaceov operátor ∇^2 , ktorý sa vypočíta zo vzťahu:

$$\nabla^2(x, y) = \frac{\partial^2 \mathbf{g}(x, y)}{\partial x^2} + \frac{\partial^2 \mathbf{g}(x, y)}{\partial y^2} \approx \mathbf{g}(x, y) * \mathbf{L}_n. \quad (11)$$

Obrazová funkcia $\mathbf{g}(x, y)$ je ale reprezentovaná ako diskkrétne pixely a je teda potrebné použiť konvolučný kernel, ktorý by zodpovedal Laplaceovmu operátoru a bol schopný aproximovať 2. deriváciu. Dva obvykle používané kernely podľa sú:

$$\mathbf{L}_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{L}_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (12)$$

kde \mathbf{L}_4 značí kernel pre 4-susedstvo a \mathbf{L}_8 pre 8-susedstvo stredného pixelu. (Znamienka pri koeficientoch v konvolučnom jadre je možné zapísať aj opačne.) Táto aproximácia je silne ovplyvnená šumom v obraze a často je sa preto využíva filter s Gaussovým rozdelením na vyhladenie obrazu pred použitím Laplaceových kernelov. Tento proces zjednotenia Gaussovho vyhladzovacieho filtra a Laplaceána využíva LoG operátor. Pre prechody nulou pri LoG operátore podľa platí:

$$\nabla^2(\mathbf{G} * \mathbf{g}) = (\nabla^2 \mathbf{G}) * \mathbf{g} = 0. \quad (13)$$

Derivácia $\nabla^2 \mathbf{G}$ sa dá vopred vypočítať a teda je nutná iba jedna konvolúcia s obrazom. [18, 24]

Cannyho detektor hrán je detektor navrhnutý ako optimálny na detekciu skokových hrán v obraze. Je založený na troch podstatných kritériách. Tieto kritéria sú: správna detekcia všetkých podstatných hrán, dobrá lokalizácia na minimalizovanie vzdialenosti detegovaných a reálnych hrán a nakoniec zaistenie iba jedinej odozvy pri jednej hrane, tj. zamedzenie viacnásobných detekcií jednej a tej istej hrany. Cannyho detektor na dodržanie týchto kritérií využíva viacero krokov. Najprv je obraz vyhladený pomocou Gaussovho filtra, po ňom nasleduje nájdenie gradientu obrazu. Po zistení gradientu je potrebné potlačiť hodnoty, ktoré nie sú maximum (tzv. Non-maximum suppression) čím nastane stenčenie hrán. Posledný krok nazývaný hysteréza, spočíva v tom, že Cannyho metóda používa dva prahy, horný a dolný, ak pixel prekročí hornú hranicu je uznaný ako hrana, ak je pixel pod dolnou hranou je vylúčený, ak sa ale nachádza medzi týmito prahmi je vyšetrené jeho okolie a ak má tento pixel susedný pixel, ktorý prekonal prah tak aj nami vyšetřovaný pixel bude vyhodnotený ako hrana. [18, 22, 23, 24, 25]

3.1.5 Morfológické transformácie

Morfologické transformácie sú operácie založené na zmene pôvodného obrazu pomocou štruktúrnych prvkov. Transformácie sú obvykle vykonávané na binárnych obrazoch, avšak niektoré verzie sú použiteľné aj pri obrazoch v odtieňoch sivej. Pomocou transformácie je pôvodný obraz spolu so štruktúrnym elementom skombinovaný použitím množinových operátorov. Základné morfologické operácie sú dilatácia a erózia a pomocou nich je možné získať aj komplexnejšie formy operácií ako sú napríklad morfologické otvorenie alebo zatvorenie obrazu a iné. [18, 23, 26]

Dilatácia

Efekt dilatácie pri použití na binárny obraz je rozšírenie okrajov oblastí. Táto operácia sa dá popísať pomocou vektorového súčtu dvoch množín (známe aj ako Minkovského množinový súčet). Vyjadrenie dilatácie ako zjednotenie dvoch bodových množín X a B posunutých o $b \in B$, kde X vyjadruje obraz a B štruktúrny element, je možné vidieť vo vzorci (14).

$$X \oplus B = \bigcup_{b \in B} X_b \quad (14)$$

Pri dilatácii je vyšetřovanému pixelu priradená hodnota 1 v prípade ak aspoň jeden iný pixel jeho okolia v rámci kernelu je tiež 1. Názorné použitie dilatácie je zobrazené na obr. 11.



Obr. 11: Znázornenie dilatácie. [26]

Erózia

Erózia je druhým operátorom zo základných morfologických operátorov. Erózia ponechá pixelu z pôvodného obrazu hodnotu 1 iba ak sú všetky pixely jeho okolia v rámci kernelu tiež 1, ak pixel túto hodnotu nesplní tak je jeho hodnota zmenená na 0. Erózia je definovaná podľa vzorca (15) a jej použitie zobrazuje obr. 12.

$$X \ominus B = \bigcap_{b \in B} X_{-b} \quad (15)$$



Obr. 12: Názorná ukážka erózie. [26]

Otvorenie

Otvorenie je transformácia, ktorá použije eróziu a následne po nej dilatáciu pri použití rovnakého štruktúrneho elementu na úpravu obrazu. Otvorenie je definované pomocou vzorca (16).

$$X \circ B = (X \ominus B) \oplus B \quad (16)$$

Pri obvyklej jednoduchšej erózii je možné odstrániť nechcený šum, typu „salt noise“, popredných pixelov s hodnotou 1, avšak touto úpravou sú zasiahnuté nielen pixely šumu ale všetky pixely, tento nechcený efekt napravíme nasledujúcou dilatáciou. Ak eróziu nasleduje dilatácia tak po erodovaní nechceného šumu dilatácia upraví zostávajúce oblasti na ich pôvodnú podobu, ktorú mali pred erodovaním. Transformácia pomocou morfológického otvorenia je znázornená na obrázku 13.



Obr. 13: Morfológické otvorenie. [25]

Uzavretie

Transformácia nazývaná uzavretie používa podobne ako otvorenie kombináciu dilatácie a erózie v poradí kde je dilatácia nasledovaná eróziou. Uzavretie je definované podľa vzorca (17).

$$X \cdot B = (X \oplus B) \ominus B \quad (17)$$

Uzavretie je možné použiť pri šume typu „pepper noise“, dilatácia vyplní diery v obraze spôsobené týmto šumom a následná erózia napravi nechcené rozšírenie oblastí od dilatácie. Príklad použitia uzavretia na odstránenie šumu „pepper noise“ je zobrazený na obr. 14.



Obr. 14: Morfológické zatvorenie. [25]

Dôležitou vlastnosťou morfológického otvorenia a zatvorenia je idempotentnosť. Idempotentnosť je vlastnosť keď sa po opakovanom použití operácie už získaný výsledok ďalej nemení. To znamená, že použitím morfológického otvorenia na množinu **X** pomocou štruktúrneho elementu **B** je výsledná množina otvorená vzhľadom k **B** a opakované použitie tejto operácie už túto množinu nezmení.

Existujú ešte viaceré ďalšie operácie ako napríklad skelet, hit or miss, stenčovanie, zosilňovanie a iné. Pri popise objektu tenkými čiarami sa používa skelet. Skelet je vytvorený z čiar, ktoré sú zjednotené stredy kružníc, tieto kružnice sú situované vo vnútri objektu a dotýkajú sa ho v aspoň dvoch bodoch. Transformácia hit or miss sa dá popísať ako hľadanie zhody medzi obrazom a zvoleným štruktúrnym elementom. Použitím hit or miss je možné zosilniť alebo stenčiť obraz. [18, 23, 25, 26]

3.1.6 Segmentácia

Segmentácia je krok, pri ktorom rozčleníme obraz do viacerých oblastí, ktoré majú súvislosť s predmetmi alebo oblasťami z reálneho sveta. Segmentácia je potrebným krokom spracovania, po ktorom je obraz možné ďalej analyzovať vo vyššej úrovni spracovania. Výsledok segmentácie sú oblasti, ktoré sa neprekrývajú a v prípade kompletnej segmentácie jednoznačne odpovedajú objektom na obraze alebo v prípade čiastočnej segmentácie nemusia priamo odpovedať objektom. Segmentácia môže vychádzať buď z globálnych znalostí obrazu ako sú napríklad farby, tvary, polohy objektov. Segmentácia môže vychádzať aj z určovaní hraníc medzi oblasťami alebo môže dané oblasti vytvárať.

Problémy ovplyvňujúce segmentáciu sú často spôsobené nepresnosťou získaných obrazových dát. Existujú viaceré metódy segmentovania obrazu a aj s ich rôznymi nastaveniami parametrov je možné získať rozdielne výsledky a preto je potrebné voliť metódy a ich parametre podľa aktuálne riešenej úlohy.

Segmentačné metódy môžeme rozdeliť na :

- Prahovaním
- Na základe detekcie hrán
- Narastaním oblastí
- Pomocou porovnávania so vzorom
- Hybridné metódy

Ďalej nasleduje presnejší popis niektorých vybraných metód segmentácie. [18, 27]

Segmentácia prahovaním

Segmentácia prahovaním je metóda, ktorá sa zakladá na jednotnosti vlastností (farba, jas, atď.) pre jeden objekt. Táto metóda je veľmi jednoduchá a nenáročná na výpočet. Prahovanie je segmentácia vstupného obrazu $f(i, j)$ na výstupný segmentovaný binárny obraz $g(i, j)$ pri použití prahu T podľa vzorca (18).

$$g(i, j) = \begin{cases} 1 & \text{pre } f(i, j) \geq T, \\ 0 & \text{pre } f(i, j) < T \end{cases} \quad (18)$$

Je možné použiť aj prahovanie s viacerými prahmi, čiastočné, adaptívne a iné. Metódy na zistenie prahu sú napríklad experimentálne zistenie prahu, zistenie z histogramu, podľa percent, atď. [18, 27]

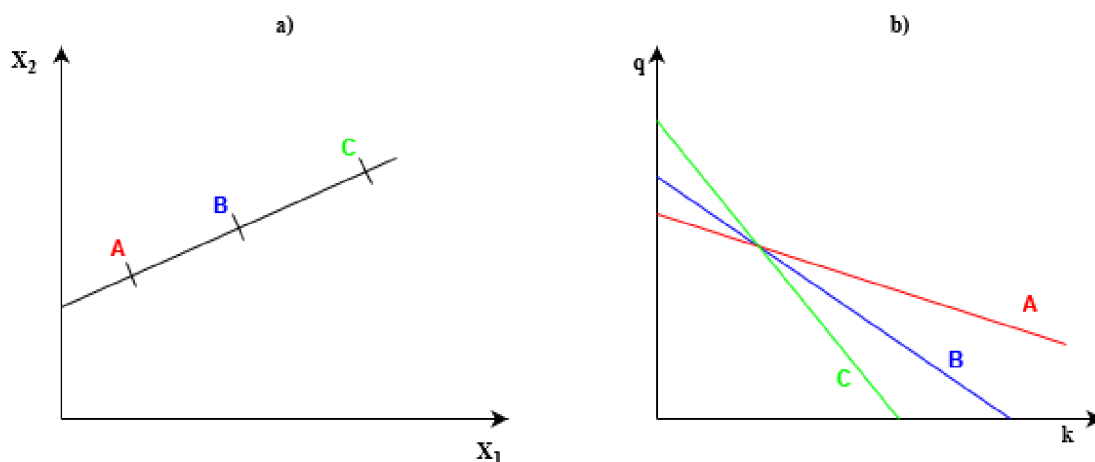
Segmentácia na základe detekcie hrán

Metóda vychádza z toho, že objekty sú ohraničené hranami a teda výsledné oblasti segmentácie je možné získať pomocou hranových detektorov. Po získaní hrán v obraze často nasleduje ďalšie spracovanie potrebné na odstránenie nechcených a problémových miest ako sú napríklad miesta, kde po použití hranového detektoru bola nájdená hrana aj keď sa v skutočnosti v tomto mieste hrana nenachádza alebo miest kde bola práve reálna hrana nenájdená. Obraz hrán je napríklad možné prahovať a tým odstrániť miesta hrán, ktoré majú hodnotu gradientu pod prahom.

V prípade ak nie sú hranice známe, ale sú určené oblasti v obraze, je možné stanoviť hranice týchto oblastí pomocou sledovania hranice. Obrazom sa postupne prechádza riadkami pokým sa nenájde bod oblasti, v tomto východnom bode oblasti sa použije okolie 3x3 a pomocou 4-susedstva alebo 8-susedstva sa hľadá ďalší bod hranice, tento proces pokračuje dovtedy dokým sa opäť nevráti na východný bod. Smer pohybu z aktuálneho bodu do ďalšieho bodu hranice je označený číslom, ako výstup sledovania hranice je zápis hranice do Freemanovho kódu, ktorý je zložený z čísel, ktoré vyjadrujú smer pohybu pri sledovaní hranice.

Houghova transformácia sa využíva pri segmentácii ak poznáme popis tvaru hľadaného objektu. Houghovu transformáciu je možné využiť pri detekcii jednoduchých známych tvarov v obraze, ktorých tvar ide analyticky popísať. Základný princíp tejto transformácie spočíva v premietnutí obrazového priestoru do priestoru parametrov. Premietnutím bodu $A = (X_1, X_2)$ z obrazového priestoru do priestoru parametrov získame priamku s parametrami k a q , táto priamka vyjadruje všetky priamky prechádzajúce bodom A v obrazovom priestore. Po nájdení takýchto priamok pre body

ktoré ležia na jednej priamke v obrazovom priestore získame priesečník priamok v priestore parametrov. Na obrázku 15 je možné vidieť princíp Houghovej transformácie, a) znázorňuje obrazový priestor a b) priestor parametrov.

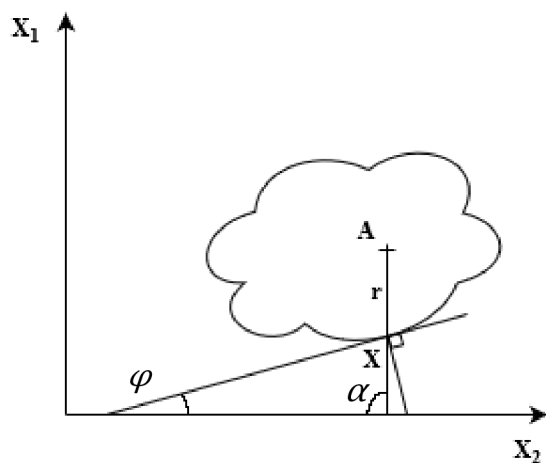


Obr. 15: Houghova transformácia.

Pri objektoch rôznych tvarov sa využívajú rôzne rovnice. Ak nie je parametrická rovnica hraníc oblastí známa tak je nutné použiť zobecnenú metódu Houghovej transformácie. Pri tejto metóde je zvolený referenčný bod vo vnútri oblasti, od tohto bodu je vedená polpriamka. V priesečníku tejto polpriamky a hranice oblasti je možné zistiť smer hrany a vzdialenosť priesečníku od referenčného bodu $R = (X_1, X_2)$. Z týchto hodnôt φ je zostavená tabuľka pre každé φ , kde ku každému φ patrí aspoň jeden bod X , ku ktorému prislúcha vzdialenosť od referenčného bodu r a uhol α . Podľa týchto hodnôt je možné napísať súradnice nájdeného bodu hrany v obraze so smerom hrany φ :

$$X = [X_1 + r(\varphi) \cdot \cos(\alpha(\varphi)), X_2 + r(\varphi) \cdot \sin(\alpha(\varphi))] . \quad (19)$$

Na obrázku 16 je možné vidieť zostrojenie polpriamky a priesečníku s objektom potrebné na zistenie hodnôt, z ktorých je zostrojená tabuľka.



Obr. 16: Konštrukcia pri Houghovej transformácii.

Výhodou Houghovej transformácie je veľká necitlivosť tejto metódy na šum a porušenie hraníc oblasti.

Ďalšie metódy využívajúce detekcie hrán sú napríklad heuristické sledovanie hranice, využívajúce prehľadávanie grafu, segmentácia z hrán pomocou znalostí o polohe, využitie aktívnych kontúr alebo využitie level-set funkcie. [18, 27]

Segmentácia narastaním oblastí

Tieto metódy spájajú oblasti na základe homogenity. Na začiatok je obraz rozdelený do malých oblastí, ktoré sa ďalej spájajú a narastajú ak splňajú dané kritérium homogenity. Taktiež je možné využiť metódu, ktorá štiepi obraz do menších oblastí na základe pyramidovej reprezentácie obrazu, ak je obraz v oblasti nemohogénny tak je oblasť rozdelená na 4 oblasti nižšej úrovne, ak je obraz v oblastiach homogénny tak sa oblasti spoja. Príkladom kritéria homogenity môže byť napríklad homogenita jasu pre susedné oblasti, kedy sú oblasti spojené ak je rozdiel priemerného jasu týchto dvoch oblastí menší ako hodnota stanovená kritériom. Výsledok sa líši pre rôzne kritériá homogenity, ale aj pre rôzne začiatkové body a poradie spájania. [18, 28]

Segmentácia porovnaním so vzorom

Metódy využívajúce vzory, šablóny hľadajú podobnosti medzi týmito vzormi a objektami v obraze. Úloha porovnania obrazu voči šablóne je však zložitejšia kvôli nepresnostiam v obraze, kde sa obvykle nevyskytuje presná zhoda šablóny. Nehľadá sa teda presná zhoda so šablónou, ale namiesto toho sa hľadá miera zhody, príkladom tejto miery je vzájomná korelácia. [18, 28]

Hybridné metódy segmentácie

Do tejto kategórie patria rôzne metódy, ktoré môžu kombinovať aj viacero prístupov k segmentácii. Príkladmi sú napríklad neurónové siete (viac v kapitole 3.2), segmentácia pomocou morfológických operácií, na základe pozorovania pohybu objektov v obraze, atď. [28]

3.1.7 Popis objektov

Po segmentácii je možné opísať tvar a rôzne geometrické vlastnosti týchto segmentovaných objektov. Príkladom geometrickej vlastnosti je dĺžka hranice oblasti, jednoduchý prípad zistenia dĺžky hranice oblasti nastane ak je hranica reprezentovaná Freemanovým kódom, kde pri 4-susedstve je každý posuv rovný vzdialenosti jedného pixelu a pri 8-susedstve sú horizontálne a vertikálne posuvy rovné vzdialenosti jedna a diagonálne $\sqrt{2}$. Niektoré ďalšie popisy oblastí sú napríklad veľkosť, ktorá je daná súčtom pixelov obsiahnutých v oblasti, orientácia, ktorú je možné určiť napríklad ako natočenie dlhšej strany opísaného obdĺžnika alebo natočenie dlhšej polosi opísanej elipsy, ťažisko objektu, atď. [18]

3.1.8 Detekcia významných bodov

V segmentovanom obraze je možné nájsť významné body tohto obrazu. Obraz je možné opísať pomocou významných bodov, ktoré sa prejavujú určitými jedinečnými vlastnosťami a teda vynikajú spomedzi bodov v ich okolí. Pri detekcii týchto bodov je potrebné aby bolo možné nájsť body opakovane. Po detekcii týchto bodov je však potrebné popísať ich vlastnosti, na tento popis sa využívajú deskriptory.

Existuje viacero rôznych detektorov a algoritmov využívajúcich významné body, ďalej sú opísané niektoré z nich, ktoré sú všeobecne používané. [29]

Harrisova metóda

Harrisov detektor významných bodov používa rohy v obraze ako významné body. Body sú hľadané pomocou hľadania rozdielov v jase pre posun (u, v) , toto vyjadruje rovnica :

$$E(u, v) = \sum_{x,y} w(x, y) \cdot [I(x + u, y + v) - I(x, y)]^2, \quad (20)$$

kde $w(x, y)$ je funkcia vyšetřovaného okna, toto okno je buď obdĺžnikové alebo Gaussove, vzhľadom na tieto okná sú vzťahované váhy a $I(x, y)$ značí jasovú funkciu. Na nájdenie rohov je potrebné nájsť miesta kde má funkcia $E(u, v)$ maximum, v tomto prípade je potrebné maximalizovať člen $I(x + u, y + v)$. Použitím Taylorovho rozvoja a po úpravách získame vzorec (21) a premenná \mathbf{M} je vyjadrená vo vzorci (22).

$$E(u, v) \approx [u \quad v] \cdot \mathbf{M} \cdot \begin{bmatrix} u \\ v \end{bmatrix} \quad (21)$$

$$\mathbf{M} = \sum_{x,y} w(x, y) \cdot \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \quad (22)$$

I_x vyjadruje v rovnici (22) parciálne derivácie jasovej funkcie v smere ose x a I_y v ose y , tieto derivácie sú aproximované diferenciou. Rovnica (23) je rovnica, podľa ktorej sa určí či v danom okne je roh alebo nie.

$$R = \det(\mathbf{M}) - \kappa \cdot (\text{trace}(\mathbf{M}))^2 \quad (23)$$

Kde $\det(\mathbf{M}) = \lambda_1 \cdot \lambda_2$, $\text{trace}(\mathbf{M}) = \lambda_1 + \lambda_2$, pričom λ_1 a λ_2 sú vlastné čísla matice \mathbf{M} a κ je empirická konštanta.

R závisí od λ_1 a λ_2 pričom:

- Ak je $|R|$ je nízke číslo, teda ak λ_1 a λ_2 sú tiež malé čísla, tak skúmaná oblasť je plochá, tj. bez hrán a rohov.
- Ak je $R < 0$, toto nastane ak $\lambda_1 \gg \lambda_2$, v oblasti sa vyskytuje hrana.
- Ak R je veľké číslo, teda λ_1 a λ_2 sú veľké a zároveň je $\lambda_1 \sim \lambda_2$, tak oblasť obsahuje roh.

S týmito znalosťami je možné opísať algoritmus, ktorý Harrisov detektor využíva. Algoritmus je nasledujúci :

- 1) Odfiltrovanie šumu

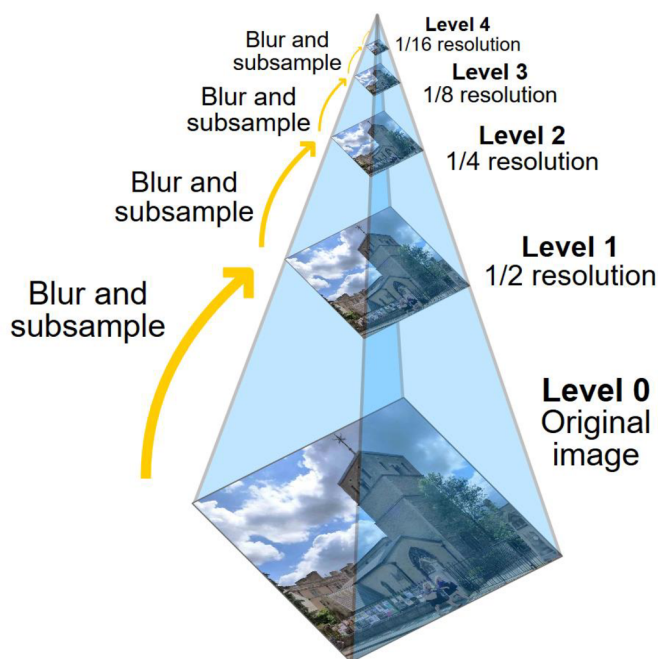
- 2) Výpočet gradientov I_x a I_y
- 3) Výpočet vlastných čísel λ_1 a λ_2 matice M
- 4) Výpočet hodnoty R
- 5) Určenie významných bodov pomocou prahu a maximálnych hodnôt v R

Hlavnými výhodami tohto algoritmu je invariancia voči rotácii, translácii a nízka výpočetná náročnosť avšak jeho hlavnou nevýhodou je, že nie je invariantný voči zmene merítka obrazu. Závislosť na zmene mierky je zjavná pri pohľade na viacero rohov v obraze, niektoré rohy sú menšie a niektoré väčšie ak pri tejto skutočnosti zmeníme mierku obrazu, rohy sa môžu zmeniť a príliš malé alebo veľké a môžu byť vyradené napríklad pri prahovaní. [29, 30]

SIFT detektor

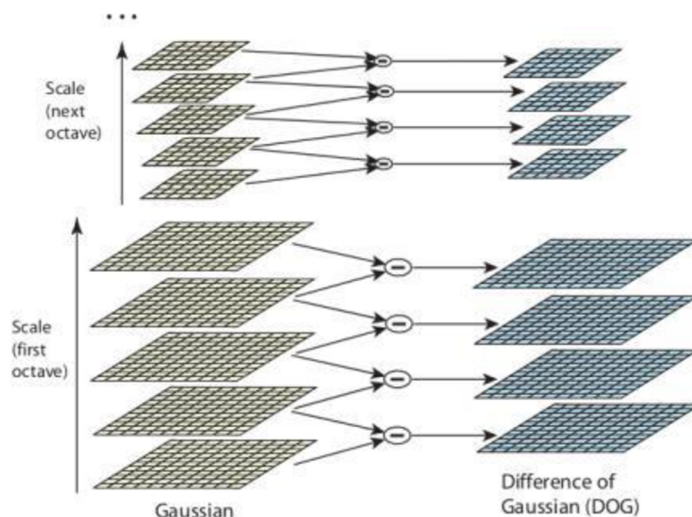
SIFT – scale-invariant feature transform, je detektor, ktorý je podľa názvu invariantný voči zmene mierky a to navyše k invariancii k rotácii a translácii. SIFT detektor je založený na použití viacerých fáz.

Prvá z týchto fáz je hľadanie extrémov v tzv. scale-space. Pri detekcii hrán podľa Harrisovej metódy je zjavné, že použitie rovnakého okna na zistenie rohov pri rozdielnych mierkach nie je možné. Na obídenie tohto problému sa teda použije scale-space. Scale-space vzniká pri použití Laplaciána Gausiána (LoG) a podvzorkovaním. LoG v tomto prípade pracuje ako jednoduchý blob detektor (blob je v tomto prípade zhuk pixelov s podobnými vlastnosťami) pre rôzne veľkosti blobov pri rôznych hodnotách σ a podvzorkovaním, kde hodnota pixelu podvzorkovanej úrovne je priemerná hodnota pixelov oblasti 2x2 predchádzajúcej úrovne, tieto úrovne sa nazývajú oktávy. Využitie Gaussovho filtru a podvzorkovania je možné vizuálne reprezentovať Gaussovou pyramídou vid' obr. 17.



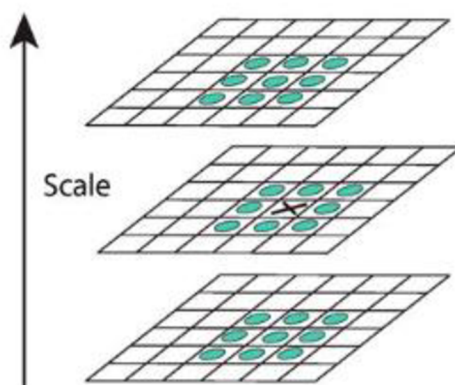
Obr. 17: Gaussova pyramída. [31]

Využitie LoG algoritmu je avšak výpočtovo náročné, preto je tento algoritmus nahradený aproximáciou pomocou rozdielu Gaussiánov (DoG). Tieto rozdiely sú medzi dvomi obrazmi, ktoré boli filtrované Gaussovým filtrom s dvomi rozdielnymi hodnotami σ . Názorná ukážka DoG je vyobrazená na obr. 18.



Obr. 18: Rozdiel Gaussiánov. [32]

Po získaní DoG sa v obrazoch DoG hľadajú maximá. Maximá sa určujú pre všetky pixely spôsobom, kde je najprv pixel porovnaný s pixelmi v jeho 8-susedstve a následne aj s pixelmi v susedných úrovniach s σ_{k+1} a σ_{k-1} , tieto susedstvá je možné vidieť na obr. 19.



Obr. 19: Extrémy DoG. [32]

Ak je tento pixel po porovnaní uznaný ako lokálny extrém, tak je možné ho označiť ako potenciálny významný bod obrazu. Po získaní týchto potenciálnych významných bodov je nutné eliminovať nechcené body, odfiltrované sú body s nízkym kontrastom. DoG je silne citlivý na hrany a tieto je tiež nutné eliminovať z výberu významných bodov. Na elimináciu hrán sa využíva koncept podobný k Harrisovmu detektoru, a to Hessova matica s rozmermi 2x2. Podobne ako pri Harrisovom detektore sa preskúmajú vlastné

čísla tejto matice. Pomer vlastných čísel Hessovej matice je možné použiť ako kritérium pre elimináciu hrán a ponechanie rohov. Na kontrolu pomeru vlastných čísel voči danému kritériu sa použije vzorec (24)

$$\frac{\text{trace}(\mathbf{H})^2}{\det(\mathbf{H})} < \frac{(r+1)^2}{r}, \quad (24)$$

kde $\text{trace}(\mathbf{H})$ je stopa a $\det(\mathbf{H})$ je determinant Hessovej matice a r je kritérium pomeru. Ďalším krokom je priradenie orientácie pre každý nájdený významný bod, priradením orientácie podľa vlastností okolia bodu sa docieli invariácia voči rotácii. Pre každý doposiaľ nevyradený bod a jeho blízke okolie $L(x,y)$ sa vypočíta veľkosť gradientu $m(x,y)$ a orientácia $\theta(x,y)$:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \quad (25)$$

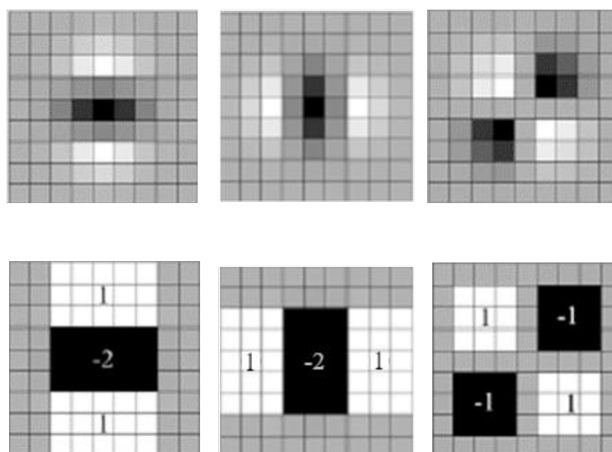
$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))). \quad (26)$$

Z orientácií je zostavený histogram s 36 intervalmi, tie rozdeľujú 360° možných orientácií. Každá vzorka pridaná do histogramu je vážená a najvyšší vrchol histogramu je vybraný spolu s vrcholmi, ktoré dosahujú 80% hodnoty najvyššieho vrcholu. Vybrané vrcholy sa účastnia na vypočítaní orientácií, týmto vzniknú viaceré body s rovnakou polohou, mierkou avšak s inou orientáciou, takéto body prispievajú k stabilite porovnávania obrazu.

Ďalej je vytvorený deskriptor významného bodu, pomocou blízkeho okolia 16x16 toto okolie je rozdelené na 16 podoblastí 4x4, pre každú z týchto oblastí sa zostaví histogram s 8 intervalmi pre orientácie, týmto postupom sa získa 128 hodnôt pre vektor opisujúci významný bod v obraze. [30,32,33]

SURF detektor

SURF detektor (Speeded-up Robust Features) podobne ako SIFT využíva aproximáciu LoG, avšak SURF na aproximáciu používa box filtre (\mathbf{D}_{xx} , \mathbf{D}_{xy} , \mathbf{D}_{yy}) túto aproximáciu je možné vidieť na obrázku 20 (hore LoG a dole aproximujúce box filtre).

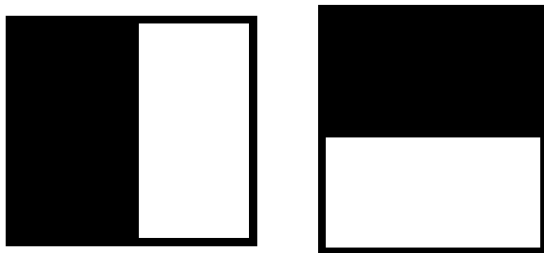


Obr. 20: Aproximácia LoG pomocou box filtrov. [34]

Aproximáciu pomocou box filtrov je možné vypočítať veľmi rýchlo za použitia integrálnych obrazov. Aproximácia Hessiána je potom podľa vzorca (27):

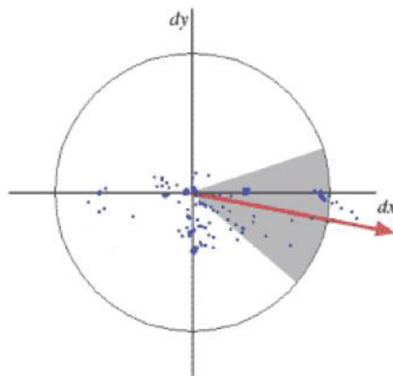
$$\det(\mathbf{H}_{approx}) = \mathbf{D}_{xx} \cdot \mathbf{D}_{yy} - (0,9 \cdot \mathbf{D}_{xy})^2. \quad (27)$$

Ďalšie vylepšenie algoritmu prináša SURF pri výpočte orientácie pre deskriptor. Pri počítaní orientácie sa využije odozvy Haarovej vlnky v smeroch x a y , pre oblasti $6s$, kde s značí mierku preskúvaného bodu. Haarove vlnky sú zobrazené na obr. 21.



Obr. 21: Haarove vlnky použité pre algoritmus SURF. [34]

Po zistení odozvy Haarovej vlnky a prepočítaním pomocou váh z Gaussiána sú tieto hodnoty zohľadnené v grafe s osami dx a dy . Smer maximálnej orientácie sa zistí sumou všetkých bodov, vo vnútri posúvajúceho sa okna o uhle 60° , toto je znázornené na obrázku 22.



Obr. 22: Orientácie okna a 60° uhol pre deskriptor SURF. [34]

Okolo skúmaného bodu sa vytvorí okolie $20s \times 20s$, kde s je opäť mierka, toto okolie je rozdelené na oblasti 4×4 . Pre každú z týchto podoblastí je nakoniec zostavený deskriptor ako vektor \vec{v} podľa (28):

$$\vec{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|). \quad (28)$$

Takto získaný vektor bude mať dĺžku 64.

Podstatné zlepšenie je získané pri použití znamienka Laplaciána. Toto označenie bodu nie je potrebné znova počítať nakoľko bol Laplacián vypočítaný už počas detekcie. Pomocou znamienka je možné odlišiť či je tmavý blob na bledom pozadí alebo naopak. Táto informácia naďalej urýchli zisťovanie zhody medzi obrazmi. [30, 33, 34]

Ďalšie druhy detektorov

Existuje ešte viacero druhov detektorov a deskriptorov významných bodov, ktoré budú vymenované, ale nebudú podrobnejšie popísané. Často používané iné detektory sú FAST, BRIEF alebo metóda ORB, ktorá bola navrhnutá developermi OpenCV, táto metóda je bezplatná alternatíva k plateným a patentovaným metódam SIFT a SURF. [30]

Hľadanie zhody významných bodov

Na zistenie zhody nájdených významných bodov sa najčastejšie používajú dve metódy. Prvou z týchto metód je Brute-force matching, táto metóda porovná vybraný deskriptor jedného významného bodu z prvého obrazu so všetkými významnými bodmi v druhom obraze a na základe výpočtu vzdialenosti vráti najbližšiu zhodu. Druhou metódou je zhoda na základe použitia FLANN (Fast Library for Approximate Nearest Neighbors). FLANN je knižnica obsahujúca algoritmy, ktoré optimalizujú rýchlosť hľadania najbližších susedov pri používaní veľkých datasetov. Použitie FLANN metód je pri veľkých datasetoch výhodnejšie ako použitie Brute-force metódy. [30]

3.2 Neurónové siete

Umelá inteligencia sa vo svete využíva čoraz viac. Umelá inteligencia a neurónové siete sú objektom výskumu aj vo využití pre bin picking a sématickú segmentáciu obrazu. Táto práca nie je zameraná na využitie umelej inteligencie, ale nakoľko je táto technológia trendom pri spracovaní obrazu je nutné opísať aspoň základy často využívaných neurónových sietí. Neurónová sieť (NN) je jeden z druhov umelej inteligencie používanej pri bin pickingu. Obrovská výhoda využitia neurónovej siete je, že je možné použiť neurónové siete nielen na rozpoznávanie objektov, ale taktiež na plánovanie cesty a uchopovanie daných objektov. Neurónová sieť je výpočtový systém, ktorý postupne zlepšuje svoje schopnosti vykonať určitú úlohu. Tieto siete využívajú jednotlivé výpočtové jednotky tzv. neuróny, tie sú usporiadané do komplexnej siete o viacerých vrstvách. Vrstvy popisujeme ako vstupné, skryté a výstupné. Podľa prepojení neurónov tiež rozoznávame doprednú neurónovú sieť a rekurentnú. Dopredné siete sú siete, v ktorých jednotlivé spojenia neurónov netvorí cyklus a teda nie sú rekurentné. Úloha každého neurónu spočíva v prenasobení vstupných parametrov s ich jednotlivými váhami, ich sčítaní a aplikácii logickej neurónovej aktivačnej funkcie, výstupom tejto funkcie je samotný výstup daného neurónu a ten môže byť využitý ako vstupný parameter spolu s ďalšími váhami do iného neurónu alebo to môže byť výstup celej siete. Pri výstupe je potrebné spraviť rozvahu o tom či daný výstup odpovedá vstupu a podľa tohto výstupu adekvátne upraviť váhy použité pri jeho výpočte. Snahou tejto úpravy váh je zlepšiť presnosť a obmedziť chybovosť systému. Tieto úpravy váh

môžeme nazvať učenie neurónovej siete. Známe spôsoby učenia neurónových sietí sú spätnoväzobné učenie, učenie s učiteľom a učenie bez učiteľa.

Spätnoväzobné učenie spočíva v získavaní spätnej väzby pri každom vykonanom úkone vo forme odmeny. Pri učení nie je známe prostredie a ani žiadna určená stratégia a teda pomocou skúšania rozdielnych možných úkonov je potrebné zistiť, ktoré z týchto činností treba vykonať za účelom získania čo najväčšej odmeny a týmto sa adaptujú váhy. Pri zložitejších prípadoch môže každé rozhodnutie ovplyvniť nielen okamžitú odmenu, ale aj odmeny za ďalšie nasledujúce možné úkony. Spätnoväzobné učenie je teda najviac ovplyvnené metódou pokus-omyl a opísanou oddialenou odmenou. Pri spätnoväzobnom učení neurčujeme metódy učenia, ale určíme problém, ktorý treba vyriešiť.

Učenie s učiteľom je metóda, kde sú váhy menené na základe predložených tréningových dát. V tréningových dátach sa hľadajú určité vzory, vďaka ktorým je možné priradiť budúcim dátam správnu triedu. Tréningové dáta sú reprezentované párami vstupov a požadovaných výstupov (triedami). Pomocou analýzy tréningových dát sa algoritmus snaží odvodiť funkciu, ktorá môže byť použitá na priradenie správnej triedy novým, budúcim dátam.

Pri učení bez učiteľa preskúmané dáta nemusia mať žiadnu triedu, preto nie je možné využiť metódu s učiteľom. Metóda slúži na nachádzanie určitých štruktúr a podobností v týchto dátach. Pomocou nájdených podobností je možné dáta roztriediť do viacerých skupín, z ktorých každá obsahuje dáta, ktoré sú podobné s dátami v ich skupine ale rozdielne od dát v iných skupinách. [35, 36, 37, 38]

Hlboké neurónové siete (deep neural networks) sú neurónové siete, ktoré majú viac ako jednu skrytú vrstvu. Týmto je možné získať viac výsledných prepojení medzi neurónami a teda zvýšenie komplexnosti a presnosti. Pri rovnakom počte neurónov je hlboká neurónová sieť možná učiť sa rýchlejšie ako sieť plytká (sieť len s jednou skrytou vrstvou). [39, 40]

Často používaný typ siete na rozpoznávanie obrazu je konvolučná neurónová sieť (CNN), ktorej štruktúra a hierarchia je podobná k nervovej sieti v zrakovom systéme zvierat a ľudí. [41, 42] Táto sieť je dopredná a používa vrstvy, v ktorých je využívaná konvolúcia na rozpoznávanie rozdielnych znakov v danom obrázku, od rozpoznania hrán pri prvých vrstvách až po rozpoznávanie komplexnejších objektov a tvarov v neskorších vrstvách. Medzi konvolyčnými vrstvami sa tiež často využívajú podvzorkovacie vrstvy na zníženie množstva dát. Konvolyčné siete sa preukázali ako schopné, úspešné a veľmi presné pri úlohách rozpoznávania obrazu. [43]

4 APLIKÁCIA PRE 2D ROBOTICKÉ VIDENIE

V tejto kapitole bude rozobraná problematika aplikácie na 2D robotické videnie, zvoleného programu a samotnej tvorbe demo aplikácie. V tejto kapitole boli využité znalosti o algoritmoch používaných pre spracovanie obrazu z kapitoly 3, pomocou týchto vedomostí bola vytvorená aplikácia, ktorá je schopná spracovať obraz a rozoznať objekt z definovanej množiny objektov.

4.1 Popis problematiky problému a zvoleného programu

Zadaná úloha spočívala vo vytvorení aplikácie pre 2D robotické videnie schopnej rozpoznania objektov z definovanej množiny. Na vytvorenie aplikácie bol zvolený program MATLAB a nasledujúce knižnice (toolboxy) funkcií Image Acquisition Toolbox [44] a Image Processing Toolbox [45]. Program MATLAB bol zvolený vďaka tomu, že jeho spomínané obsiahle knižnice majú mnohé vstavané funkcie uľahčujúce spracovanie, manipuláciu a analýzu obrazových dát.

MATLAB je platforma optimalizovaná na počítanie a riešenie rôznych problémov a rovníc, programovací jazyk MATLABu je založený na využívaní matic pri výpočtoch. Pomocou vstavaných funkcií v knižniciach programu MATLAB je možné jednoduchšie počítať a riešiť komplexné problémy vďaka využitiu mnohých známych algoritmov, ktoré boli implementované a naprogramované ako ľahko použiteľné funkcie.

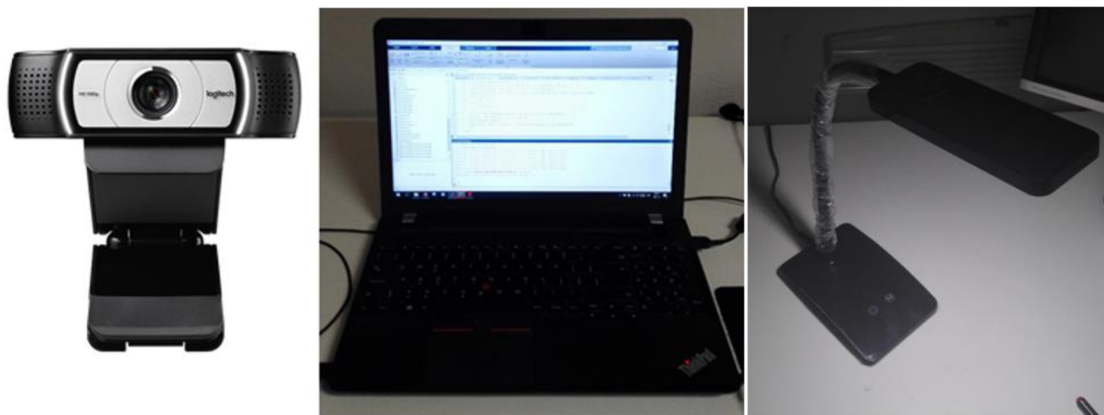
Knižnica Image Acquisition Toolbox obsahuje funkcie a bloky na pripojenie kamier a senzorov do prostredia MATLAB. Pomocou tohto toolboxu je tiež možné detegovať a ladiť rôzne vlastnosti použitého hardwaru a teda aj ním získaného obrazu alebo videa. Vďaka funkciám tejto knižnice je možné automatizovať získavanie informácií z použitého hardwaru. [44]

Knižnica Image Processing Toolbox obsahuje bohaté spektrum algoritmov a prostriedkov na spracovanie, analýzu a tiež vizualizáciu obrazu. Pomocou funkcií tejto knižnice je možné upravovať jas, kontrast, vytvárať histogramy obrazovej funkcie, segmentovať obraz, vykonávať rôzne transformácie a mnohé ďalšie operácie s obrazom. [45]

4.2 Špecifikácie použitej hardwarovej techniky na spracovanie obrazu

Prístroje použité pri spracovaní obrazu sú zobrazené na obr. 23. Tieto prístroje sú:

- Webová kamera Logitech C930e, rozlíšenie 1920 x 1080 pixelov
- Notebook ThinkPad E560, Intel® Core™ i7-6500U
- Stolná LED lampa Emos Eddy, svetelný tok 360lm a výkon 6W



Obr. 23: Hardware použitý pri počítačovom videní.

4.3 Spracovanie obrazu pomocou programu Matlab

Na získanie fotografií z webovej kamery je použitá knižnica Image Acquisition Toolbox a teda funkcie jej prislúchajúce zo zdroja [44]. V časti, kde je už získaný obraz spracovávaný, boli pre účely spracovania obrazu vytvorené funkcie kalibracia, processkalibracia, processpas a featurematch. Okrem týchto vytvorených funkcií boli pri spracovaní obrazu často využité vstavané funkcie knižnice Image Processing Toolbox [45].

4.3.1 Získanie obrazu pomocou kamery

Pre spojenie webovej kamery s MATLABom je použitý objekt `webcam`, po získaní pripojenia k webkamere je možné nastaviť rozlíšenie kamery príkazom `webcam.Resolution`, v tejto práci bolo využité rozlíšenie 640x480 pre potreby rýchleho spracovania obrazu.

4.3.2 Predspracovanie obrazu

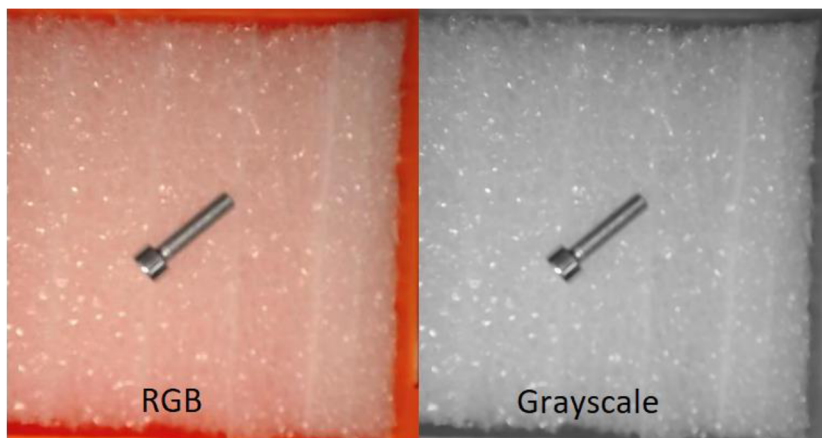
Vo vytvorenej aplikácii je možné zvoliť vstupné parametre funkcií použité na spracovanie obrazu, prípadne je možné pracovať s prednastavenými hodnotami. Po zvolení vhodných parametrov pre aktuálne podmienky osvetlenia skúmaných objektov nasleduje funkcia kalibracia, tu prebieha získanie a uloženie aktuálneho obrazu kamery, na to je využitá funkcia `snapshot`. Premenná získaná funkciou `snapshot` vyjadruje obrazovú funkciu, ktorá je uložená v poli o rozmeroch: výška obrázku, šírka a tri vrstvy pre jednotlivé zložky RGB. Po načítaní obrázkov je možné začať predspracovanie obrazu. Pri obrázkoch zmenšíme hranice na potrebnú oblasť záujmu a obrázky uložíme ako nové premenné. Na kalibrovanie programu použijeme nami vytvorenú funkciu `processkalibracia`.

Vo funkcii používame viacero vstavaných funkcií po sebe, ako prvá je `rgb2gray`, táto funkcia konvertuje vstupný obraz o troch farebných vrstvách RGB na jednu vrstvu tónov šedej farby. Výstupný obraz je matica s rozmermi danými rozlíšením obrazu

o jednej vrstve, ktorá obsahuje číselnú hodnotu odtieňu šedej farby pre jednotlivé pixely. Týmto procesom sa zachováva jas obrazu. Číselná hodnota sa získava z rovnice :

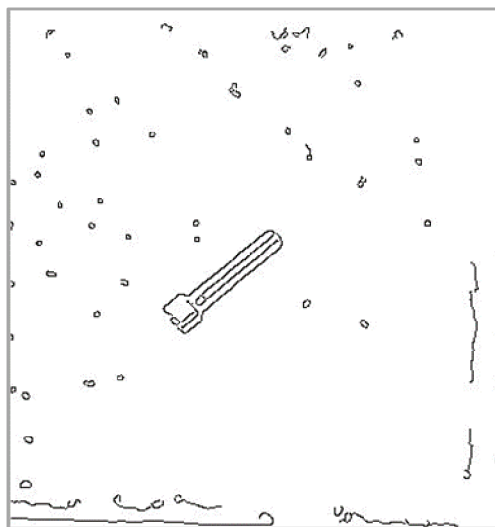
$$X = 0,2989 \cdot R + 0,5870 \cdot G + 0,1140 \cdot B , \quad (29)$$

kde X predstavuje hodnotu odtieňu šedej a R, G, B sú, v danom poradí, odtieň červenej, zelenej, modrej farby, použitie funkcie `rgb2gray` je možné vidieť na obr. 24.



Obr. 24: Použitie funkcie `rgb2gray`.

Pomocou funkcie na detekciu hrán, ktorá rozpoznáva veľké zmeny v gradiente získame obraz obsahujúci najvýraznejšie hrany. Na túto operáciu použijeme hranový detektor pomocou funkcie `edge`. Je možné použiť viacero rôznych hranových detektorov a podľa ich druhu a vstupných parametrov je možné sa zamerať na rozdielnú jemnosť hrán v obrázku a teda napríklad získať len určité hrany. V tejto práci je použitý Cannyho hranový detektor s dvomi prahmi, zvolené hodnoty týchto prahov sú 0,3 a 1. Cannyho detektor je použitý, pretože umožňuje využiť hysteréziu. Výstupom tejto operácie je binárny obraz, vid' obr. 25, kde hodnota pixelu bude 1 ak je pixel súčasťou hrany a 0 ak nie je súčasťou hrany.



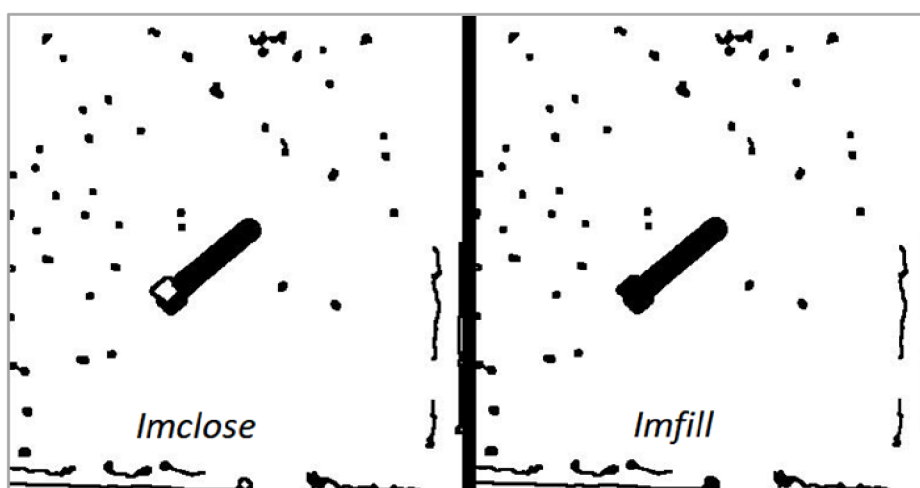
Obr. 25: Spracovanie Cannyho hranovým detektorom.

Na získanie obrazu s obrysmi objektov je nutné aplikovať na obraz rôzne morfológické operácie. Najprv obraz s hranami prejde dilatáciou, na to sa využije funkcia `bwmoprh` s parametrami 'dilate' a počet opakovaní 1, funkcia `bwmoprh` pri dilatácii využije štruktúrny element `ones(3)`. Ukážka dilatácie je na obr. 26.



Obr. 26: Dilatácia obrazu.

Dilatovaný obraz však stále zobrazuje objekty s dierami a veľkými otvorenými plochami, toto je opravené použitím morfológického zatvorenia uskutočneného funkciou `imclose` so štruktúrnym elementom vytvoreným pomocou funkcie `strel` v tvare disku s polomerom podľa nastavenia, pričom v tejto práci bol využitý polomer 1, parameter aproximácie tvaru disku je nezmenený a teda zostáva v pôvodnej hodnote 4. Po morfológickom zatvorení je možné vyplniť uzatvorené hrany funkciou `imfill`. Výstupy funkcií `imclose` a `imfill` sú znázornené na obr. 27.



Obr. 27: Morfológické zatvorenie a vyplnenie obrazu.

Po týchto operáciách je možné odstrániť všetky vyplnené oblasti, ktoré sú v kontakte s hranou obrazu, toto je možné dosiahnuť použitím `imclearborder` s parametrom

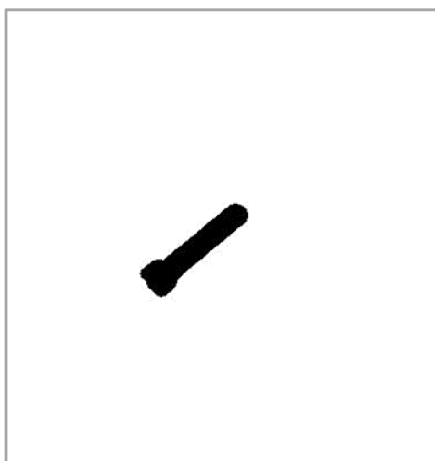
'Connectivity' nastaveným na preskúmanie 8-susedstva pixelu. Avšak stále sú na obraze problémové body šumu, ktoré neprislúchajú vyšetřovanému objektu. Problém šumu je možné vyriešiť pomocou morfológického otvorenia obrazu, teda imopen. Týmto je získaný výsledný obraz predspracovania obrazu, ktorý je možné vidieť na obr. 28. Názorná ukážka zdrojového kódu predspracovania z funkcie processkalibracia:

Výpis programu 1: Funkcia processkalibracia.

```

1: function [d10] = processkalibracia (img, canny, cannyup,
    dilate, disk, diskopen, conn)
2: %šedotónový obraz
3: grayscale=rgb2gray(img);
4: %edge detection pomocou Cannyho metódy
5: c = edge (grayscale,'canny', canny, cannyup);
6: %dilatácia hrán
7: d = bwmorph (c, 'dilate', dilate);
8: %tvorba štruktúrnych elementov
9: maticadisk=strel ('disk', disk);
10: maticadisk1=strel ('disk', diskopen);
11: %morfológické zatvorenie obrazu
12: d7=imclose (d, maticadisk);
13: %výplň uzatvorených oblastí
14: d8=imfill (d7, conn,'holes');
15: %odstránenie oblastí na hrane obrazu
16: d9=imclearborder(d8,8);
17: %morfológické otvorenie obrazu
18: d10=imopen (d9, maticadisk1);
19: imshow(d10)
20: pause
21: close all
22: end

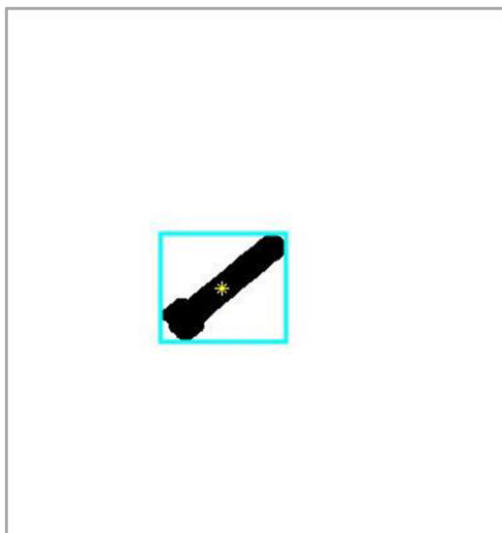
```



Obr. 28: Výsledný obraz predspracovania.

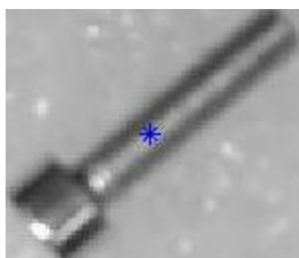
V prípade, že pri týchto opatreniach bude výsledný obraz obsahovať viacero blobov, budú dané bloby označené pomocou funkcie bwlabel. Následne budú zistené plochy týchto

označených regiónov použitím `regionprops` a najväčšia oblasť bude zvolená ako reprezentatívny objekt pre tento obraz, nakoľko sa úloha zaoberá obrazom, na ktorom je vždy iba jeden objekt. Vďaka funkcii `regionprops` je možné zistiť ťažisko oblasti a tiež súradnice s rozmermi najmenšieho možného opísaného obdĺžnika, tzv. bounding boxu, nájdený bounding box a ťažisko pre objekt sú zobrazené na obr. 29.



Obr. 29: Objekt s bounding boxom a ťažiskom.

Pomocou orezania obrázku na oblasť bounding boxu sa naďalej zmenší oblasť záujmu pre ďalšie spracovanie a hľadanie významných bodov pomocou metódy SURF. Metóda hľadania významných bodov SURF bola vybraná na základe jej invariance voči mnohým efektom a taktiež rýchlosti algoritmu, ktorú táto metóda ponúka. Vďaka menšej oblasti po zanedbaní časti obrázku mimo bounding boxu sa zníži čas potrebný na výpočty, táto novonadobudnutá oblasť spolu s ťažiskom objektu je zobrazená ako obrázok v odtieňoch šedej na obr. 30.



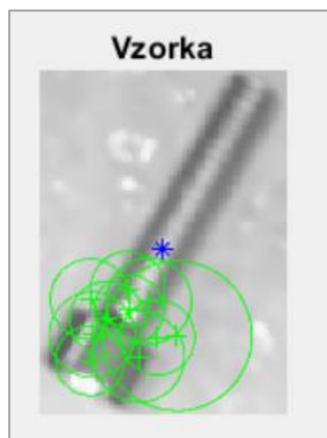
Obr. 30: Oblasť záujmu pre detekciu SURF.

4.3.3 Významné body SURF

Po dokončení predspracovania je obraz pripravený na vyhľadanie bodov SURF. V predspracovanom obraze môžeme nájsť významné body pomocou funkcie `detectSURFfeatures`. Tejto funkcii je možné upraviť viacero parametrov a to:

- 'MetricThreshold' – prah významnosti bodu, snížením tohoto prahu sa zvýši počet výstupných bodov.
- 'NumOctaves' – počet oktáv použitých pri tvorbe Gaussovej pyramídy.
- 'NumScaleLevels' – koľko rôznych stupňov sa vyskytuje v jednej oktáve.

V práci je za účelom zvýšenia počtu detegovaných bodov použitá hodnota 300 pre parameter 'MetricThreshold' a zvyšným dvom parametrom sú ponechané ich základné hodnoty 3 pre 'NumOctaves' a 4 pre 'NumScaleLevels'. Funkcia deteguje významné body SURF (Speeded-Up Robust Features) našich kalibrovaných vzorkov v oblasti bounding boxu. Tieto detegované body sa neskôr využijú pri porovnaní podobnosti kalibru s preskúvaným dielom. Výstupom `detectSURFFeatures` je objekt `SURFPoints`, tento objekt obsahuje informácie o jednotlivých významných detegovaných bodoch ako napríklad ich veľkosť, polohu, silu, atď. Na obr. 31 je zobrazených 10 najsilnejších významných bodov získaných algoritmom SURF.



Obr. 31: Významné body SURF.

Objekt `SURFPoints` nám umožňuje použiť získané dáta z detekcie ako vstupné hodnoty do funkcie `extractFeatures`, ktorá z `SURFPoints` extrahuje vektory, deskriptory, opisujúce významné body v obraze. Výstupom tejto funkcie bude matica všetkých deskriptorov a tiež bodov SURF. Výhoda algoritmu SURF je jej krátka výpočtová doba, ktorú je možné vidieť na obrázku 32.

Profile Summary

Generated 19-May-2019 12:02:09 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
detectSURFFeatures	1	0.045 s	0.005 s	
extractFeatures	1	0.040 s	0.002 s	
SURFPoints>SURFPoints.SURFPoints	2	0.029 s	0.001 s	
...Impl>SURFPointsImpl.SURFPointsImpl	2	0.027 s	0.002 s	
extractFeatures>parseInputs	1	0.026 s	0.004 s	
extractFeatures>extractSURFFeatures	1	0.013 s	0.003 s	
SURFPointsImpl>SURFPointsImpl.validate	2	0.012 s	0.004 s	
SURFPointsImpl>SURFPointsImpl.configure	2	0.009 s	0.004 s	
checkPoints	1	0.007 s	0.001 s	
detectSURFFeatures>parseInputs	1	0.007 s	0.005 s	
detectSURFFeatures>checkImage	1	0.006 s	0.001 s	
extractFeatures>parseSURFInputs	1	0.006 s	0.003 s	
FeaturePoints>FeaturePoints.subsref	6	0.006 s	0.005 s	
checkPoints>checkPointObject	1	0.006 s	0.002 s	
...ntsImpl>FeaturePointsImpl.validate	2	0.005 s	0.002 s	
validateImage	2	0.005 s	0.002 s	

Obr. 32: Doba funkcie na získanie bodov SURF.

Po získaní všetkých potrebných dát o objektoch z kalibrácie je možné získavať aktuálne dáta z webovej kamery, tieto obrázky prejdú vytvorenou funkciou `processpas`, kde bude uskutočnené spracovanie obrazu podobné ako pri obrázkoch použitých pri kalibrácii. Časťou tohoto kroku je opäť funkcia `regionprops` použitá na získanie informácií o polohe a orientácii segmentovaného objektu v obraze. Výstupmi funkcie `regionprops` s parametrami 'Centroid' bude poloha ťažiska objektu a 'Orientation' bude orientácia hlavnej osy opísanej elipsy.

Po extrahovaní dát kalibrovaných vzorkov a objektu aktuálne pod kamerou je následne možné porovnať a zistiť zhodu týchto obrázkov pomocou vytvorenej funkcie `featurematch`, kde budú detegované a extrahované body `SURFPoints` pre novo získaný obraz a tie sú následne porovnané použitím `matchFeatures`. Vstupné parametre funkcie `matchFeatures` sú:

- 'Method' – pri nastavení metódy 'Exhaustive' sú páry tvorené využitím brute-force a pri 'Approximate' sa použijú metódy hľadania najbližšieho suseda.
- 'MatchThreshold' – reprezentuje vzdialenosť od dokonalej zhody v percentách.
- 'MaxRatio' – je prah, pri ktorom sa eliminujú nejednoznačné zhody
- 'Metric' – druh výpočtu použitý na zistenie podobnosti, možnosť 'SAD' počíta so sumou rozdielov absolútnych hodnôt a 'SSD' so sumou rozdielov druhých mocnín.

- 'Unique' – voľba true, spôsobí, že každý významný bod bude viazaný len v jednej jedinečnej dvojici na rozdiel od false, kedy je možné aby bol jeden bod súčasne spojený s viacerými významnými bodmi.

Parametre 'Method', 'MaxRatio', 'Metric', 'Unique' boli ponechané na ich predvolených hodnotách a teda 'Exhaustive'; 0,6; 'SSD' a false, avšak hodnota 'MatchThreshold' bola navýšená na hodnotu 10, nakoľko boli porovnávané objekty zo známej množiny zo značnými rozdielmi. Ukážka funkcie featurematch je zobrazená vo výpise 2.

Výpis programu 2: Funkcia featurematch.

```
1: function [A, matchedPoints, matchedPoints1, points2] =  
   featurematch (d8, fkalib, vptskalib, dtctthrs, mtchthrs,  
   maxrat)  
2:   %získanie významných bodov SURF  
3:   points2 = detectSURFFeatures (d8, 'MetricThreshold',  
   dtctthrs);  
4:   %extrahovanie bodov  
5:   [f2, vpts2] = extractFeatures (d8, points2);  
6:   %nájdienie zhodných bodov  
7:   [indexPairs, matchmetric] = matchFeatures (fkalib, f2,  
   'MatchThreshold', mtchthrs, 'MaxRatio', maxrat);  
8:   matchedPoints = vptskalib (indexPairs(:, 1));  
9:   matchedPoints1 = vpts2 (indexPairs(:, 2));  
10:  %uloženie počtu zhodných párov  
11:  A=size (indexPairs, 1);  
12:  end
```

Reprezentatívny prehľad doby hľadania zhody dvoch obrázkov pomocou významných bodov SURF použitím funkcie matchFeatures je možné vidieť na obr. 33.

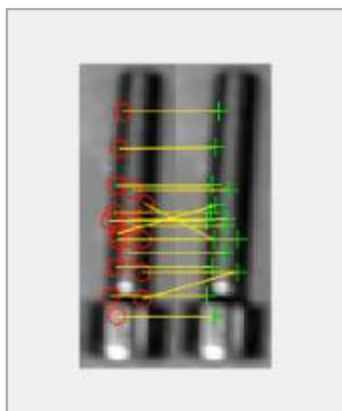
Profile Summary

Generated 19-May-2019 12:03:27 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
detectSURFFeatures	1	0.046 s	0.005 s	
matchFeatures	1	0.043 s	0.004 s	
extractFeatures	1	0.040 s	0.002 s	
SURFPoints>SURFPoints.SURFPoints	2	0.030 s	0.001 s	
...Impl>SURFPointsImpl.SURFPointsImpl	2	0.027 s	0.002 s	
extractFeatures>parseInputs	1	0.026 s	0.005 s	
cvalgMatchFeatures	1	0.018 s	0.002 s	
matchFeatures>parseInputs	1	0.017 s	0.010 s	
cvalgMatchFeatures>findMatchesExhaustive	1	0.013 s	0.002 s	
extractFeatures>extractSURFFeatures	1	0.012 s	0.004 s	
FeaturePoints>FeaturePoints.subsref	8	0.012 s	0.006 s	
SURFPointsImpl>SURFPointsImpl.validate	2	0.012 s	0.004 s	
SURFPointsImpl>SURFPointsImpl.configure	2	0.010 s	0.004 s	
checkPoints	1	0.007 s	0.001 s	

Obr. 33: Doba trvania funkcie featurematch.

Na zobrazenie párov zhodných bodov je použitá funkcia `showMatchedFeatures` s parametrom `'montage'`, týmto parametrom dosiahneme ukážku oboch obrázkov vedľa seba v jednom okne a v nich budú naznačené páry zhodných bodov. Výsledky tejto vizualizácie pri obmedzenom počte bodov je možné vidieť na obr. 34.



Obr. 34: Ukážka zhodných párov bodov.

Výstupom `matchFeatures` sú všetky zhodné páry podľa daných parametrov podobnosti. Počet nájdených zhodných párov je vďaka tomu možné vyčíslit' a porovnať tieto hodnoty pre rôzne kalibračné vzorky, podľa tejto charakteristiky je možné zistiť, s ktorým zadaným objektom počas kalibrácie sa aktuálny objekt pod kamerou najviac podobá.

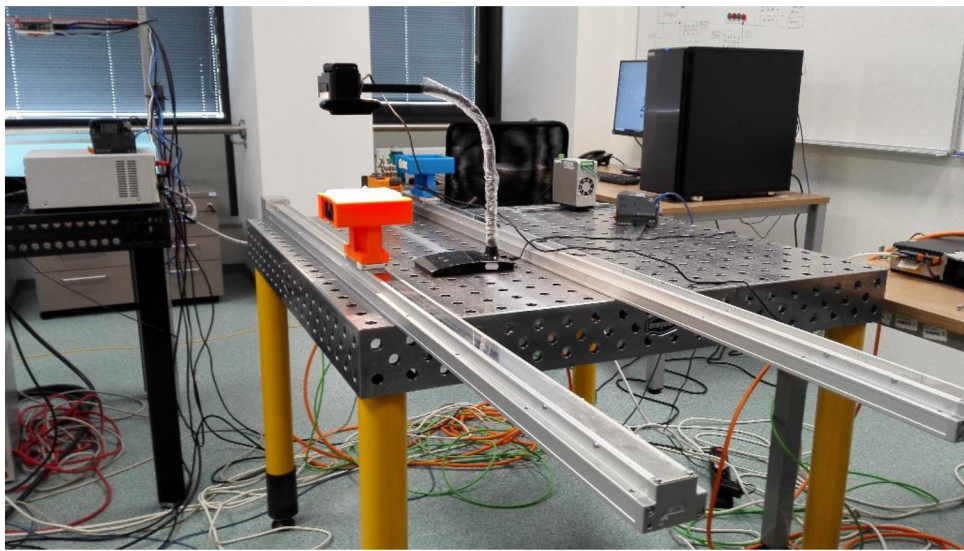
4.3.4 Overovací experiment

Praktická realizácia spočívala v overení funkčnosti vytvorenej aplikácie na reálnom probléme, kde je daný objekt umiestnený v nádobe pripevnenej na vozík jazdiaci na ose,

toto je možné vidieť na ukázkovom videu v prílohe A. Celá zostava osy bola postavená na technológii B&R a SMC, pričom súčasťou zostavy boli:

- SMC LEFB25UNZS-1400C – osa
- B&R 8LVA13.B1030D000-0 – motor
- B&R 80VD100PD.C000-01 – frekvenčný menič
- B&R X20CP1584 – PLC

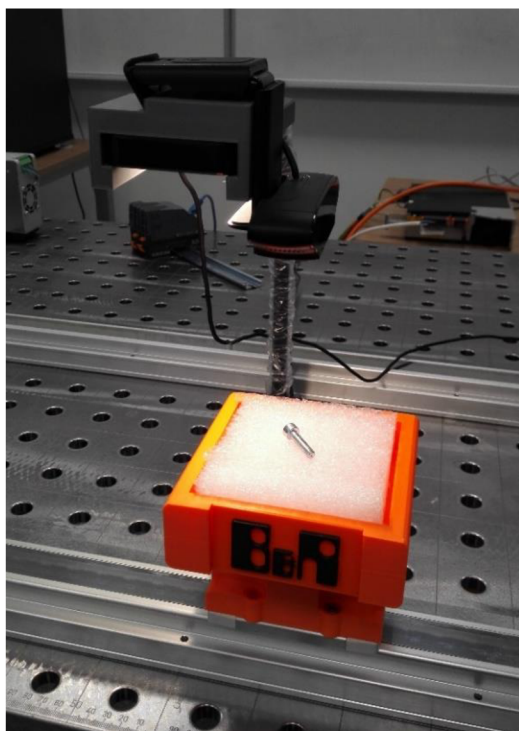
Na obrázku 35. je možné vidieť riadenú osu a snímacie zariadenie pozostávajúce z osvetľovacej lampy a webovej kamery.



Obr. 35: Pohľad na osu s kamerou a lampou.

Kamera bola upevnená nad dráhou na stolnej lampe a snímala obraz kolmo nadol. V tomto prípade bola kamera umiestnená v strede dráhy a v tomto mieste bol taktiež nastavený bod zastavenia vozíku. Pohľad spredu na zostavu dráhy je na obr. 36.

Priebeh testu spočíva v pohybe vozíku po ose a jeho pravidelnom zastavení priamo pod kamerou na dobu kratšiu ako jedna sekunda, počas tejto doby je získavaný obraz pomocou webovej kamery. Tento obraz je spracovaný a v ňom je hľadaná zhoda so vzorkami získanými z kalibrácie.



Obr. 36: Zostava kamery a osy z predného pohľadu.

Kamera umiestnená nad miestom zastavenia vozíku neprestajne posiela aktuálne obrazové dáta a nasledovne sú tieto dáta spracované. Pri spracovaní obrazu v reálnom čase je rýchlosť veľmi dôležitá, pretože sa scéna pod kamerou neustále mení. Prehľad doby predspracovania funkciou `processpas` je zobrazený na obr. 37.

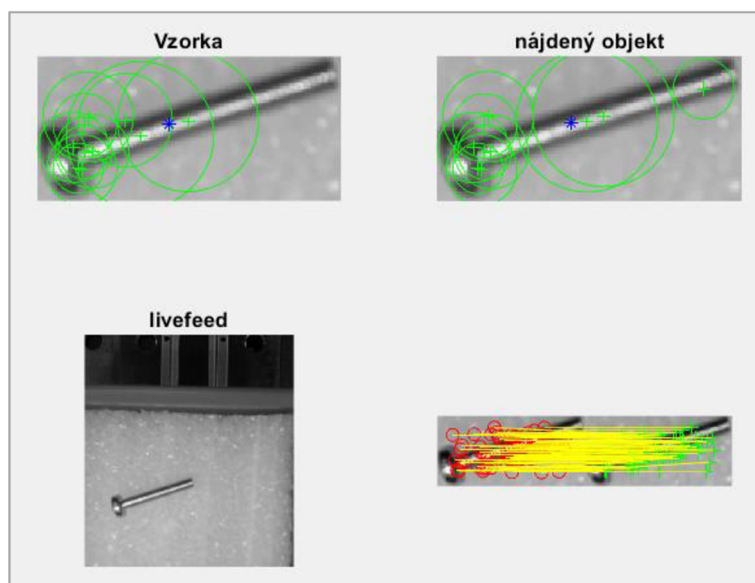
Profile Summary

Generated 19-May-2019 12:07:38 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
processpas	1	0.349 s	0.016 s	
edge	1	0.101 s	0.007 s	
regionprops	5	0.084 s	0.011 s	
imclose	1	0.051 s	0.008 s	
morphop	5	0.049 s	0.010 s	
imfill	2	0.048 s	0.008 s	
edge>smoothGradient	1	0.045 s	0.004 s	
imclearborder	1	0.037 s	0.006 s	
edge>thinAndThreshold	1	0.037 s	0.002 s	
imfilter	4	0.036 s	0.006 s	
bwselect	1	0.034 s	0.004 s	
morphop>ParseInputs	5	0.031 s	0.014 s	
imerode	3	0.029 s	0.002 s	
regionprops>createOutputStatsStructure	5	0.026 s	0.004 s	
imdilate	2	0.023 s	0.001 s	

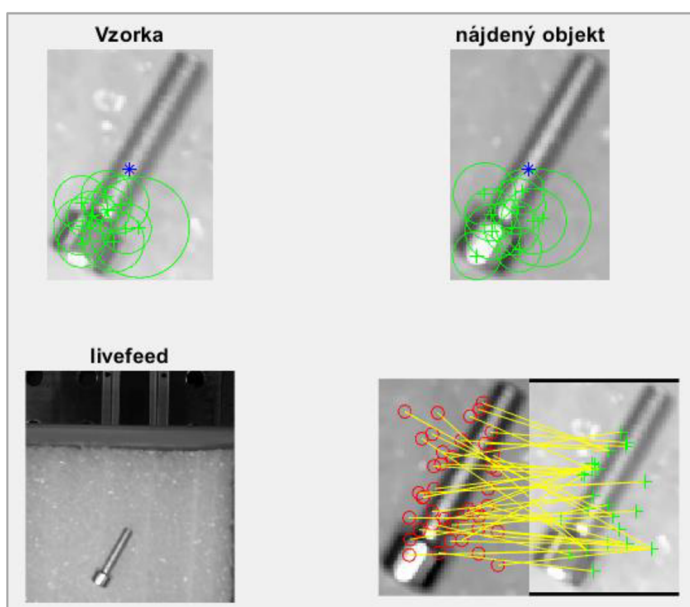
Obr. 37: Doba predspracovania obrazu.

Spracované dáta sú ďalej porovnávané s kalibrovacími vzorkami. Rýchlosť porovnávania vzorkov takýchto obrázkov bola už preskúmaná na obr. 33. Ukážka správnej detekcie objektu na idúcom vozíku je zobrazená na obr. 38.



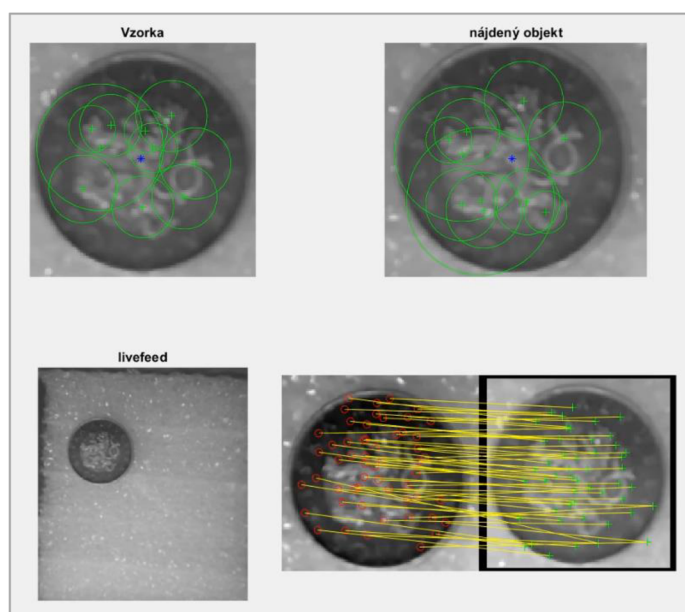
Obr. 38: Zobrazenie správnej detekcie č. 1.

Na obrázku 38 je vľavo hore zobrazená kalibračná vzorka odpovedajúca detegovanej vzorke zobrazenej v pravej hornej časti. V ľavej dolnej časti je zobrazená aktuálna snímka zachytená kamerou a v pravej dolnej časti je vizualizácia párov zhodných významných bodov vzorky a detegovaného objektu pod kamerou. Ďalšia ukážka úspešnej detekcie je na obr. 39.



Obr. 39: Zobrazenie správnej detekcie č. 2.

Aplikácia bola otestovaná pre ďalšie objekty ako napríklad minca, vid' obr. 40.



Obr. 40: Detekcia mince.

Obvyklé počty nájdených párov zhodných významných bodov sa pohybovali v hodnotách 30-80, pri rôznych umiestneniach a typoch objektov. Pri kontrole zhody kalibrovanej súčasti a objektu na zosnímanom obraze bol nastavený minimálny počet zhodných párov významných bodov na hodnotu približne v okolí jednej pätiny priemerného počtu párov, táto hodnota bola zvolená „10“. Ak bol počet nájdených párov pod touto hodnotou tak výstup programu vypísal: „Nič tu nie je.“ a daná falošná zhoda bola označená ako falošná a bola zanedbaná. Táto hodnota bola zavedená na elimináciu a obmedzenie falošných pozitívnych detekcií v momentoch, keď vozík nie je pod kamerou.

5 ZÁVER

V rámci bakalárskej práce boli preskúmané druhy a technológie bin pickingu. Pre potreby vytvorenia aplikácie bola vykonaná rešerš počítačového videnia a ním používaných algoritmov. V tejto časti práce bola podrobne rozobraná teória spracovania obrazu od získania obrazu cez rôzne metódy predspracovania obrazu, segmentácie až po detekciu významných bodov v obraze.

V praktickej časti práce bola vytvorená aplikácia pre 2D robotické videnie, pri vytvorení tejto aplikácie bolo využité prostredie MATLAB a knižnice pre získanie a spracovanie obrazu. Vytvorená aplikácia je nakalibrovaná vzorkami zo známej množiny súčastí. Tieto kalibračné vzorky sú získané z kamery pomocou knižnice Image Acquisition Toolbox a následne sú v MATLABe predspracované. Pri predspracovaní bola využitá knižnica Image Processing Toolbox a hlavnými úlohami predspracovania bolo odstrániť šum a získať segmentovaný obraz, podľa ktorého je možné obmedziť oblasť záujmu pre ďalšie úkony spracovania. Pre oblasť záujmu sú detegované významné body SURF, ktoré sú ďalej extrahované. Po získaní aktuálneho obrazu kamerou je nutné aj tento obraz spracovať. Zo spracovaného obrazu je vybraný najväčší objekt, ktorý je zvolený ako reprezentatívny pre tento obraz. Postup vybraného najväčšieho objektu je založený na tom, že v nádobe sa nachádza práve jeden objekt. Vďaka tomuto spracovaniu je možné zo segmentovaného obrazu získať polohu a rotáciu objektu. Spracovaním je taktiež nájdená užšia oblasť záujmu obsahujúca preskúmaný objekt. V oblasti záujmu sú detegované významné body a tieto body sú porovnávané s významnými bodmi získanými zo vzoriek pri kalibrácii. Prah minimálnej podobnosti pre nájdené páry bodov bol v tejto práci nastavený na hodnotu 90%. Hodnota 90% bola vybraná pre jednoduchosť a veľké rozdiely kalibračných vzorkov. Pri použití komplexnejších vzorkov je túto hodnotu možné upraviť. Podľa počtu spárovaných bodov je vybraná vzorka, s ktorou je možné vytvoriť najviac párov. Táto vzorka je vybraná spomedzi množiny ako zhodná s aktuálnym objektom pod kamerou. Ukážky niektorých detekcií boli zaznamenané ako video a sú súčasťou prílohy A.

Pre praktickú časť s využitím vozíku na pohyblivej ose bol zavedený prah minimálnej podobnosti objektu. Prah slúži na zamedzenie falošných pozitívnych detekcií objektov v momentoch, kedy sa vozík nenachádza pod kamerou. Hodnota prahu bola zvolená ako približne jedna pätina priemernej hodnoty počtu zhodných párov úspešných detekcií. Pri použití vozíku bolo nutné získavať dáta v reálnom čase a preto bola v práci preskúmaná aj rýchlosť spracovania aktuálneho obrazu a jeho porovnanie so vzorkami. Doba nutná na spracovanie obrazu je približne 0,35 sekundy a doba porovnania už spracovaného aktuálneho objektu k vzorku je približne 0,05 sekundy. Táto doba závisí od použitého hardwaru a v praxi bývajú často využité vstavané systémy ako je napríklad NVIDIA Jetson AGX Xavier, ktoré sú vytvorené na riešenie podobných problémov a sú schopné dobu spracovania znížiť.

Pred použitím programom získaných hodnôt na uchopenie dielu je nutné vopred kalibrovať kameru pre z-ovú polohu tohto dielu a pomocou tejto hodnoty a rýchlosti vozíku stanoviť súradnice, ktoré budú využité pre uchopenie dielu robotom. Pre budúce využitie je tiež možné nahradiť metódu hľadania významných bodov SURF za novšiu metódu ORB.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] *Photoneo* [online] ©2019 [cit. 2019-04-27]. Dostupné z: <https://www.photoneo.com/>
- [2] Structured vs Random Bin Picking. *Motion Control Robotics* [online] ©2019 [cit. 2019-03-24] Dostupné z: <https://motioncontrolsrobotics.com/structured-vs-random-bin-picking/>
- [3] ANANDAN, M. Tanya. Robotic Bin Picking – The Holy Grail in Sight. In: *Robotics online* [online]. 17.3.2016 [cit. 2019-03-24] Dostupné z: https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Robotic-Bin-Picking-The-Holy-Grail-in-Sight/content_id/6002
- [4] 04 Základy Priemyselnej robotiky- Základné delenie robotov. In: *Daily Automation* [online] ©2019 [cit. 2019-05-08]. Dostupné z: <http://dailyautomation.sk/01-zaklady-priemyselnej-robotiky-zakladne-delenie/>
- [5] Fanuc 3D Area Sensor [online]. Praha: Fanuc, 2017 [cit. 2019-04-19]. Dostupné z: <https://www.fanuc.eu/~media/files/pdf/products/robots/brochures/mbr-02196-ro%203d%20area%20sensor-v1-2/brochure%203d%20areasensor.pdf?la=cs>
- [6] MISHRA, Chintan a Zeeshan KHAN. *Development and Evaluation of a Kinect based Bin-Picking System* [online]. 2015 [cit. 2019-05-23]. Dostupné z: <http://mdh.diva-portal.org/smash/get/diva2:814934/FULLTEXT01.pdf>
- [7] Products. *Intel real sense* [online] ©2019 [cit. 2019-04-27]. Dostupné z: <https://www.intelrealsense.com/#Products>
- [8] HADZIMA, Jaroslav. 3D skener se strukturovaným osvětlením [online]. Brno, 2017 [cit. 2019-03-30]. Dostupné z: <http://hdl.handle.net/11012/67244>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky. Vedoucí práce Aleš Jelínek.
- [9] Random Bin Picking Software. *Euclid Labs* [online] ©2016 [cit. 2019-04-06]. Dostupné z: <https://www.euclidlabs.it/random-bin-picking-software/>
- [10] SCHYJA, Adrian a Bernd KUHLKOTTER. Realistic simulation of industrial bin-picking systems. In: *2015 6th International Conference on Automation, Robotics and Applications (ICARA)* [online]. IEEE, 2015, s. 137-142 [cit. 2019-04-06]. DOI: 10.1109/ICARA.2015.7081137. Dostupné z: <https://ieeexplore.ieee.org/document/7081137>
- [11] MAW, Isaac. New Bin Picking Solution Developed by Lean Team of Engineers. In: *Engineering.com* [online]. 23.2.2018 [cit. 2019-04-06]. Dostupné z: <https://www.engineering.com/AdvancedManufacturing/ArticleID/16539/New-Bin-Picking-Solution-Developed-by-Lean-Team-of-Engineers.aspx>
- [12] FANTONI, Gualtiero, Marco SANTOCHI, Gino DINI, et al. Grasping devices and methods in automated production processes. *CIRP Annals - Manufacturing Technology* [online]. Elsevier, 2014, 63(2), 679-701 [cit. 2019-04-27]. DOI: 10.1016/j.cirp.2014.05.006. ISSN 0007-8506. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0007850614001887>
- [13] PhoXi ® 3D Scanner In a Bin Picking Application. *Photoneo* [online] ©2019 [cit. 2019-04-27]. Dostupné z: <https://www.photoneo.com/download/phoxi-3d-scanner-in-a-bin-picking-application/?wpdmdl=4974>
- [14] Products. *Pick-it 3D* [online]. ©2019 [cit. 2019-05-23]. Dostupné z: <https://www.pickit3d.com/product>
- [15] Application areas of the Ensenso 3D cameras. *IDS Imaging* [online]. ©2019 [cit. 2019-05-23]. Dostupné z: <https://en.ids-imaging.com/ensenso-applications.html>

- [16] Great examples of automated bin picking by Pick-it 3D. *Pick-it 3D* [online]. ©2019 [cit. 2019-05-17]. Dostupné z: <https://blog.pickit3d.com/great-examples-of-pick-it-3d-bin-picking>
- [17] BROWNLEE Jason. A Gentle Introduction to Computer Vision. In: *Machine Learning Master*. [online] 19.3.2019 [cit 2019-05-17]. Dostupné z: <https://machinelearningmastery.com/what-is-computer-vision/>
- [18] HLAVÁČ, Václav a Milan ŠONKA. *Počítačové vidění*. Praha: Grada, 1992, 272 s. : il. ISBN 80-85424-67-3.
- [19] KALOVÁ, Ilona. *Předzpracování obrazu: Řetězec zpracování obrazu* [online]. [cit. 2019-04-14]. Dostupné z: <http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/02%20-%20Predzpracovani%20obrazu.pdf>
- [20] JANÁKOVÁ, Ilona. *Filtrace šumu a poruch* [online]. [cit. 2019-04-14]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/07_Filtrace_sumu_a_poruch.pdf
- [21] Lheme, Leonardo O. *Frequency Domain Bandpass Filtering for Image Processing* [online]. [cit. 2019-05-23]. Dostupné z: <https://www.scribd.com/doc/51981950/Frequency-Domain-Bandpass-Filtering-for-Image-Processing>
- [22] KRIG, Scott. *Computer Vision Metrics: Textbook Edition*. [online] Cham: Springer International Publishing, 2016, s. 35-74 [cit. 2019-04-14] DOI: 10.1007/978-3-319-33762-3. ISBN 9783319337616. Dostupné z: <https://link.springer.com/book/10.1007%2F978-3-319-33762-3>
- [23] FISHER Robert, Simon PERKINS, Ashley WALKER a Erik WOLFART. *The Hypermedia Image Processing Reference* [online]. © 2004 [cit. 2019-04-28]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm
- [24] HORÁK, Karel. *Detekce hran a rohů* [online]. [cit. 2019-05-10]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/08_Detekce_hran_a_rohu.pdf
- [25] Image Processing module. *Open Source Computer Vision* [online]. ©2019 [cit 2019-05-14]. Dostupné z: https://docs.opencv.org/4.1.0/d7/da8/tutorial_table_of_content_imgproc.html
- [26] HORÁK, Karel. *Matematická morfologie* [online]. [cit 2019-05-14]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/11_Matematicka_morfologie.pdf
- [27] JANÁKOVÁ, Ilona. *Segmentace* [online]. [cit 2019-05-16]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/Lectures/05_Segmentace.pdf
- [28] JANÁKOVÁ, Ilona. *Detekce geometrických primitiv* [online]. [cit 2019-05-16]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/Lectures/06_Detekce_geometrickych_primitiv.pdf
- [29] HORÁK, Karel. *Lokální příznaky a korespondence* [online]. [cit 2019-05-17]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/Lectures/03_Lokalni_priznaky_a_korespondence.pdf
- [30] Feature Detection and Description *Open Source Computer Vision* [online]. ©2019 [cit 2019-05-17]. Dostupné z: https://docs.opencv.org/3.4.3/db/d27/tutorial_py_table_of_contents_feature2d.html
- [31] Cmglee. Illustration of an image pyramid with 5 levels by CMG Lee. [online] CC-BY-SA-3.0. [cit. 2019-05-21] Dostupné z : https://commons.wikimedia.org/wiki/File:Image_pyramid.svg
- [32] LOWE, David. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* [online]. Boston: Kluwer Academic Publishers, 2004, **60**(2), 91-110 [cit. 2019-05-16]. DOI: 10.1023/B:VISI.0000029664.99615.94. ISSN 0920-5691. Dostupné z: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

- [33] HORÁK, Karel, Jan KLEČKA. *Deskriptory oblastí* [online]. [cit 2019-05-17]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/Lectures/05_Deskriptory_oblasti.pdf
- [34] BAY, H., T. TUYTELAARS a L. VAN GOOL. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* [online]. 2006, **3951**, s. 404-417 [cit. 2019-05-17]. DOI: 10.1007/11744023_32. ISBN 3540338322. ISSN 03029743. Dostupné z: <https://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
- [35] KŘEPELKA, Pavel. Indoor robot - řídicí neuronová síť [online]. Brno, 2009 [cit. 2019-03-31]. Dostupné z: <http://hdl.handle.net/11012/1466>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky. Vedoucí práce Luděk Žalud.
- [36] MUNZAR, Milan. Hluboké neuronové sítě v rozpoznávání obrazu [online]. Brno, 2013 [cit. 2019-03-31]. Dostupné z: <http://hdl.handle.net/11012/55010>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav počítačové grafiky a multimédií. Vedoucí práce Michal Hradiš.
- [37] SUTTON, S. Richard a G. Andrew BARTO. *Reinforcement learning*. [online] London: MIT Press, 1998, s. 3-5 [cit. 2019-03-31]. ISBN 0-262-19398-1. Dostupné z: <https://books.google.cz/books?id=CAFR6IBF4xYC&lpg=PP1&hl=sk&pg=PR15#v=onepage&q&f=false>
- [38] LIU, Bing. Web Usage Mining. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, s. 63-208. [cit. 2019-03-31] DOI: 10.1007/978-3-642-19460-3. ISBN 978-3-642-19459-7. Dostupné z: <https://link.springer.com/book/10.1007%2F978-3-642-19460-3>
- [39] DELALLEAU, Olivier a Yoshua BENGIO (2011). Shallow vs. Deep Sum-Product Networks. In: *Advances in Neural Information Processing Systems* [online] NIPS, 2011, **9(24)**, 666-674 [cit. 2019-04-06]. ISBN: 978-1-61839-599-3 Dostupné z: <https://papers.nips.cc/paper/4350-shallow-vs-deep-sum-product-networks.pdf>
- [40] BIANCHINI, Monica a Franco SCARSELLI. On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures. *IEEE Transactions on Neural Networks and Learning Systems* [online]. IEEE, 2014, **25(8)**, 1553-1565 [cit. 2019-04-06]. DOI: 10.1109/TNNLS.2013.2293637. ISSN 2162-237X. Dostupné z: <https://ieeexplore.ieee.org/document/6697897>
- [41] CHENG, Fuchao, Hong ZHANG, Wenjie FAN a Barry HARRIS. Image Recognition Technology Based on Deep Learning. *Wireless Personal Communications* [online]. New York: Springer US, 2018, **102(2)**, 1917-1933 [cit. 2019-04-06]. DOI: 10.1007/s11277-018-5246-z. ISSN 0929-6212. Dostupné z: <https://link.springer.com/article/10.1007%2Fs11277-018-5246-z>
- [42] FUKUSHIMA, Kunihiko. Neocognitron. *Scholarpedia* [online]. 2007, **2(1)**, 1717 [cit. 2019-04-06]. DOI: 10.4249/scholarpedia.1717. Dostupné z: <http://www.scholarpedia.org/article/Neocognitron>
- [43] CIREŞAN, Dan, Ueli MEIER, Jonathan MASCI a Jürgen SCHMIDHUBER. Multi-column deep neural network for traffic sign classification. *Neural Networks* [online]. Elsevier, 2012, **32**, 333-338 [cit. 2019-04-25]. DOI: 10.1016/j.neunet.2012.02.023. ISSN 0893-6080. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0893608012000524>
- [44] Image Acquisition Toolbox. *MathWorks* [online] © 1994-2019. [cit. 2019-05-18]. Dostupné z: <https://www.mathworks.com/help/imaq/>
- [45] Image Processing Toolbox. *MathWorks* [online] © 1994-2019. [cit. 2019-04-28]. Dostupné z: <https://www.mathworks.com/help/images/>

- [46] PAJDLA, Tomáš. Global optimization in camera and robot calibration: Globální optimalizace v kalibraci kamer a robotů. V Praze: České vysoké učení technické, [2017]. ISBN 978-80-01-06114-5.
- [47] KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.

ZOZNAM SKRATIEK

CAD	Computer aided design
SCARA	Selective Compliance Assembly Robot Arm
2D	Dvojmerné
3D	Trojmerné
SIFT	Scale-invariant feature transform
LoG	Laplacian of Gaussian
DoG	Difference of Gaussian
SURF	Speeded up robust features
FAST	Features from accelerated segment test
ORB	Oriented FAST and rotated BRIEF
BRIEF	Binary Robust Independent Elementary Features
OpenCV	Open source computer vision
FLANN	Fast Library for Approximate Nearest Neighbors
NN	Neural Network
RGB	Red, green, blue color model. Farebný model červená-zelená-modrá
PLC	Programmable Logic Controller

ZOZNAM OBRÁZKOV

Obr. 1	Bin picking	17
Obr. 2	Štruktúrovaný bin picking	18
Obr. 3	Semi-štruktúrovaný bin picking	18
Obr. 4	Náhodný bin picking	19
Obr. 5	Metóda štruktúrovaného osvetlenia	20
Obr. 6	Transformácia jasovej stupnice	24
Obr. 7	Vyrovnanie histogramu	24
Obr. 8	Geometrická transformácia	25
Obr. 9	Použitie pásmového frekvenčného filtra	26
Obr. 10	Konvolučný filter 5x5 s Gaussovým rozdelením	27
Obr. 11	Znázornenie dilatácie	30
Obr. 12	Názorná ukážka erózie	31
Obr. 13	Morfologické otvorenie	31
Obr. 14	Morfologické zatvorenie	32
Obr. 15	Houghova transformácia	34
Obr. 16	Konstruktia pri Houghovej transformácii	34
Obr. 17	Gaussova pyramída	37
Obr. 18	Rozdiel Gausiánov	38
Obr. 19	Extrémy DoG	38
Obr. 20	Aproximácia LoG pomocou box filtrov	39
Obr. 21	Haarove vlnky použité pre algoritmus SURF	40
Obr. 22	Orientácie okna a 60° uhol pre deskriptor SURF	40
Obr. 23	Hardware použitý pri počítačovom videní	44
Obr. 24	Použitie funkcie rgb2gray	45
Obr. 25	Spracovanie Cannyho hranovým detektorom	45

Obr. 26	Dilatácia obrazu	46
Obr. 27	Morfologické zatvorenie a vyplnenie obrazu	46
Obr. 28	Výsledný obraz predspracovania	47
Obr. 29	Objekt s bounding boxom a ťažiskom	48
Obr. 30	Oblasť záujmu pre detekciu SURF	48
Obr. 31	Významné body SURF	49
Obr. 32	Doba funkcie na získanie bodov SURF	50
Obr. 33	Doba trvania funkcie <i>featurematch</i>	52
Obr. 34	Ukážka zhodných párov bodov	52
Obr. 35	Pohľad na osu s kamerou a lampou.	53
Obr. 36	Zostava kamery a osy z predného pohľadu	54
Obr. 37	Doba predspracovania obrazu	54
Obr. 38	Zobrazenie správnej detekcie č. 1	55
Obr. 39	Zobrazenie správnej detekcie č. 2	55
Obr. 40	Detekcia mince	56

ZOZNAM PRÍLOH

PRÍLOHA A. CD-ROM

A. CD-ROM

Tab. 1: Obsah CD – hlavní adresář („\Bakalárska práca“)

Adresár	Súbor
\Bakalárska práca\	2019_BP_Robota_Jakub_191542.pdf
\Hlavný program Matlab\	info.txt BP_zdrojovy_kod.zip
\Videa\	Jazda_voziku.mp4 Detekcia_1.mp4 Detekcia_2.mp4 Detekcia_3.mp4 Detekcia_4.mp4