

Bakalárska práca



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra kybernetiky

## Plánování pohybu mobilního robotu v průmyslovém prostředí

**Adam Ukleh**

Školitel: Ing. Vladimír Smutný, Ph.D.  
Odbor: Kybernetika a robotika  
Máj 2020



## Podakovanie

Moje podakovanie patrí vedúcemu práce Ing. Vladimírovi Smutnému, Ph.D za odborné vedenie, cenné rady a usmernenia. Rád by som poďakoval svojej rodine za podporu pri štúdiu.

## Prehlásenie

Čestne prehlasujem, že som predloženú bakalársku prácu vypracoval samostatne s použitím uvedenej odbornej literatúry a zdrojov, v súlade s Metodickým pokynom o dodržiavaní etických princípov pri príprave vysokoškolských záverečných prác.

V Prahe, 1. mája 2020

## Abstrakt

Hlavným cieľom bakalárskej práce bolo navrhnúť a skonštruovať roadmapu, po ktorej by sa mobilný robot dokázal bezkolízne pohybovať. Roadmapu sme reprezentovali ako orientovaný graf. V závere práce sme sa venovali priradeniu smerov cestám vo vytvorenej roadmape. Matematická morfológia a spracovanie obrazu boli základom konštrukcie roadmapy. Práca obsahuje návrh a konštrukciu roadmapy, popis algoritmov a ich následnú implementáciu a experimentálne výsledky, ktoré dokazujú funkčnosť celkového návrhu.

**Kľúčové slová:** mobilný robot, roadmap, plánovanie, navigácia, spracovanie obrazu

**Školiteľ:** Ing. Vladimír Smutný, Ph.D.

## Abstract

The main goal of bachelor thesis was to design and construct a roadmap, on which a mobile robot would be able to move without collision. Roadmap was represented as an oriented graph. At the end of the thesis we focused on assigning directions to roads in constructed roadmap. The basic of a roadmap construction was a mathematical morphology and image processing. The thesis consists of a design and construction of a roadmap, algorithm description and its implementation and experimental results. Experimental results prove functionality of the overall design.

**Keywords:** mobile robot, roadmap, planning, navigation, image processing

**Title translation:** Mobile Robot Motion Planning in Shopfloor

## Obsah

<b>1 Úvod</b>	<b>1</b>	4.1.4 Mriežka obsadenosti (Occupancy grid map) . . . . .	14
<b>2 Automatickí mobilní roboti v priemyselnom prostredí</b>	<b>3</b>	4.2 Základy zostrojenia ciest pre mobilného robota . . . . .	14
2.1 Navigácia automatických mobilných robotov . . . . .	3	4.2.1 Využitie binárnej dilatácie . . .	14
2.2 Lokalizácia automatických mobilných robotov . . . . .	4	4.2.2 Využitie skeletonizácie . . . . .	16
2.3 Plánovanie pohybu automatických mobilných robotov . . . . .	5	4.2.3 Vytvorenie stredových čiar ciest . . . . .	17
2.4 Komunikácia v systéme automatických mobilných robotov . .	6	4.2.4 Vytvorenie finálnych ciest . . .	18
<b>3 Definícia a analýza problému</b>	<b>7</b>	4.3 Dátová reprezentácia roadmapy .	18
3.1 Definícia problému . . . . .	7	<b>5 Implementácia riešenia</b>	<b>21</b>
3.2 Analýza problému . . . . .	8	<b>6 Experimenty</b>	<b>23</b>
<b>4 Návrh riešenia</b>	<b>9</b>	6.1 Vytvorenie ciest pre mobilného robota . . . . .	23
4.1 Použité postupy a metódy . . . . .	9	6.1.1 Úprava vstupných dát . . . . .	24
4.1.1 Skeletonizácia . . . . .	9	6.1.2 Základy zostrojenia ciest pre mobilného robota . . . . .	26
4.1.2 Matematická morfológia . . . . .	11	6.1.3 Vytvorenie stredových čiar ciest . . . . .	30
4.1.3 Graf . . . . .	12	6.1.4 Vytvorenie finálnych ciest pre mobilného robota . . . . .	33

6.2 Vytvorenie dátovej reprezentácie cest ..... 35	35
6.2.1 Detekcia vrcholov ..... 36	36
6.2.2 Algoritmus na detekciu vrcholov ..... 36	36
6.2.3 Algoritmus na koštrukciu podgrafu ..... 37	37
6.2.4 Príprava orientovaného grafu 41	41
<b>7 Záver</b> 43	<b>43</b>
<b>Literatúra</b> 45	<b>45</b>
<b>A Obsah CD</b> 49	<b>49</b>
<b>B Zadanie práce</b> 51	<b>51</b>

## Obrázky

4.1 Definícia v zmysle najväčších diskov (a). Najväčšie disky v danom tvare a skeleton ako stred všetkých maximálnych diskov (b); prevzaté z [11] .....	10
4.2 Vľavo je príklad neorientovaného grafu a na pravo je príklad orientovaného grafu.....	13
4.3 Obrázok naľavo sú prekážky v mriežke obsadenosti pred aplikáciou binárnej dilatácie. Obrázok napravo sú prekážky v mriežke obsadenosti po aplikácii binárnej dilatácie.....	15
4.4 Očakávaný výsledok skeletonizácie. ....	16
4.5 Hore mriežka obsadenosti bez stredových čiar ciest. Dole mriežka obsadenosti po vytvorení stredových čiar ciest. ....	17
6.1 Naľavo je pôvodná mriežka obsadenosti. Napravo je mriežka obsadenosti po úpravách. ....	25
6.2 Robot Jackal. ....	26
6.3 Naľavo je zobrazená mriežka obsadenosti pred aplikáciu binárnej dilatácie. Napravo je mriežka obsadenosti po aplikácii binárnej dilatácie (výsek medzi 70 m-110 m). ....	28
6.4 Výsledné kostry spĺňajúce podmienku na dĺžku, zobrazené červenou farbou (výsek medzi 80 m-100 m). ....	29
6.5 Naľavo je zobrazená mriežka obsadenosti pred aplikáciu dvojitej binárnej dilatácie. Napravo je mriežka obsadenosti po aplikácii dvojitej binárnej dilatácie (výsek medzi 70 m - 110 m).....	31
6.6 Kostry reprezentujúce stredové čiaru ciest, zobrazené červenou farbou (výsek medzi 70 m-100 m). ....	32
6.7 Porovnanie pôvodnej mriežky obsadenosti naľavo s mriežkou obsadenosti s vytvorenými stredovými čiarami napravo (výsek medzi 70 m-110 m). ....	33
6.8 Naľavo je zobrazená mriežka obsadenosti obsahujúca stredové čiaru pred aplikáciou binárnej dilatácie. Napravo je mriežka obsadenosti so stredovými čiarami po aplikácii binárnej dilatácie (výsek medzi 70 m-110 m). ....	34
6.9 Kostry reprezentujúce finálnu roadmapu, znázornené červenou farbou (výsek medzi 80 m-100 m). ....	35
6.10 Horná časť obrázku znázorňuje koncové body ciest. Dolná časť obrázku znázorňuje križovatky. ...	36
6.11 Detekované vrcholy na cestách, označené zelenou farbou (výsek medzi 90 m-95 m na výšku a 0 m-10 m na šírku).....	37

6.12 Na hornom obrázku sú tyrkysovou farbou vyznačené spoločné pixely prvotnej a finálnej cesty. Na dolnom obrázku sú modrou farbou vyznačené cesty, ktoré sú obojsmerné a zelenou farbou sú vyznačené cesty, ktorým užívateľ môže priradiť smer ľubovoľne. ....	42
--	----





# Kapitola 1

## Úvod

Od nepamäti sa ľudia snažili ulahčovať a zrýchľovať si prácu. Hľadali rôzne spôsoby a možnosti, ktorými by sa preprava materiálu a konkrétne pracovné činnosti, stali efektívnejšími.

V dnešnej dobe sa väčšina výrobných tovární a skladov automatizuje. Pracovné činnosti, ktoré predtým ľudí zafažovali, znižovali efektivitu alebo boli nebezpečné, preberajú na seba roboti. Jednými z kategórie mobilných robotov, ktoré pomáhajú ľuďom v skladoch a továrňach, sú automatické mobilné vozíky. Automatické mobilné vozíky ulahčia prepravu tovaru, bezpečnosť pri jeho prevážaní a ľudská činnosť sa môže presunúť na efektívnejšie pracovné činnosti.

Problematikou, ktorou sa v práci zaoberáme, si môžeme predstaviť v podobe problému, s ktorým sa často stretávajú cestári. Bežnou praxou cestárov je, že dostanú zadaný priestor, po ktorom sa autá môžu pohybovať. Ich úlohou je namaľovať čiary, ktorými vytvoria cesty. Našou úlohou v práci bolo vytvoriť takéto cesty na reprezentácii priestoru v podobe mriežky obsadenosti. Výsledné cesty reprezentujeme ako orientovaný graf, čím pripravíme potrebné podmienky pre plánovanie pohybu mobilných robotov, po nami vytvorených cestách.

Osnova našej práce sa skladá z viacerých častí. Na začiatku uvedieme čitateľa do systému automatických mobilných robotov. Následne predstavíme metódy, ktoré sme v našom riešení využívali; predložíme naše riešenie a popis algoritmov; preukážeme finálnu implementáciu, výsledky nášho skúmania a riešenia problému. Záverom práce bude zhrnutie nadobudnutých poznatkov.



## Kapitola 2

### Automatickí mobilní roboti v priemyselnom prostredí

Prvý automatický mobilný vozík vznikol už v 50. rokoch 20. storočia. Bol to odťahový vozík, ktorý sledoval drôt v podlahe ako náhradu koľajnice [1]. V súčasnosti sú automatickí mobilní roboti považovaní za najvhodnejší spôsob prepravy tovaru v skladoch, z miesta vyzdvihnutia na miesto doručenia.

Riadiaci systém automatických mobilných robotov sa skladá z troch hlavných častí. Prvá časť systému je navigácia a lokalizácia, ktorá slúži na to, aby sa automatickí mobilní roboti vedeli pohybovať a lokalizovať po halách, v ktorých sú zavedené. Druhou časťou je plánovanie pohybu, aby sa automatickí mobilní roboti vedeli dostať do vopred určeného cieľa. Tretiu časť reprezentuje komunikácia medzi robotom a centrálnou jednotkou, ktorá zadáva robotovi úlohy na vykonanie [2].

#### 2.1 Navigácia automatických mobilných robotov

Mobilný robot, rovnako ako človek, sa v okolitom prostredí musí vedieť orientovať. Na navigáciu pre automatických mobilných robotov v skladoch a halách môžeme využívať jeden alebo viacero z nasledujúcich, v súčasnosti najčastejšie používaných mechanizmov:

- Navigácia pomocou magnetov - funguje na základe magnetov, ktoré sú umiestňované do podlahy alebo pod povrchy, ktoré nezabraňujú vodivosti magnetu. Referenčné body sú uložené v počítači. Na detekciu týchto magnetov sa používa magnetický senzor, ktorý je umiestnený na spodnej strane automatického mobilného robota [3].
- Navigácia pomocou čiar - tento princíp funguje na základe vopred nainštalovanej magnetickej pásky, ktorú robot so senzorm na spodnej strane sleduje [3].
- Navigácia pomocou QR kódu - barcode/QR kódy sú umiestnené na zemi alebo na stenách, prípadne na regáloch. Automatický mobilný robot detekuje tieto kódy pomocou laserovho skenera a vypočíta svoju pozíciu porovnaním kódu s mapou v počítači [3].
- Navigácia pomocou laseru - reflexná páska je umiestnená na fixované miesta a je detekovaná otáčajúcim sa laserom. Robot má informácie o prostredí, v ktorom sa pohybuje, uložené v pamäti [3].

## 2.2 Lokalizácia automatických mobilných robotov

Dôležitú úlohu v systéme automatických mobilných robotov zohráva lokalizácia. Mobilný robot sa potrebuje simultánne lokalizovať a vytvárať pritom mapu svojho okolia. Táto metóda sa nazýva Simultánna lokalizácia a mapovanie (SLAM) [4]. Mobilný robot využíva sadu nástrojov, ktoré mu poskytujú potrebné informácie k určeniu polohy.

Jedným z týchto nástrojov je **odometria**, ktorá zbiera informácie zo snímačov pohybu na kolesách či pásoch a dokáže určiť relatívnu polohu robota od jeho počiatocnej polohy. Ďalším je **inerciálna meriaca jednotka** (IMU), ktorá využíva kombináciu akcelerometra, gyroskopu a prípadne magnetometru a poskytuje údaje o lineárnom a rotačnom zrýchlení robota. Činnosť vyššie uvedených nástrojov sa zameriava na meranie vnútorného stavu robota. Ďalším nástrojom využívaným pri lokalizácii robota je **laserový dialkomer** (LiDAR), ktorého funkcia sa od ostatných líši v tom, že meria vzdialenosť medzi robotom a predmetmi v jeho okolí.

Lokalizáciu polohy robota môžeme deliť na dva možné scenáre [5]:

Prvým scenárom je prípad, ak robot nemá žiadne apriórne informácie o svojej polohe. Príkladom je spustenie robota v novom prostredí alebo zlyhanie

navigácie. Tento prvý scenár je nazývaný globálna lokalizácia a zameriava sa na hľadanie polohy v celom priestore polôh. Tento typ lokalizácie funguje na princípe zistenia aktuálnej pozície robota, na základe dát zo senzorov a referenčnej polohy. Tieto kroky opakujeme do doby, pokiaľ nebude možné jasne určiť polohu, v ktorej sa robot nachádza.

Druhým scenárom je prípad, v ktorom na začiatku lokalizácie vieme približnú začiatočnú polohu robota v priestore. Tento scenár je nazývaný lokálna lokalizácia a jej princípom je kompenzovanie odometrických chýb v priebehu navigácie. Lokálna lokalizácia sa zameriava na hľadanie polohy od najbližšej známej polohy robota.

Dôležitý rozdiel medzi týmito dvoma technikami je v tom, že lokálna lokalizácia sa nedokáže vysporiadať s vážnymi chybami pri hľadaní polohy.

## 2.3 Plánovanie pohybu automatických mobilných robotov

Schopnosť plánovať cestu od aktuálnej pozície robota do cieľovej pozície, a súčasne sa vyhýbať prekážkam v okolí, je dôležitou vlastnosťou systému automatických mobilných robotov. Naplánovaná cesta musí spĺňať optimalizačné kritéria, ako sú dĺžka naplánovanej cesty, čas potrebný na prechod robota po naplánovanej ceste a spotreba energie. Hlavným kritériom je dĺžka naplánovanej cesty, pretože má priamy vplyv na čas a spotrebovanú energiu.

Plánovanie ciest pre mobilného robota môžeme rozdeliť na globálne (offline) a lokálne (online) plánovanie [6]:

Globálne plánovanie ciest môžeme vykonávať, ak máme vopred k dispozícii informácie o prostredí robota, v ktorom bude pohyb vykonávať. Tento typ plánovania sa vykonáva ešte pred tým, než sa robot začne hýbať. Takto naplánovaná cesta dokáže robota previesť okolitým prostredím a vyhnúť sa statickým objektom.

Lokálne plánovanie ciest sa vykonáva v reálnom čase, v priebehu prechodu mobilného robota prostredím. Využíva sa v dynamickom prostredí, kde sa môžu vyskytovať pohyblivé objekty alebo nečakané prekážky. Lokálne plánovanie je závislé od senzorov robota, ktoré zachytávajú objekty v jeho okolí. Náležitou vlastnosťou lokálneho plánovania musí byť jeho rýchla odozva.

Systémy automatických mobilných robotov v súčasnosti využívajú globálne aj lokálne plánovanie. Kombinácia týchto dvoch metód sa nazýva hybridné plánovanie [7].

## 2.4 Komunikácia v systéme automatických mobilných robotov

Pre určenie cieľa pohybu robota je potrebné, aby sme túto činnosť dokázali vykonávať diaľkovo. Na určenie cieľa v systéme automatických mobilných robotov slúži centrálny počítač, v ktorom pomocou rozhrania vieme nakonfigurovať cieľ robota. Centrálny počítač komunikuje s mobilným robotom na diaľku [8].

## Kapitola 3

### Definícia a analýza problému

#### 3.1 Definícia problému

V našej bakalárskej práci sme sa zaoberali vytvorením virtuálnej roadmapy a jej následnou dátovou reprezentáciou. Roadmapa je súbor virtuálnych ciest, po ktorých sa robot môže pohybovať. V prípade, že sa začiatočná prípadne cieľová pozícia robota nenachádza na roadmape, potrebujeme pohyb robota naplánovať lokálne k najbližšej virtuálnej ceste. Lokálnym plánovaním sa zabezpečí, že sa robot presunie zo štartovacej pozície na roadmapu, po ktorej cestuje do približnej pozície cieľa. Z tejto približnej pozície cieľa zide z roadmapy a lokálnym plánovaním sa zabezpečí jeho pohyb do presnej pozície cieľa, ktorý mu je na začiatku zadaný [9].

Naším cieľom bolo to, aby sa proces vyrobenia roadmapy zautomatizoval a nebolo potrebné manuálne zakresľovať a prepočítavať možné cesty v priestore robota.

Náš prístup berie do úvahy rozmery mobilného robota, čo zabezpečí, aby robot pri prechádzaní nevošiel na miesta, cez ktoré neprejde alebo odbočí príliš blízko prekážky a narazí do nej. Roadmapa bude koncipovaná tak, aby bolo možné cestám priradiť smer, po ktorom sa majú roboti pohybovať. V prípade, že sa mobilný robot vojde do určitého priestoru viac ako jedenkrát, bude dané miesto považované za viac prúdu cestu.

Popísaný prístup má viaceru výhod. V prvom rade dokáže zvýšiť bezpečnosť mobilného robota a jeho okolitého priestoru. Ďalšou výhodou je, že dokáže znížiť náklady na zavedenie systému automatických mobilných robotov do priemyslových prostredí. Tým, že nebude závislý na fyzických pomôckach pre navigáciu, bude lepšie udržiavateľný a urýchli čas na inštaláciu systému. Pri využití viacerých robotov sa zvýši prepravná kapacita.

Ďalším cieľom bolo roadmapu reprezentovať v dátovej reprezentácii, ktorá sa následne bude dať využiť na globálne plánovanie pohybu mobilného robota.

## 3.2 Analýza problému

V súčasných inteligentných priemyselných prostrediach sa využívajú roadmapy, ktoré sú navrhované manuálne. Tento úkon si vyžaduje špecialistu, ktorý vytvorí roadmapu v CAD softwari. Vytvorenie takejto roadmapy je časovo veľmi náročné a výsledok sa často líši od ideálu. Ideál v tomto prípade je maximálne pokrytie miest, kde mobilný robot môže vykonávať svoj pohyb.

Existuje mnoho metód na automatické vyrobenie roadmapy. Algoritmy, kolektívne nazývané ako pravdepodobnostné roadmapové metódy, sa ukázali ako dobre fungujúce v mnohých praktických využitíach [9]. Sú koncipované tak, že berú do úvahy iba jedného robota. Ideou za týmito metódami je vytvorenie grafu náhodne vygenerovaných konfigurácií bez kolízií s okolitými prekážkami.

Výhodou ich využitia je, že sú rýchle a ľahko implementovateľné. Nevýhodou je však nižšia efektívnosť v určitých situáciách, ako sú napríklad úzke priestory. Ďalšou nevýhodou uvedených metód je, že neberú ohľad na rozmery mobilného robota.



## Kapitola 4

### Návrh riešenia

V tejto kapitole predstavíme náš návrh pre automatickú konštrukciu roadmapy na mriežke obsadenosti, ktorá je reprezentovaná ako binárny obraz.

Naším cieľom je pomocou binárnej dilatácie a skeletonizácie skonštruovať virtuálne cesty pre mobilného robota. Od binárnej dilatácie očakávame rozšírenie prekážok, a tým následnú detekciu miest, po ktorých sa mobilný robot dokáže bezkolízne pohybovať. Skeletonizáciu využijeme na reprezentovanie voľného priestoru a vytvorenie ciest. Následne popíšeme vytvorenie dátovej reprezentácie skonštruovaných ciest v podobe neorientovaného grafu. V neposlednom rade popíšeme postup prípravy orientovaného grafu, ktorý bude reprezentovať smery ciest.

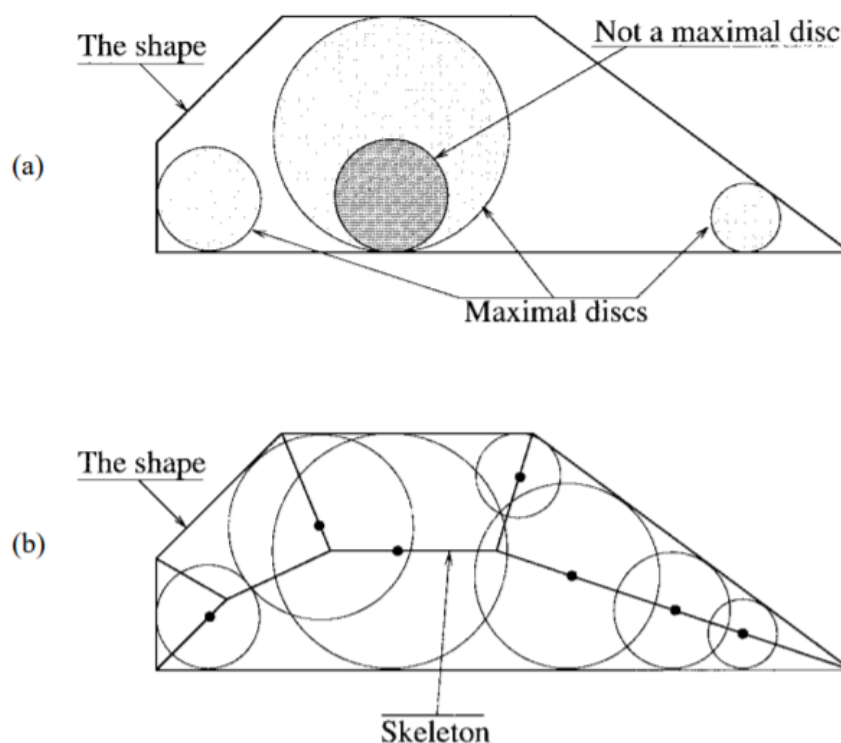
#### 4.1 Použité postupy a metódy

##### 4.1.1 Skeletonizácia

Skeleton, inými slovami kostra, je považovaný za morfológickú reprezentáciu binárnych obrazov. Skeletonická reprezentácia bola vyvíjaná nezávisle od matematickej morfológie [10]. Na vyrobenie skeletonu nám postačia aj štandardné morfológické operácie. Vďaka tomu ju môžeme považovať za morfológickú reprezentáciu umožňujúcu analýzu obrazov morfológickými nástrojmi.

Intuitívnu definíciu skeletu je podľa Bluma [11], možné si predstaviť ako trávový záhon bez vopred definovaného tvaru. V čase  $t = 0$  celú hranicu tohto záhonu zapálime a požiar sa začne šíriť od hranice záhonu dovnútra konštantnou rýchlosťou. Súbor bodov, v ktorých oheň zhasne, predstavuje kostru tvaru záhonu.

Od intuitívnej definície sa môžeme presunúť k matematickej definícii. V dnešnej dobe sú k dispozícii rôzne tvary matematických definícií skeletu. Každá definícia má rôzny tvar modelu “horiaceho záhonu” a vykazujú ekvivalentné výsledky pre spojité planárne tvary. Spoločná definícia matematického modelu skeletu znie: “Nech je maximálny kruh vpísateľný do tvaru  $X \subset R^2$ , pričom kruh patrí do tvaru  $X$ , ale nie je súčasťou iného kruhu tvaru  $X$ . Kostra tvaru  $X$  je množina stredov všetkých jeho maximálnych kruhov.” [10] Schématické znázornenie definície je na Obrázku 4.1.



**Obrázok 4.1:** Definícia v zmysle najväčších diskov (a). Najväčšie disky v danom tvare a skeleton ako stred všetkých maximálnych diskov (b); prevzaté z [11]

Morfologická teória skeletonu sa v literatúre vyvíjala v dvoch vetvách, ako poznamenal Serra [12]. Prvá vetva bola z topologického hľadiska a druhá vetva bola z algebraického hľadiska.

Z topologického hľadiska kostru tvaru môžeme vidieť ako tenkú, topologicky zachovávanú karikatúru tvaru, ktorá je užitočná pre rozpoznávanie tvarov a analýzu obrazov.

Z algebraického hľadiska je kostra výsledkom dekompozície danej množiny do superpozície jednoduchších elementov, vybraných z preddefinovanej rodiny prvkov (disky zväčšujúcich sa rozmerov). Táto dekompozícia nám poskytuje obrazovú reprezentáciu, ktorá pozostáva z kolekcie bodov patriacich do kostry a rádiusov korešpondujúcich s maximálnymi kruhmi. Rekonštrukcia originálneho obrazu je získaná zjednotením všetkých maximálnych kruhov.

## ■ 4.1.2 Matematická morfológia

### ■ Binárna dilatácia

Binárna dilatácia patrí pod záštitu matematickej morfológie. Matematická morfológia sa zaoberá analýzou textúr a útvarov v obrazoch. Analýza sa vykonáva na binárnych obrazoch.

Morfologické operácie sa vykonávajú na dvoch obrazoch. Prvý sa nazýva aktívny obraz ( $A$ ) a druhý sa nazýva štrukturovací element ( $B$ ). Štrukturovací element si môžeme predstaviť ako sondu alebo filter, ktorý aplikujeme na aktívny obraz. Aplikáciou štrukturovacích elementov rôznych tvarov a veľkostí, dokážeme upravovať aktívny obraz.

Binárna dilatácia vzniká spojením dvoch množín, na základe vektorového sčítania ich prvkov. Prvý krát bola uvedená Minkowským, po ktorom je aj často nazývaná Minkowskeho súčet. Binárna dilatácia obrazu  $A$  obrazom  $B$  je množina všetkých možných vektorových súčtov párov prvkov, jedny pochádzajú z obrazu  $A$  a druhé pochádzajú z obrazu  $B$ . Binárnu dilatáciu definujeme ako: Nech  $A, B \subset E^n$ , binárna dilatácia  $A$  podľa  $B$ , označovaná ako  $A \oplus B$ . Matematická definícia [13]:

$$A \oplus B = \{c \in E^n \mid c = a + b, \text{ pre niektoré } a \in A \text{ a } b \in B\} \quad (4.1)$$

Obrazy A a B sú v binárnej dilatácii voči sebe symetrické. Interpretáciu pre binárnu dilatáciu si môžeme predstaviť tak, že je to miesto všetkých stredov  $c$ , takých, že priložením počiatku B na  $c$  sa dotkneme množiny A.

### ■ Hit or Miss transformácia

Hit or Miss transformácia (triaf či miň transformácia) je metódou využívanou na detekovanie objektov so špecifickým tvarom [14]. Transformácia je odvodená pomocou erózie. Stredový pixel pod maskou štrukturovacieho elementu Hit je označený hodnotou 1, ak tvar pod maskou štrukturovacieho elementu je zhodný s tvarom skúmaným v binárnom obraze. V prípade štrukturovacieho elementu Miss nadobúda pixel hodnotu 0. Na detekciu špecifických tvarov je potrebná kombinácia dvoch erózií.

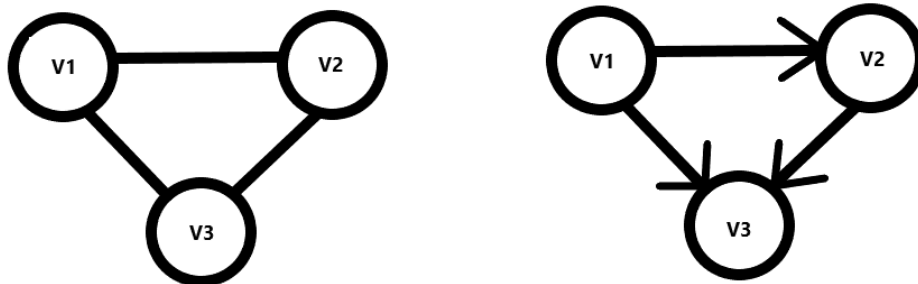
Transformáciu Hit or Miss môžeme matematicky definovať ako:

$$A(*)B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad (4.2)$$

s podmienkou, že  $B_1 \cap B_2 \neq \emptyset$  kde  $B_1$  je štrukturovací element Hit a  $B_2$  je štrukturovací element Miss [14].

### ■ 4.1.3 Graf

Všeobecne je graf G reprezentovaný usporiadanou dvojicou  $G = (V, E)$ . Usporiadaná dvojica pozostáva z množiny V, obsahujúcej N vrcholov alebo uzlov a množiny  $E \subseteq V \times V$  pozostávajúcej z hrán alebo oblúkov [15]. Grafy môžeme rozdeliť do dvoch hlavných kategórií, ktorými sú neorientovaný a orientovaný graf. Znázornenie kategórií grafov je na Obrázku 4.2.



**Obrázok 4.2:** Vľavo je príklad neorientovaného grafu a na pravo je príklad orientovaného grafu.

V neorientovanom grafe sú prepojenia medzi jednotlivými vrcholmi obojsmerné. Pre každú dvojicu vrcholov  $i$  a  $j$  platí, že ak existuje hrana  $i \rightarrow j$ , tak potom existuje aj hrana  $j \rightarrow i$ . Neorientované grafy sú zvyčajne využívané na modelovanie explicitných obojsmerných prepojení [15].

V orientovanom grafe je každej hrane priradený smer. To znamená, že hrana  $i \rightarrow j$  je rôzna od hrany  $j \rightarrow i$ . Orientovaný, ohodnotený (vážený) graf je orientovaný graf, ktorého hranám je priradená hodnota (váha). Orientovaný graf je silne súvislý ak obsahuje orientovanú cestu, ktorá spája každý pár vrcholov grafu [15].

Podgraf grafu  $G$  je graf  $Y$ , pre ktorý platí [15]:

$$V(Y) \subseteq V(G) \text{ a } E(Y) \subseteq E(G) \quad (4.3)$$

Hrana sa nazýva mostom, ak po jej odobraní z grafu sa zvýši počet súvislých komponent [15].

Komponenta grafu (súvislá komponenta grafu)  $G$  je súvislý podgraf grafu  $G$ , ktorý nie je obsahom žiadneho väčšieho súvislého podgrafu  $G$  [15].

#### ■ 4.1.4 Mriežka obsadenosti (Occupancy grid map)

Hlavnou myšlienkou mriežky obsadenosti je reprezentácia 2D alebo 3D priestoru, do mriežky obsahujúcej bunky.

Každá bunka obsahuje pravdepodobnostný odhad jej stavu. Premenná stavu je definovaná ako diskrétna náhodná premenná s dvoma stavmi, ktoré nadobúdajú hodnoty: obsadená alebo prázdna (voľná) [16].

Mriežku obsadenosti, ktorú používame v našej práci vytvoril Nováček [17].

### ■ 4.2 Základy zostrojenia ciest pre mobilného robota

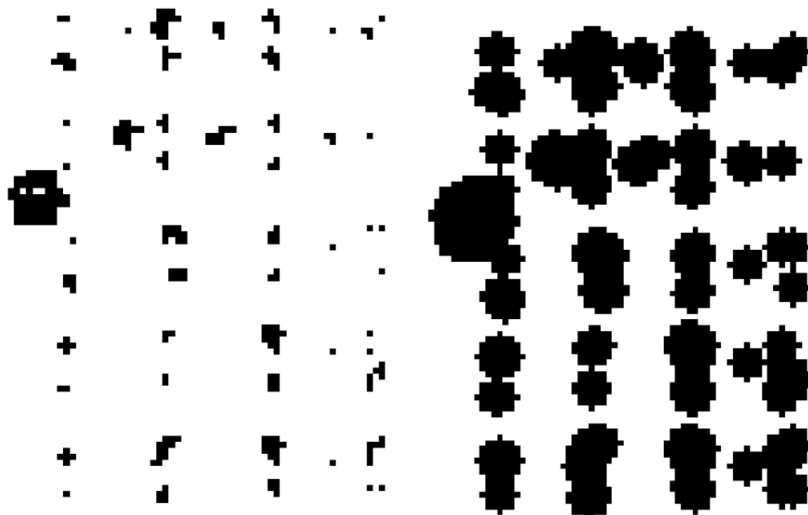
Zostrojuvanie ciest pre pohyb mobilného robota bolo jedným z hlavných cieľov našej bakalárskej práce. Samotné zostrojenie ciest pozostáva z niekoľkých krokov. Návrh riešenia jednotlivých krokov popisujeme v nasledujúcich podkapitolách.

#### ■ 4.2.1 Využitie binárnej dilatácie

Za začiatok procesu zostrojovania ciest sme považovali hľadanie spôsobov, ktoré nám umožnia nájsť miesta v mriežke obsadenosti, vhodné pre pohyb mobilného robota. Dôležité bolo myslieť na ohraničenie priestoru a prekážky, ktoré by pohyb robota zaiste obmedzovali. Našou prioritou preto bolo zabezpečiť bezkolízne podmienky.

Nami využívaná mriežka obsadenosti je reprezentovaná binárnym obrazom. Na základe jej reprezentácie sme plánovali samotného robota reprezentovať ako súbor pixelov. Zameranie na stred robota nám zjednoduší prácu s robotom v priestore, predstavenom mriežkou obsadenosti. Dôležitou súčasťou práce pri zostrojení ciest pre robota je poznanie jeho presných parametrov (dĺžka, šírka, rádius). Pomyselným pripočítaním určitého rozmeru k prekážkam a stenám, medzi ktorými by sa robot pohyboval, docielime zamedzenie potenciálnym nechceným kolíziám. Na virtuálne zväčšenie všetkých prekážok a stien, ktoré

pre robota predstavujú riziko kolízie, sme zvolili binárnu dilatáciu. Očakávaný výsledok po aplikácii binárnej dilatácie je uvedený na Obrázku 4.3:



**Obrázok 4.3:** Obrázok naľavo sú prekážky v mriežke obsadenosti pred aplikáciou binárnej dilatácie. Obrázok napravo sú prekážky v mriežke obsadenosti po aplikácii binárnej dilatácie.

Vo vzťahu k rozmerom robota sme určili, že šírka robota bude kľúčovým parametrom pre prácu s binárnou dilatáciou. Zameraním na rádus robota, to znamená na najväčšiu vzdialenosť od stredu robota, by s veľkou pravdepodobnosťou došlo k zamedzeniu miest, do ktorých by robot dokázal bezkolízne vojsť a vycúvať, a to len z dôvodu nedostatočného priestoru pre jeho rotáciu.

Pohyb robota je vždy uskutočňovaný medzi prekážkami (prekážkou rozumieme stenu a statické objekty v priestore). Pridaním polovice šírky robota ku všetkým stranám prekážok, by sa mohlo dostatočne znížiť riziko kolízie, a tým umožniť bezpečný pohyb mobilného robota. Preto sme práve polovicu šírky robota v pixeloch zvolili ako vstup do binárnej dilatácie. Polovicu šírky robota v pixeloch získame nasledujúcim spôsobom: podiel šírky robota v metroch a rozlíšenie mriežky obsadenosti v metroch na pixel, vydelíme dvoma. Výsledok po výpočte je potrebné zaokrúhliť smerom nahor.

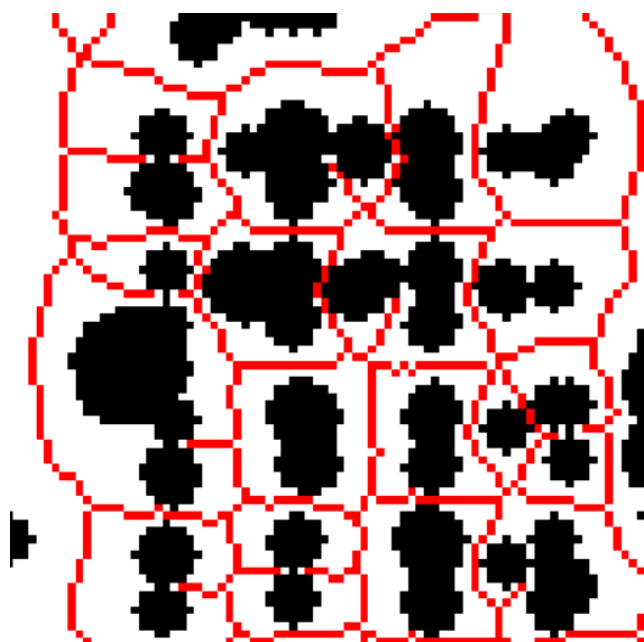
Výsledkom metódy binárnej dilatácie bude oddilatovaná mriežka obsadenosti. Vo výsledku získanej oddilatovanej mriežky obsadenosti zostanú voľné pixely predstavujúce miesta, po ktorých by sa mobilný robot dokázal bezkolízne pohybovať.

### 4.2.2 Využitie skeletonizácie

Po detekovaní miest, po ktorých sa mobilný robot dokáže bezkolízne pohybovať, bude ďalším krokom vytvorenie ciest. Z voľných pixelov získaných binárnou dilatáciou, vznikne množina všetkých miest, dostupných pre pohyb robota. Zostrojenie ciest vyžaduje úpravu vzniknutej množiny pixelov do podoby jednoznačnej kostry so šírkou jedného pixelu.

Pre tvorbu kostry využijeme proces skeletonizácie, ktorej cieľom bude reprezentovať voľný priestor v dilatovanej mriežke obsadenosti. Došlo by tak k zmenšeniu priestoru voľných miest na šírku jedného pixelu.

Výsledná kostra bude predstavovať konkrétne cesty pripravené na pohyb mobilného robota. Získanú kostru môžeme reprezentovať ako neorientovaný graf pozostávajúci z jednotlivých hrán a vrcholov. V tomto bode by sme hranami rozumeli cesty a vrcholmi križovatky ich spájajúce. Očakávaný výsledok skeletonizácie uvádzame na Obrázku 4.4:



Obrázok 4.4: Očakávaný výsledok skeletonizácie.

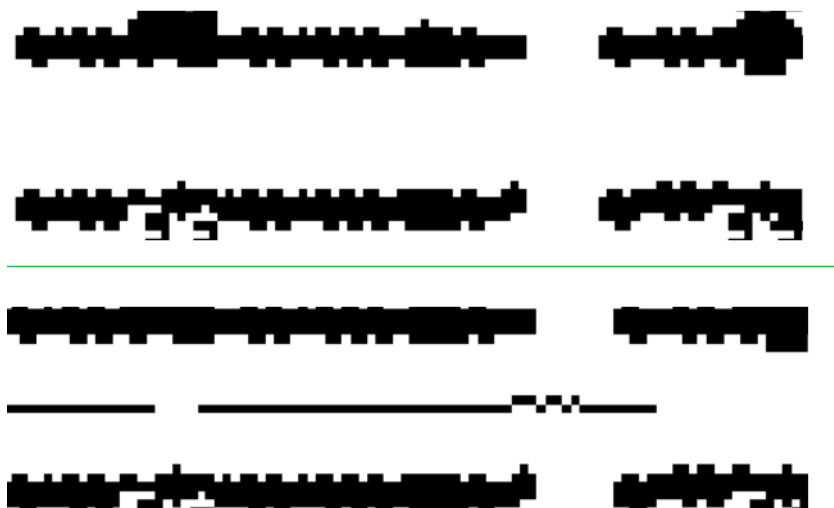


### 4.2.3 Vytvorenie stredových čiar ciest

Ďalším z cieľov našej bakalárskej práce bolo, v miestach to umožňujúcich, vytvoriť dvojprúde cesty. Logickým prvým krokom bude detekovať miesta v mriežke obsadenosti, ktorých rozmery budú dostatočné pre vytvorenie dvojprúdových ciest.

Opätovným využitím binárnej dilatácie bude možné detekciu zrealizovať. Rozdielom, v porovnaní s prvotnou dilatáciou (sekcia 4.2.1), je potreba zvažovať určité bezpečnostné rezervy v rozmeroch dilatovaných virtuálnych prekážok. Dôvodom je potenciálny paralelný pohyb viacerých robotov dvojprúdovými cestami, a teda pridaním dostatočnej bezpečnostnej rezervy by sa zabezpečil ich bezpečný a bezkolízny prechod. Dosiahneme toho dilatovaním prekážok nielen o polovicu šírky robota, ako v predošlom prípade, ale aj o dostatočne veľkú bezpečnostnú rezervu vzhľadom na aktuálne rozlíšenie mriežky obsadenosti. Vo výsledku dôjde k celkovým dvom dilatáciám: v prvom prípade dilatácia o polovicu šírky robota a v druhom prípade dilatácia o polovicu šírky robota s pridaním vhodne zvoleného rozmeru bezpečnostnej rezervy.

Po obidvoch potrebných dilatáciách by sme zahájili proces skeletonizácie, ktorej výsledkom bude kostra predstavujúca stredovú čiaru. Túto stredovú čiaru by sme vo forme virtuálnej prekážky pridali do mriežky obsadenosti, čím by došlo k rozdeleniu priestoru pre pohyb robotov na dve časti. Očakávaný výsledok vytvorením stredových čiar na Obrázku 4.5:



**Obrázok 4.5:** Hore mriežka obsadenosti bez stredových čiar ciest. Dole mriežka obsadenosti po vytvorení stredových čiar ciest.

#### ■ 4.2.4 Vytvorenie finálnych ciest

Po vytvorení virtuálnych prekážok v mriežke obsadenosti bude možné pracovať na tvorbe finálnych ciest. Za finálnu cestu považujeme výsledok nižšie popísaného procesu binárnej dilatácie a skeletonizácie, čo by predstavovalo výslednú roadmapu, po ktorej by sa mobilný robot dokázal pohybovať.

V tvorbe finálnych ciest budeme využívať mriežku obsadenosti, s už vytvorenými stredovými čiarami. Binárnou dilatáciou prekážok v rozmere polovice šírky robota, budeme určovať miesta umožňujúce bezkolízny pohyb. Následne bude potrebné reprezentovať určené miesta procesom skeletonizácie. Cieľom skeletonizácie je, rovnako ako v popise skeletonizácie v sekcii 4.2.2, zhotoviť kostru všetkých miest umožňujúcich bezkolízny pohyb robota. Výsledná kostra bude predstavovať finálnu roadmapu, na ktorej bude možné globálne plánovať pohyb mobilného robota. Finálna roadmapa by v tomto kroku bola v podobe binárneho obrazu.

### ■ 4.3 Dátová reprezentácia roadmapy

Štúdiom literatúry [18], [19], [20] sme zistili existenciu rôznych dátových reprezentácií pre plánovacie algoritmy. Pre účely našej bakalárskej práce sme si zvolili dátovú reprezentáciu v podobe orientovaného grafu. Základom vytvorenia orientovaného grafu bude práca s neorientovaným grafom, ktorý našu roadmapu dokáže reprezentovať vrcholmi a hranami. Dátová reprezentácia neorientovaného grafu bude následne pozostávať z dvoch grafov. Prvý graf bude obsahovať zostavené hrany (postupnosť pixelov), na základe ktorých sa bude mobilný robot vedieť pohybovať. Druhý graf bude obsahovať vrcholy a k nim prislúchajúce susedné vrcholy, bude základom pre plánovanie pohybu mobilného robota.

Začiatkom tvorby neorientovaného grafu bude detekcia miest reprezentujúcich vrcholy v roadmape, predstavenej formou binárneho obrazu. Inšpirovali sme sa dopravnými cestami, v ktorých vrcholy reprezentujú križovatky a koncové body ciest. Koncovými bodmi myslíme miesta, v ktorých dochádza k ukončeniu cesty a jej ďalšiemu nerozvetvovaniu.

Na detekciu vrcholov neorientovaného grafu bude potrebné využiť algoritmus, umožňujúci vyhľadávanie konkrétnych tvarov v binárnom obraze. Z dostupných algoritmov považujeme za vhodnú transformáciu Hit and Miss, patriacu do matematickej morfológie. V úvode využitia uvedenej transformácie bude potrebné si určiť možné tvary križovatiek a koncových bodov ciest v binárnom obraze. Využitie transformácie Hit and Miss vedie k detekcii hľadaných vrcholov, od ktorých sa odvíjajú ďalšie kroky v tvorbe neorientovaného grafu.

Ďalším postupom po detekcii vrcholov, bude zostrojovanie hrán. Za algoritmus vhodný pre našu prácu považujeme prehľadávanie do šírky (Breadth-first search) [21]. Predpokladáme, že uvedený algoritmus nám poskytne základ pre konštrukciu hrán a nájdenie im prislúchajúcich vrcholov.

Za finálny krok považujeme zostrojenie orientovaného grafu. V našej práci sa vo vzťahu k tomuto finálnemu kroku venujeme iba príprave, ktorá vytvorí podklad pre potenciálnu ďalšiu prácu v budúcnosti. V rámci prípravy sa zameriame na priradenie smeru hranám, ktoré budú obojsmerné a na detekciu hrán, ktorých smer si bude môcť užívateľ zvoliť sám. Hrany, ktoré budú obojsmerné, bude možné detekovať prostredníctvom kontroly súvislých komponent. Hrany, ktorých smer si užívateľ bude môcť zvoliť sám, budeme určovať porovnaním prvotnej a finálnej kostry. Hrany, ktoré po porovnaní kostier budú zhodné, budú tie hrany, ktorých smer si užívateľ bude môcť zvoliť sám. Hrany, ktoré nepatria do vyššie uvedených dvoch možností, budú súčasťou ďalšieho kroku v tvorbe orientovaného grafu, ktorý bližšie odporúčame pre potenciálnu ďalšiu prácu v kapitole 7 (Záver).



## Kapitola 5

### Implementácia riešenia

Potrebnú úpravu vstupných dát sme previedli v grafickom nástroji pre úpravu a tvorbu rastrovej grafiky s názvom GIMP. Návrh riešenia sme implementovali v programovacom jazyku Python 3, ktorý patrí do kategórie interpretovaných programovacích jazykov. Vytvorili sme hlavné dva programy. Prvý program slúži na vytvorenie finálnej roadmapy. Druhý z programov vytvorí grafovú dátovú reprezentáciu v podobe neorientovaného grafu a prípravu pre zostrojenie orientovaného grafu.

Prvý program využíval ďalej popísané funkcie. Pre potreby načítania a zápisu binárnych obrazov využívame funkciu z knižnice OpenCv s názvom `cv2.imread()` a `cv2.imwrite()` [22]. Na aplikáciu binárnej dilatácie využívame funkciu z knižnice OpenCV s názvom `cv2.dilate()` [23]. Štrukturovací element, ktorý využívame vo funkcii binárnej dilatácie, sme vytvorili v programe Matlab funkciou `strel()` [24]. Pre aplikáciu skeletonizácie budeme využívať funkciu s názvom `skeletonize()` [25], z knižnice `scikit-image`, konkrétne z modulu `morphology`. Knižnica `scikit-image` obsahuje algoritmy pre spracovanie obrazov. Vykresľovanie výsledkov jednotlivých operácií prevádzame pomocou knižnice `matplotlib` [26]. Na zápis a načítanie súborov s príponou `csv` využívame knižnicu s názvom `csv` [27].

Druhý program využíval na detekciu vrcholov pri konštrukcii neorientovaného grafu funkciu `binary_hit_or_miss` z knižnice `SciPy` z modulu `ndimage` [28]. Graf, ktorý obsahuje jednotlivé vrcholy a im prislúchajúce susedné vrcholy, je reprezentovaný ako slovník. Graf s jednotlivými hranami je reprezentovaný ako pole. Pri detekcii spojitých komponentov využívame funkciu `label()` z knižnice `SciPy` z modulu `ndimage.measurements` [29].



## Kapitola 6

### Experimenty

V tejto kapitole sa budeme zaoberať overením nášho riešenia, ktoré prinesieme formou detailného opisu a príslušného obrázkového materiálu.

#### 6.1 Vytvorenie ciest pre mobilného robota

V tejto kapitole podrobne popíšeme aplikáciu nášho návrhu pre automatickú konštrukciu roadmapy. Naším cieľom je pomocou binárnej dilatácie a skeletonizácie skonštruovať cesty pre mobilného robota. Následne vytvoríme neorientovaný graf, ktorý bude dátovou reprezentáciou skonštruovaných ciest a pripravíme základy pre konštrukciu orientovaného grafu.

Riešenie sme testovali na reálnych vstupných dátach a pracovali s robotom Jackal UGV od spoločnosti Clearpath.

Vstupné dáta, na ktorých sme vytvárali roadmapu, nám boli poskytnuté z práce uvedenej v sekcii 4.1.4.

### ■ 6.1.1 Úprava vstupných dát

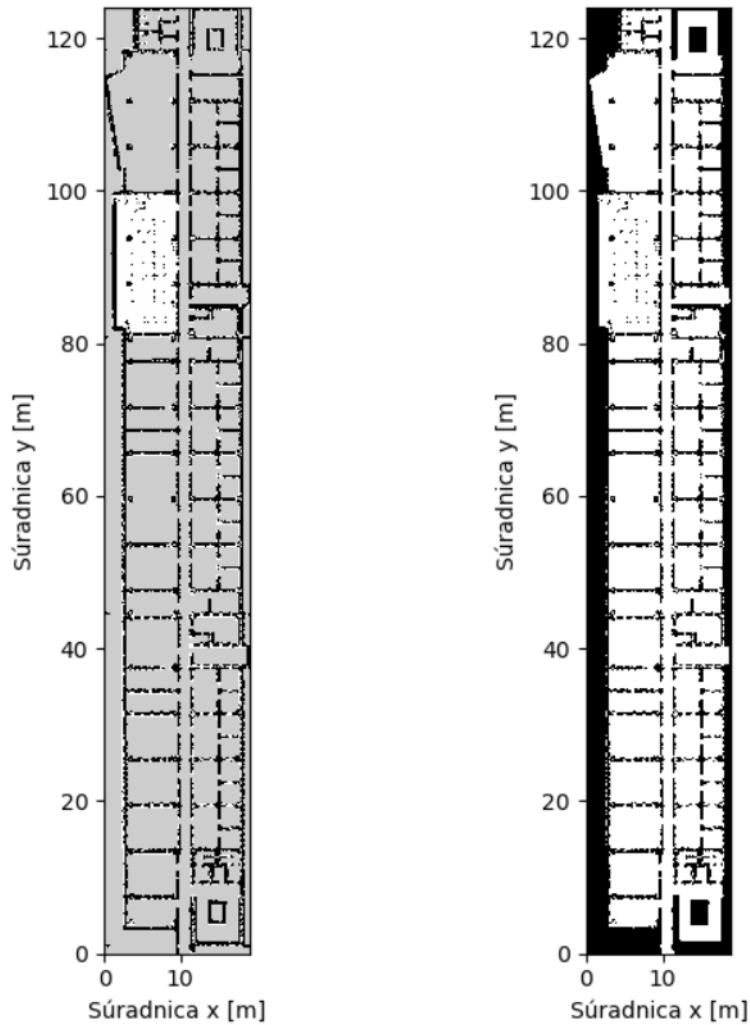
V obstaraných vstupných dátach boli miesta, ktorými by vzhľadom k parametrom nášho robota, nebolo možné prejsť. Takými miestami sú napríklad schodisko alebo vonkajšia časť budovy. Uvedené miesta sú v mriežke obsadenosti zakreslené ako voľné, a preto ich bolo potrebné upraviť pred následným využitím v našom algoritme.

Úpravu vstupných dát sme previedli v grafickom nástroji pre úpravu a tvorbu rastrovej grafiky s názvom GIMP. Miesta, ktorými mobilný robot neprejde (napríklad schodisko, výťah, vonkajšia časť budovy), sme v mriežke obsadenosti nastavili na hodnotu 0, čo v našom ponímaní znamená, že sú to miesta obsahujúce prekážky.

Ďalšou úpravou, ktorú si práca so vstupnými dátami vyžadovala, bolo zjednotenie hodnôt pixelov na 0 (obsadené políčko - prekážka) a 1 (voľné políčko). Zjednotenie hodnôt sme dosiahli porovnaním hodnoty každého pixelu. Ak hodnota pixelu bola väčšia ako 0, pixelu sme priradili hodnotu 1. Dôvodom zjednotenia hodnôt pixelov bolo používanie binárnej dilatácie. Zjednotenie hodnôt nám zaručuje jasné vyčlenenie prekážok, respektíve voľných plôch. Úprava vstupných dát je znázornená na Obrázku 6.1.



Mriežka obsadenosti pred úpravami Mriežka obsadenosti po úpravách



**Obrázok 6.1:** Naľavo je pôvodná mriežka obsadenosti. Napravo je mriežka obsadenosti po úpravách.

Vzhľadom k rozsiahlosti pôdorysu podlažia, budeme pre lepšiu ilustráciu využívať určité jeho výseky. Výseky pôdorysu podlažia volíme na základe prítomnosti naskenovaných prekážok, ktoré vyrobil mobilný robot svojim prechodom miestnosťou.

## 6.1.2 Základy zostrojenia ciest pre mobilného robota

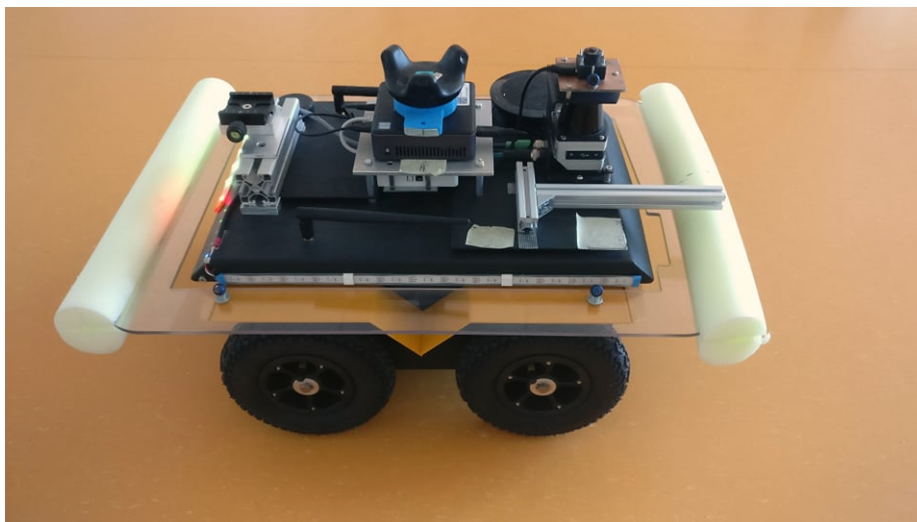
### Aplikácia binárnej dilatácie

Na konštrukciu ciest pre mobilného robota budeme ako vstup požadovať upravené vstupné dáta.

Prvým krokom je využitie binárnej dilatácie, ktorá nám poskytne informáciu o miestach, kde stred robota dokáže prejsť bez kolízie s okolitými prekážkami. Pre vstup do binárnej dilatácie sú potrebné dáta vo formáte hodnôt pixelov reprezentujúcich prekážku = 1 a voľné miesto = 0. V našom prípade to znamená, že musíme upravené vstupné dáta invertovať.

Štrukturovací element v binárnej dilatácii sme zvolili na základe parametrov nášho robota, ktoré sú 710mm (dĺžka) x 461mm (šírka). Zaujímá nás len dĺžka a šírka robota, pretože pracujeme v dvojdimenzionálnom priestore.

Robot Jackal, s ktorým pracujeme, je na Obrázku 6.2:



Obrázok 6.2: Robot Jackal.

Rozlíšenie našej mriežky obsadenosti je 0.084 m/pixel, preto sme museli vypočítať koľko pixelov bude zaberat náš robot na šírku. Potrebujeme, aby sa strany prekážok rozšírili o polovicu šírky robota, a preto získaný výsledok vydáme dvoma. Výpočet prevedieme pomocou vzorca popísaného v sekcii 4.2.1. Po aplikácii vzorca na konkrétne čísla, s ktorými sme pracovali, je výsledný počet zaberajúcich pixelov nasledovný:

$$\frac{(0.461/0.0840)}{2} = 2.75 \doteq 3[px] \quad (6.1)$$

Štrukturovací element, ktorý reprezentuje polovicu šírky robota 3 pixely, sme vytvorili v programe Matlab. Štrukturovací element vyrobený v Matlabe bola matica o rozmere 7x7, ktorá mala na každom mieste hodnotu 1. Robota sme reprezentovali v tvare kruhu a bolo potrebné štrukturovací element upraviť a zachovať polovicu jeho šírky. Štrukturovací element mal po úprave tvar:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6.2)$$

Po aplikovaní binárnej dilatácie na upravené vstupné dáta, sme získali novú mriežku obsadenosti. Miesta, kde stred mobilného robota prejde bez kolízie s okolím, sú vyznačené ako voľné (pixely s hodnotou 0). Výsledok binárnej dilatácie je znázornený na Obrázku 6.3.



**Obrázok 6.3:** Naľavo je zobrazená mriežka obsadenosti pred aplikáciu binárnej dilatácie. Napravo je mriežka obsadenosti po aplikácii binárnej dilatácie (výsek medzi 70 m-110 m).

### ■ Aplikácia skeletonizácie

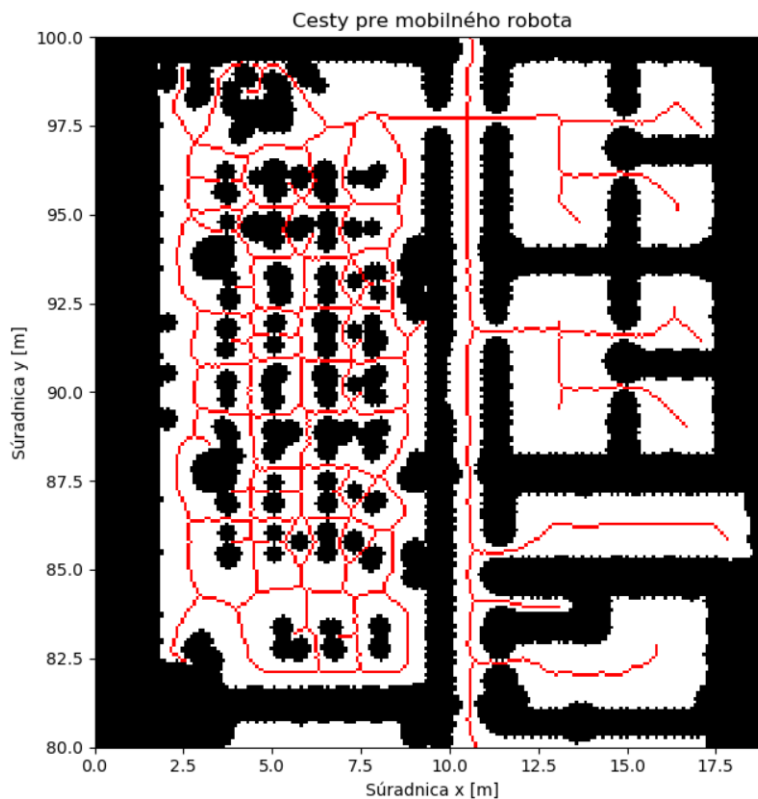
Ďalším krokom pri vytváraní ciest pre robota je aplikácia skeletonizácie na dilatovanú mriežku obsadenosti. Skeletonizáciou získame pixely, ktoré budú reprezentovať cesty, po ktorých sa robot bude môcť pohybovať.

Algoritmus skeletonizácie na vstupe vyžaduje, aby sme invertovali dilatovanú mriežku obsadenosti. Po invertovaní aplikujeme algoritmus na vytvorenie kostry. Výstupom algoritmu je binárny obraz. V binárnom obraze máme označené pixely, ktoré patria do kostry s hodnotou 1 a ostatné s hodnotou 0. Výstupom skeletonizácie môže vzniknúť viacero kostier. Dôvodom vzniku viacerých kostier sú neprepojené miesta v mriežke obsadenosti. Príkladom je chodba a miestnosti, do ktorých sú v mriežke obsadenosti uzatvorené dvere.

Kostry sa algoritmom v chodbe a v uzavretých miestnostiach vytvoria, ale nebudú navzájom prepojené. Skonstruované kostry mali na niektorých miestach málo spojených pixelov, teda cesty boli príliš krátke na to, aby mal pohyb mobilného robota po nich význam. Na základe nami zvolenej podmienky, aby dĺžka cesty mala aspoň 1 meter, zodpovedajúcej hodnote 13 pixelov, sme odstránili kostry nespĺňajúce túto podmienku.

Na odstránenie kostier, ktoré nezodpovedali nami stanovenej podmienke, sme využili algoritmus pozostávajúci z dvoch úkonov. Prvým úkonom bolo nájsť spojené komponenty v binárnom obraze a zistiť, aký počet pixelov obsahujú. Druhým úkonom bolo vymazať kostry nevyhovujúce našej podmienke pre ich dĺžku.

Výstupom algoritmu boli len kostry, ktoré zodpovedali nami zadanej podmienke týkajúcej sa dĺžky ciest. Výstup môžeme vidieť na Obrázku 6.4:



**Obrázok 6.4:** Výsledné kostry spĺňajúce podmienku na dĺžku, zobrazené červenou farbou (výsek medzi 80 m-100 m).

### 6.1.3 Vytvorenie stredových čiar ciest

Predošlá sekcia obsahovala popis hlavných algoritmov a princípov riešenia pre vytváranie ciest vhodných pre pohyb mobilného robota. V našej práci sme sa zaoberali aj možnosťou paralelného pohybu viacerých mobilných robotov. Nasledujúca sekcia pozostáva z popisu vytvárania stredových čiar ciest.

#### Aplikácia dvojitej binárnej dilatácie

Mriežka obsadenosti musí spĺňať podmienky pre aplikáciu dvojitej binárnej dilatácie, detailne popísané v sekcii 6.1.2 (Aplikácia binárnej dilatácie). Cieľom dvojitej binárnej dilatácie je detekovať miesta v mriežke obsadenosti, ktorými môžu prechádzať dvaja mobilní roboti paralelne, bez kolízie medzi sebou a okolitým prostredím.

Pre zaistenie dostatočného priestoru medzi mobilnými robotmi a okolitým prostredím použijeme dva rozličné štrukturovacie elementy. Prvý štrukturovací element, ktorý aplikujeme na mriežku obsadenosti, bude mať rovnaký tvar, ako tvar popísaný v sekcii 6.1.2 (Aplikácia binárnej dilatácie). Druhý štrukturovací element bude väčší o vhodne zvolenú bezpečnostnú rezervu vzhľadom na zabezpečenie dostatočného priestoru pre prechod dvoch mobilných robotov v rovnakom mieste. Štrukturovací element, ktorý aplikujeme, ako druhý v poradí, bude reprezentovaný maticou v tvare:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6.3)$$

Zadaný štrukturovací element má v šírke 4 pixely, čo je v prepočte 0.34 metra. Pre zamedzenie kolízie mobilných robotov, sme experimentálne zvolili bezpečnostnú rezervu 1 pixel. Výsledok dvojitej binárnej dilatácie je na Obrázku 6.5:



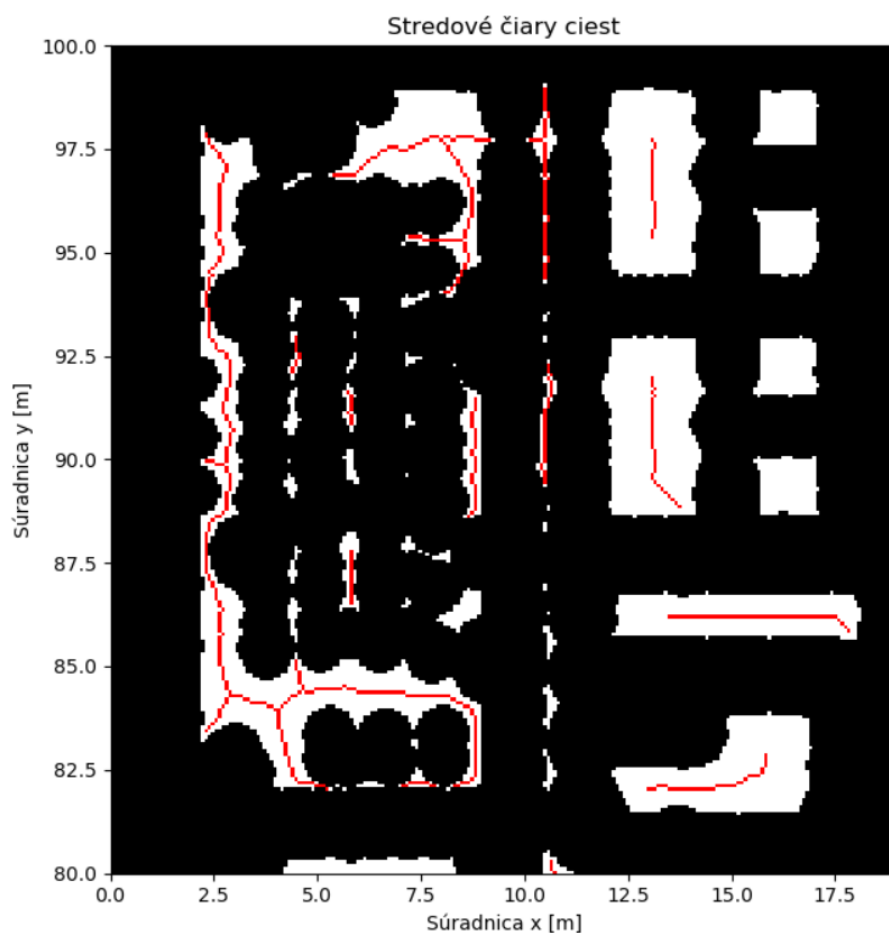
**Obrázok 6.5:** Naľavo je zobrazená mriežka obsadenosti pred aplikáciou dvojitej binárnej dilatácie. Napravo je mriežka obsadenosti po aplikácii dvojitej binárnej dilatácie (výšek medzi 70 m - 110 m).

### ■ Aplikácia skeletonizácie a vytvorenie stredových čiar ciest

Ďalším krokom po detekcii miest, kde sa roboti môžu paralelne bezkolízne pohybovať, je aplikácia skeletonizácie. Vstupom tohto algoritmu je invertovaná dvojito dilatovaná mriežka obsadenosti. Touto skeletonizáciou získame pomyselné stredové čiary, ktoré nám v ďalšom postupe riešenia vytvoria cesty pre viacerých mobilných robotov. Výstupom algoritmu skeletonizácie je binárny obraz, v ktorom sú pixely patriace do kostier označené hodnotou 1 a ostatné pixely hodnotou 0. Pre vytvorenie virtuálnych stredových čiar využijeme tie pixely, ktoré do kostier patria.

Pred vytvorením virtuálnych stredových čiar sme odstránili kostry, ktoré nespádali do nami zadaného kritéria týkajúceho sa ich dĺžky. Zadané kritérium

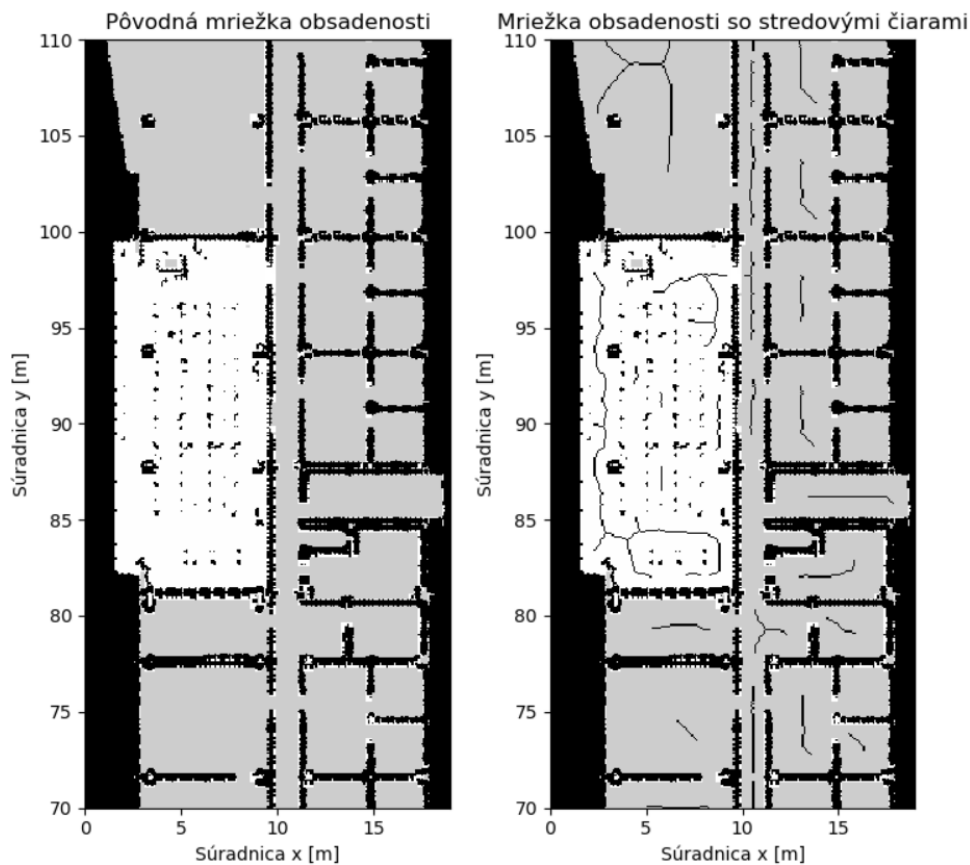
bolo, aby kostra obsahovala aspoň 10 pixelov, čo predstavuje jeden a pol dĺžky robota (0.84 m). Dôvodom zavedenia tohto kritéria je zabezpečenie paralelného pohybu dvoch mobilných robotov bez kolízie. Kostry, ktoré reprezentujú stredové čiary ciest, sú znázornené na Obrázku 6.6.



**Obrázok 6.6:** Kostry reprezentujúce stredové čiary ciest, zobrazené červenou farbou (výsek medzi 70 m-100 m).

Virtuálne stredové čiary vytvoríme podľa nami navrhnutého algoritmu. Tento algoritmus funguje na princípe prehľadávania binárneho obrazu, ktorý je výstupom skeletonizácie. V binárnom obraze vyhľadávame indexy pixelov s hodnotou 1 a do upravenej mriežky obsadenosti ich zapisujeme ako prekážku (pixel s hodnotou 0). Tieto prekážky nám budú reprezentovať virtuálne stredové čiary, rozdeľujúce miesta, kde sa bezkolízne paralelne pohybujú dvaja mobilní roboti, na dvojprúdovú cestu. Na nižšie uvedenom Obrázku 6.7 môžeme vidieť vytvorené stredové čiary v mriežke obsadenosti.





**Obrázok 6.7:** Porovnanie pôvodnej mriežky obsadenosti naľavo s mriežkou obsadenosti s vytvorenými stredovými čiarami napravo (výsek medzi 70 m-110 m).

#### 6.1.4 Vytvorenie finálnych ciest pre mobilného robota

Vytvorením stredových čiar ciest, ktorých konštrukciu sme popísali v predošlej sekcii, sa môžeme posunúť v riešení nášho problému a vytvoriť finálnu roadmapu pre mobilného robota.

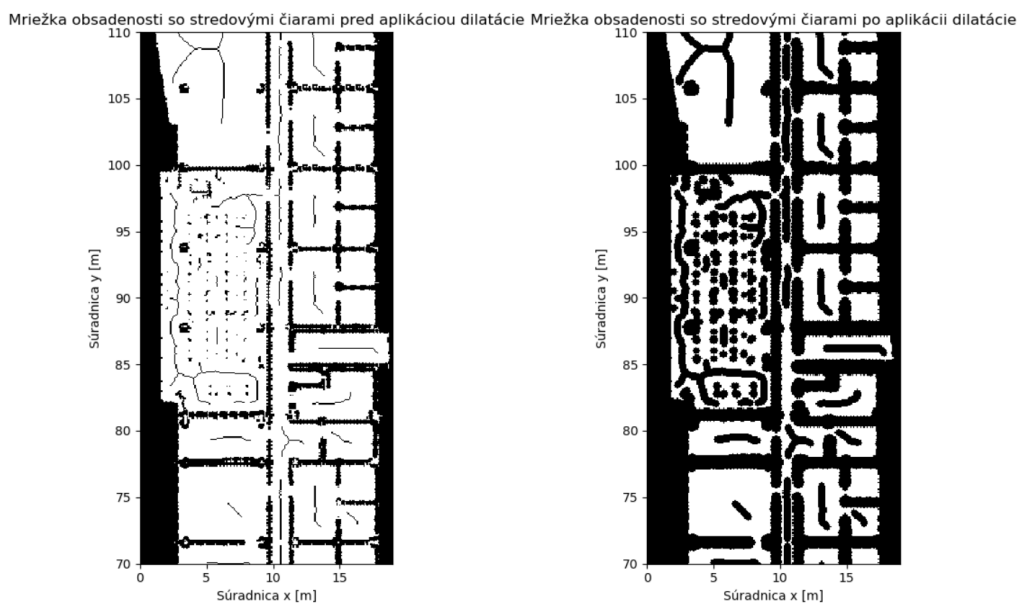
##### Aplikácia binárnej dilatácie

Vstupom do algoritmu je upravená mriežka obsadenosti, v ktorej sú vytvorené virtuálne prekážky. Základné spracovanie vstupných dát pre algoritmus je popísané detailne v sekcii 6.1.2 (Aplikácia binárnej dilatácie). Aplikujeme algoritmus binárnej dilatácie so štrukturovacím elementom, ktorého reprezentácia

je matica v tvare:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6.4)$$

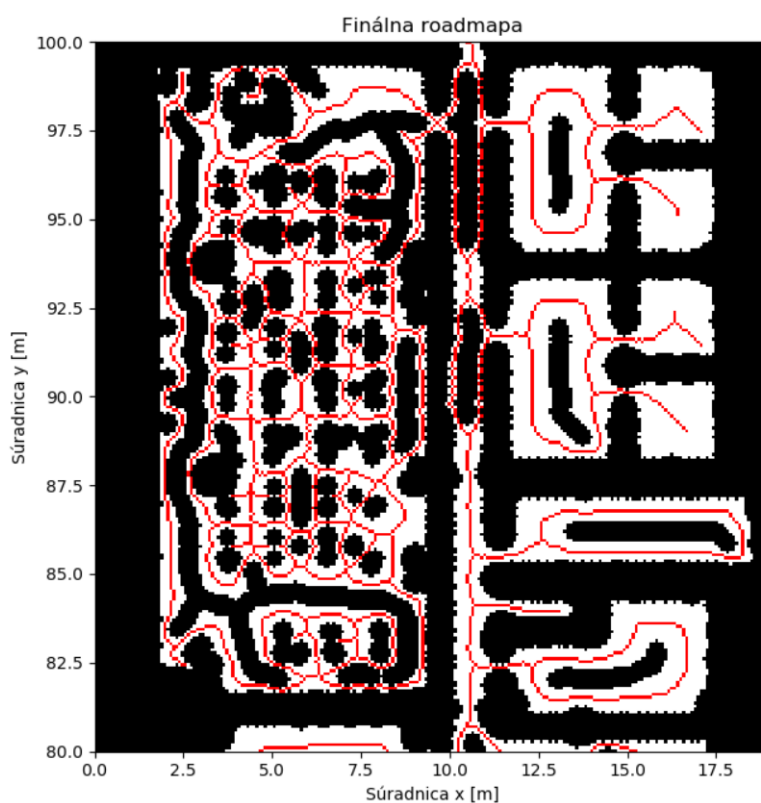
Výstupom algoritmu je binárny obraz, na ktorom detekujeme miesta, kde sa mobilný robot dokáže pohybovať bezkolízne. V algoritme sme použili štrukturovací element, ktorý reprezentuje polovicu šírky jedného mobilného robota. Výstup binárnej dilatácie aplikovanej na mriežku obsadenosti so stredovými čiarami, môžeme vidieť na Obrázku 6.8:



**Obrázok 6.8:** Naľavo je zobrazená mriežka obsadenosti obsahujúca stredové čiary pred aplikáciou binárnej dilatácie. Napravo je mriežka obsadenosti so stredovými čiarami po aplikácii binárnej dilatácie (výsek medzi 70 m-110 m).

### ■ Aplikácia skeletonizácie

Algoritmus na vstupe vyžaduje invertovaný výstup binárnej dilatácie. Využitím algoritmu skeletonizácie sme vytvorili kostry, ktoré reprezentujú finálne cesty pre mobilného robota. Finálne cesty odfiltrujeme pomocou postupu popísaného v sekcii 6.1.2 (Aplikácia skeletonizácie). Finálnu roadmapu môžeme vidieť na Obrázku 6.9.



**Obrázok 6.9:** Kostry reprezentujúce finálnu roadmapu, znázornené červenou farbou (výsek medzi 80 m-100 m).

Finálnou úpravou výstupu algoritmu skeletonizácie je zjednotenie pixelov. Pixely zjednotíme prechádzaním výstupu algoritmu a porovnávaním hodnôt každého pixelu. Ak hodnota pixelu je iná ako 0, danú hodnotu prepíšeme na hodnotu 1. Výsledok bude binárny obraz, v ktorom budú cesty reprezentované pixelmi s hodnotou 1.

Pre účely dátovej reprezentácie finálnej roadmapy budeme potrebovať zoznam pixelov, ktoré sa po porovnaní prvotnej a finálne kostry zhodujú. Zoznam je následne potrebné uložiť do súboru.

## 6.2 Vytvorenie dátovej reprezentácie ciest

V predošlej kapitole sme si detailne popísali algoritmus na konštrukciu ciest pre mobilných robotov. Skonštruované cesty sú v podobe pixelov s hodnotami 1, po ktorých sa môže mobilný robot pohybovať.

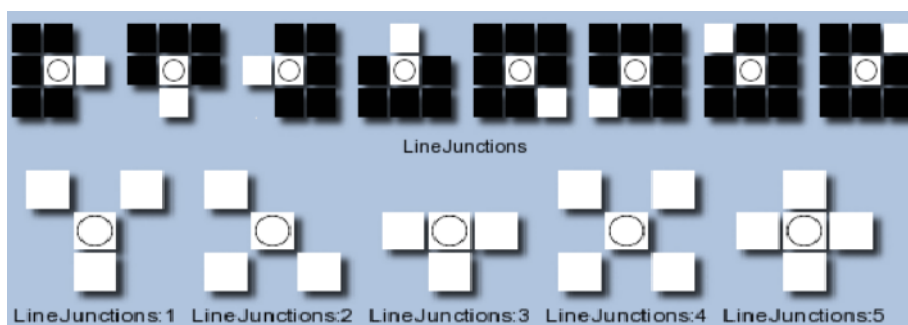
Výsledné cesty je potrebné uchovávať v dátovej štruktúre, pre následné využitie v plánovacích algoritmoch.

Pre naše riešenie sme si vybrali dátovú reprezentáciu v podobe orientovaného grafu.

### 6.2.1 Detekcia vrcholov

Prvým krokom bolo určiť si miesta, ktoré budú v binárnom obraze ciest reprezentovať vrcholy. K detekcii takýchto miest sme využili algoritmus, ktorý patrí do matematickej morfológie s názvom Hit and Miss.

Na začiatku bolo potrebné určiť tvar križovatiek a koncových bodov ciest v binárnom obraze. Najčastejšie tvary križovatiek a koncových bodov sú zobrazené na Obrázku 6.10 [30],[31]:

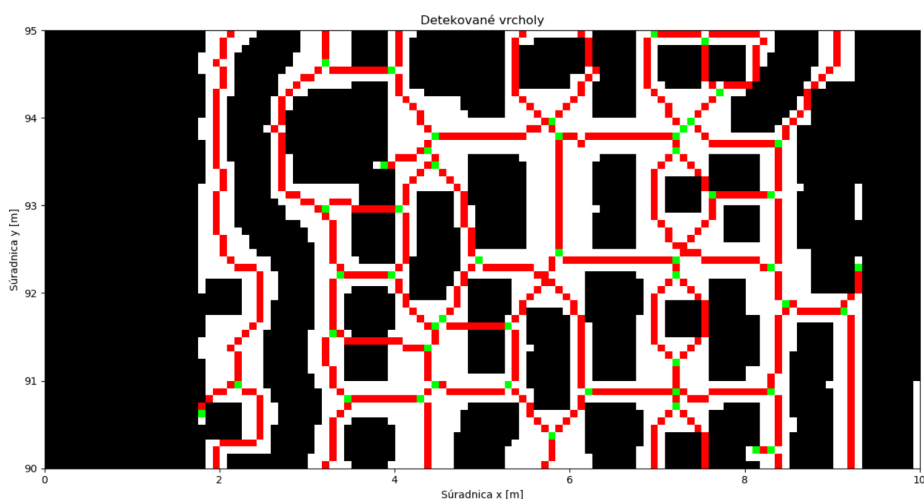


**Obrázok 6.10:** Horná časť obrázku znázorňuje koncové body ciest. Dolná časť obrázku znázorňuje križovatky.

### 6.2.2 Algoritmus na detekciu vrcholov

Jedným zo vstupov do algoritmu je binárny obraz, získaný ako výstup skeletonizácie pre konštrukciu finálnych ciest pre mobilného robota. Vstupom do algoritmu sú aj štrukturovacie elementy, reprezentujúce tvary križovatiek. Každý tvar má vlastné dva štrukturovacie elementy. Prvým z nich je štrukturovací element Hit, ktorý znázorňuje matica v tvare 3x3. Táto matica má na miestach vyznačených v obrázkoch bielou farbou hodnoty 1, a na ostatných miestach hodnoty 0. Druhý štrukturovací element sa nazýva Miss, ktorý pozostáva z invertovaného elementu Hit.

Výstupom algoritmu je binárny obraz obsahujúci pixely s hodnotou 1, ktoré reprezentujú detekované miesta križovatiek a koncových bodov ciest. Ostatné pixely v binárnom obraze majú hodnotu 0. Ďalším výstupom je súbor, ktorý obsahuje indexy pixelov reprezentujúcich vrcholy. Následne sme indexy pixelov s hodnotou 1 uložili do textového súboru pre následné využitie v konštrukcii grafu. Detekciu vrcholov aplikovanú na cesty je možné vidieť na Obrázku 6.11:



**Obrázok 6.11:** Detekované vrcholy na cestách, označené zelenou farbou (výšek medzi 90 m-95 m na výšku a 0 m-10 m na šírku).

### 6.2.3 Algoritmus na koštrukciu podgrafu

V tejto sekcii sa budeme detailne venovať popisu algoritmu pre koštrukciu podgrafu. Počet skonštruovaných podgrafov zodpovedá počtu kostier vo výstupe skeletonizácie pri vytváraní finálnych ciest.

#### Priradenie vrcholov do podgrafov

Výstupom algoritmu sú dva zoznamy. Prvým je zoznam obsahujúci podgrafy a im prislúchajúce indexy pixelov. Druhý zoznam obsahuje podgrafy a im priradené vrcholy. Vstupom do algoritmu bol binárny obraz s finálnymi cestami.

Prvým krokom bolo zistiť počet kostier v binárnom obraze. Detekciu počtu kostier sme previedli algoritmom, ktorý mal na výstupe dva parametre.

Prvým parametrom bol počet súvislých komponent, ktorý zodpovedal počtu kostier. Súvislá komponenta v binárnom obraze je množina pixelov, ktoré sú vzájomne spojené. Druhým parametrom bolo označenie pixelov, ktoré patrili do jednotlivých súvislých komponent. Vytvorili sme pomocnú dátovú reprezentáciu, ktorá vyjadrovala počet podgrafov a následne sme jednotlivým podgrafom priradili im prislúchajúce pixely.

V ďalšom kroku sme načítali indexy z vytvoreného súboru, ktorý sme popísali v sekcii 6.2.2. Priradenie indexov vrcholov podgrafom sme vykonali prostredníctvom nami navrhnutého algoritmu. Algoritmus pozostával z vytvorenia dátovej reprezentácie, do ktorej sme priradili vrcholy prislúchajúce podgrafom. Priradenie indexov vrcholov sme previedli pomocou porovnávania indexov pixelov v jednotlivých podgrafoch s indexmi vrcholov.

Pozorovaním výstupu algoritmu vyvstali situácie, kedy ani jeden vrchol nebol priradený k podgrafu. Znamenalo to, že algoritmus na detekciu vrcholov nedetekoval vrchol v kostre, ktorá reprezentovala cestu.

Prípady, v ktorých vrcholy neboli detekované, boli v tvare kružníc. Riešením problému bolo vybrať ľubovoľný pixel z daného podgrafu, v ktorom vrchol nebol detekovaný, a priradiť ho ako vrchol k súhlasnému podgrafu.

### ■ Selektívny výber podgrafu

K vytvoreniu algoritmu selektívneho výberu podgrafu nás motivovala predstava určenia podgrafu, v ktorom sa mobilný robot nachádza. Výhodou tohto prístupu je optimalizácia následného algoritmu na zostavenie susedných vrcholov a hrán podgrafu.

Vstupom algoritmu je index pozície mobilného robota a zoznam pixelov, ktoré patria jednotlivým podgrafom. Zoznam podgrafov sme získali v sekcii 6.2.3 (Priradenie vrcholov do podgrafov). V algoritme sme porovnávali index pozície mobilného robota s indexmi pixelov v jednotlivých podgrafoch. Po nájdení príslušného podgrafu so zhodujúcim sa indexom pixelu a indexu pozície mobilného robota, sme vybrali príslušný podgraf s jeho vrcholmi. Výstupom algoritmu sú vrcholy vybraného podgrafu.

## ■ Zostavenie hrán a susedných vrcholov podgrafu

V tejto sekcii si predstavíme algoritmus, ktorý sa zaoberá konštrukciou hrán a susedných vrcholov v podgrafe.

Jeho základom je prehľadávanie do šírky. Na vstupe algoritmu sú vrcholy podgrafu, ktoré sme získali z algoritmu v sekcii 6.2.3 (Selektívny výber podgrafu) a binárny obraz s finálnymi cestami. Výstupom algoritmu sú dva grafy. Prvý graf uchováva zostavené hrany. Prvý index v hrane je vrchol, z ktorého hrana vychádza a posledný index je vrchol, kde hrana končí. Druhý graf uchováva prehľadávané vrcholy a k nim prislúchajúce susedné vrcholy.

Algoritmus pozostáva z dvoch cyklov. Hlavný cyklus slúži na prechádzanie vrcholov podgrafu. Podcyklus je vnorený do hlavného cyklu a zaoberá sa zostavením hrán a susedných vrcholov.

### Hlavný cyklus

Hlavný cyklus má na vstupe indexy vrcholov podgrafu a prebieha pokiaľ sa neprehľadajú všetky indexy vrcholov. Na začiatku vytvoríme zoznamy s názvom `visited` a `frontier`. Zoznam `visited` slúži na uchovávanie indexov pixelov, ktoré sme už prehľadali. Zoznam `frontier` slúži na uchovávanie indexov pixelov, ktoré musíme prehľadať a je implementovaný ako dátová štruktúra `Fronta`. Dôležité je vytvorenie premennej, ktorá slúži na uchovávanie počtu vrcholov, ktoré je potrebné nájsť.

Vytvoríme základ pre podcyklus, ktorý zahŕňa prvé prehľadanie vrcholu a vytvorenie začiatku hrán. Prehľadávanie spočíva v detekovaní každého z možných ôsmich susedných pixelov s hodnotou 1. Ak je susedná hodnota prehľadávaného pixelu 1 a index susedného pixelu patrí do zoznamu indexu vrcholov na vstupe, tak index susedného pixelu pridáme do zoznamu susedných vrcholov pre daný vrchol.

Pokiaľ je susedná hodnota prehľadávaného pixelu 1, ale index susedného pixelu sa nenachádza v zozname indexov vrcholov na vstupe, algoritmus vytvorí začiatok hrany a pridá do nej susedný index pixelu. Susedný index pixelu pridáme následne do zoznamu `frontier`. Ďalším krokom je pridanie prehľadávaného indexu pixelu do zoznamu `visited`. Potrebné je si uvedomiť, že počet začatých hrán sa rovná počtu vrcholov, ktoré je potrebné nájsť.

## Podcyklus

Podcyklus prebieha pokiaľ nie sú všetky potrebné vrcholy nájdené. Algoritmus vezme prvý index pixelu z frontiera a začne prehľadávať všetkých osem možných susediacich pixelov. Paralelne detekuje, ktorej hrane patrí prehľadávaný pixel. Ak je hodnota susedného pixelu 1 a susedný pixel nie je v zozname visited a súbežne nepatrí do iných hrán, algoritmus môže pokračovať dvomi spôsobmi.

Prvý spôsob nastáva v prípade, že sa index susedného pixelu nachádza v zozname indexov vrcholov. Index susedného pixelu následne pridáme do premennej, ktorá uchováva nájdený index vrcholu v danej ceste. Počet vrcholov, ktoré je potrebné nájsť sa nám zníži. Nutnosťou je vymazať všetky pridané indexy susedných pixelov, ktoré sme pridali do zoznamu frontier v priebehu prehľadávania okolia prehľadávaného pixelu.

Druhý spôsob nastáva v prípade, že index susedného pixelu nepatrí do zoznamu indexov vrcholov. Index susedného pixelu následne pridáme do zoznamu frontier a do hrany, do ktorej patrí prehľadávaný pixel. Po prehľadaní všetkých susedných pixelov pridáme prehľadávaný index do zoznamu visited a odstránime ho zo zoznamu frontier.

Na výstupe podcyklu máme vytvorené hrany a priradené susediace vrcholy k prehľadávanému vrcholu. Vytvoreným hranám pridáme ako prvý index, index vrcholu, ktorý sme prehľadávali. Ako posledný index pridáme hranám hodnotu premennej, ktorá uchovávala index nájdeného vrcholu v danej hrane.

Špecifickým typom hrany je kružnica. Kružnicu detekujeme tým, že zoznam frontier je prázdny a počet vrcholov, ktoré potrebujeme nájsť nie je nulový.

Hranu kružnice vytvoríme nasledujúcim spôsobom. Pokiaľ hodnota dĺžky hrany je väčšia ako 0 a vrchol v hrane nebol nájdený, tak hranu pridáme do hrany kružnice a zároveň ju vymažeme. Opakujeme to pre každú vzniknutú hranu. Súčasne kontrolujeme aj dĺžku hrany kružnice. Dôvodom kontroly je, že pridaním hrany do kružnice, musíme ďalšej pridávanej hrane invertovať poradie indexov pixelov tak, aby v hrane mali indexy pixelov správne poradie. Ako prvý a posledný index k hrane kružnice pridáme index vrcholu, ktorý sme prehľadávali. Hranu kružnice duplikujeme a invertujeme poradie indexov pixelov. Vytvorené hrany pridáme do grafu, ktorý obsahuje všetky hrany podgrafu.

Prehľadávanému vrcholu priradíme vrcholy s ním susediace. Susediace



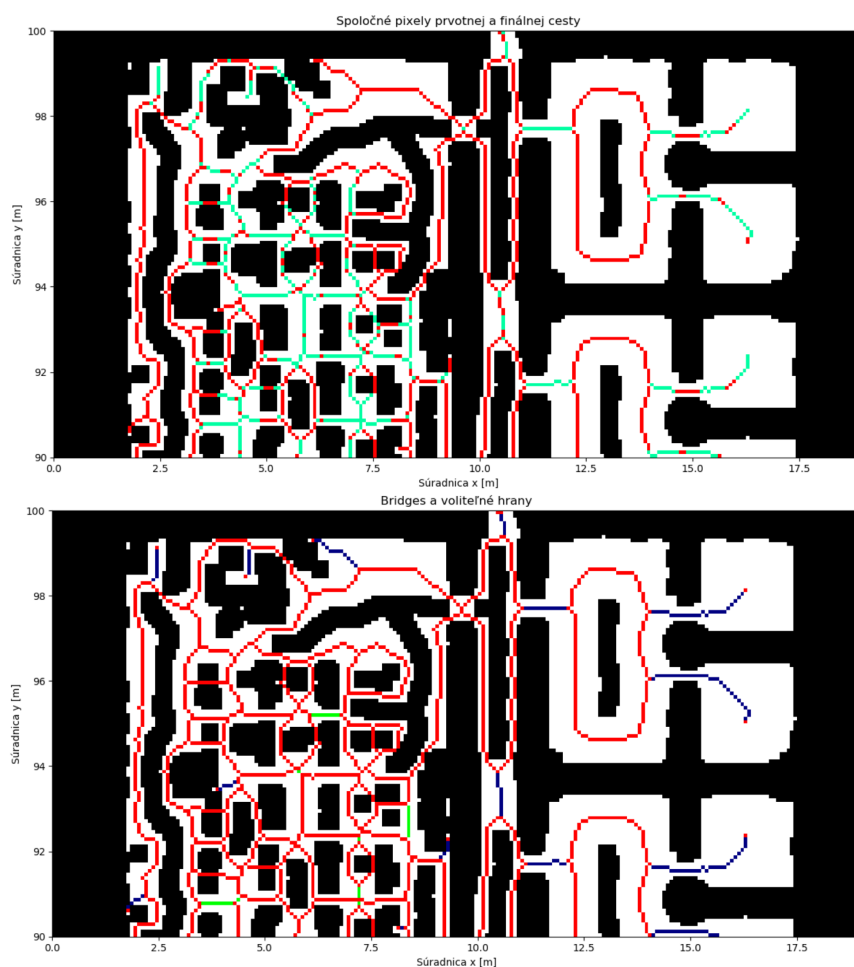
vrcholy sú uložené v premenných, ktoré uchovávajú indexy vrcholov v daných hranách. Vrcholy s priradenými susediacimi vrcholmi pridáme do grafu, ktorý obsahuje prehľadávané vrcholy a k nim prislúchajúce susediace vrcholy.

#### 6.2.4 Príprava orientovaného grafu

Nasledujúca sekcia obsahuje popis algoritmov na prípravu vytvorenia orientovaného grafu. Vstupom do algoritmu je graf hrán neorientovaného podgrafu vytvoreného algoritmom v sekcii 6.2.3 (Zostavenie hrán a susedných vrcholov podgrafu), binárny obraz obsahujúci finálnu roadmapu, zoznam spoločných indexov pixelov, ktorý sme vytvorili v sekcii 6.1.4 (Aplikácia skeletonizácie) a zoznam indexov vrcholov. Hrany, ktoré reprezentujú cesty, po ktorých sa môže pohybovať mobilný robot sme rozdelili na dve kategórie:

Prvou kategóriou sú hrany, ktoré sú obojsmerné. Detekujeme ich algoritmom, ktorý sa zakladá na kontrolovaní počtu spojitých komponentov v binárnom obraze finálnej roadmapy. Na začiatku algoritmu zistíme počet spojitých komponentov v binárnom obraze finálnej roadmapy. V ďalšom kroku vezmeme prvú hranu z grafu hrán. Postupne prechádzame indexy pixelov v hrane a meníme ich hodnotu 1 na hodnotu 0. Výnimkou je prvý a posledný index pixelu hrany. Po zmene hodnôt indexov pixelov v hrane skontrolujeme počet spojitých komponentov. Ak sa hodnota zmenila, hrana je obojsmerná. Postup opakujeme pre každú hranu v zozname. Výstupom algoritmu je zoznam hrán, ktoré majú obojsmernú orientáciu.

Druhou kategóriou sú hrany, ktorým môže užívateľ priradiť smer podľa potreby. Algoritmus vezme graf hrán neorientovaného podgrafu a odstráni hrany, ktoré určil ako obojsmerné. Následne odstráni zo zoznamu spoločných indexov pixelov indexy, ktoré reprezentujú vrcholy. Ďalším krokom je vziať si prvú hranu zo zoznamu hrán. Postupne začneme porovnávať indexy pixelov v hrane s pixelmi, ktoré zostali v zozname spoločných pixelov. Výnimkou je prvý a posledný index pixelu hrany. Ak sa počet indexov pixelov hrany bez prvého a posledného pixelu, rovná počtu nájdených indexov pixelov v zozname spoločných indexov pixelov, hranu pridáme do zoznamu hrán, ktorým užívateľ priradí orientáciu podľa potreby. Postup opakujeme pre každú hranu v zozname. Na Obrázku 6.12 je ukážka spoločných pixelov prvotnej a finálnej kostry (horný obrázok) a ukážka obojsmerných a voliteľných hrán (dolný obrázok).



**Obrázok 6.12:** Na hornom obrázku sú tyrkysovou farbou vyznačené spoločné pixely prvotnej a finálnej cesty. Na dolnom obrázku sú modrou farbou vyznačené cesty, ktoré sú obojsmerné a zelenou farbou sú vyznačené cesty, ktorým užívateľ môže priradiť smer ľubovoľne.

V priebehu experimentálneho overovania nášho riešenia vyvstali vopred nepredpokladané situácie. Pri rozhodovaní o výbere štrukturovacieho elementu sme experimentovali s rôznymi tvarmi a formami štrukturovacieho elementu, s cieľom nájsť tvar štrukturovacieho elementu vhodného reprezentovať mobilného robota Jackal. Pri overovaní návrhu uvádzame výsledný zvolený tvar štrukturovacieho elementu. Súčasťou našich experimentov bola aj úprava mriežky obsadenosti. Obsahovala prídanie a odstránenie dverí do jednotlivých miestností. V dvojitej binárnej dilatácii sme experimentovali s rôznymi veľkosťami bezpečnostnej rezervy, aby sme dosiahli najoptimálnejšie riešenie pre pohyb mobilného robota. Experimentálne kroky v procese overovania nášho riešenia považujeme za cennú pôdu pre ďalšie skúmanie a prácu s problematikou automatickej generácie roadmapy pre pohyb mobilného robota.

## Kapitola 7

### Záver

Naša bakalárska práca sa zaoberá problematikou automatického výrobenia roadmapy pre mobilného robota. Cieľom našej práce bolo navrhnúť postup a implementáciu pre automatické vyrobenie roadmapy a následnú dátovú reprezentáciu pre budúce globálne plánovanie pohybu mobilného robota.

Roadmapu sme vytvárali na pôdoryse podlažia budovy, ktorý bol reprezentovaný mriežkou obsadenosti v binárnom obraze. Navrhnutý postup pre vyrobenie roadmapy pozostával z využitia binárnej dilatácie a skeletonizácie. Binárna dilatácia nám umožňovala detekovať miesta, ktorými mobilný robot dokáže bezkolízne prechádzať. Skeletonizáciu sme využili pre konštrukciu jednopruých a dvojpruých ciest.

Vytvorenú roadmapu sme sa rozhodli reprezentovať dvoma grafmi. Prvý graf obsahuje zostavené hrany (postupnosť pixelov), na základe ktorých sa bude mobilný robot vedieť pohybovať. Na druhom grafe obsahujúcom vrcholy a k nim prislúchajúce susedné vrcholy, sa bude plánovať pohyb mobilného robota. Pri konštrukcii grafov sme využili transformáciu Hit and Miss, ktorá nám umožnila detekovať križovatky na cestách, ktoré reprezentovali vrcholy. Základom vytvorenia algoritmu pre konštrukciu hrán a susedných vrcholov bol algoritmus prehľadávania do šírky.

Prípravou pre budúce globálne plánovanie pohybu mobilného robota na vyrobenej roadmape bolo predpripravenie orientovaného grafu. Jeho predpríprava obsahovala detekovanie obojsmerných ciest a ciest, ktorým užívateľ priradí smer podľa potreby.

Problematika, ktorej sa v práci venujeme si vyžaduje neustále zlepšovanie a napredovanie. Uvádzame niekoľko podnetov na zlepšenie. Prvým odporúčaním je pracovať s väčším počtom vstupných dát pre kvalitnejšie overenie nami navrhnutého riešenia. Ďalším odporúčaním je pracovať s mriežkou obsadenosti, ktorá má väčšie rozlíšenie pixelov, čo zabezpečí väčšiu presnosť. Odporúčame sa zamerať na presnejšie odvodenie určenia bezpečnostnej rezervy, využívanéj pri dvojitej binárnej dilatácii.

Téma mobilných robotov a automatizácie je v súčasnosti výrazne aktuálnou a napredujúcou. Aplikácia kvalitných automatických systémov do priemyselného prostredia môže významne napomôcť v efektívite ľudskej práce.



## Literatúra

- [1] J.LONG and CH.L.ZHANG, “The Summary of AGV Guidance Technology,” *Advanced Materials Research*, vol. 591, pp. 1625–1628, Nov. 2012.
- [2] B.Y.QI, Q.L.YANG, and Y.Y.ZHOU, “Application of AGV in intelligent logistics system,” *Fifth Asia International Symposium on Mechatronics*, pp. 1–5, 2015.
- [3] L.LYNCH *et al.*, “Automated Ground Vehicle (AGV) and Sensor Technologies- A Review,” *Twelfth International Conference on Sensing Technology (ICST)*, pp. 347–352, 2018.
- [4] G.BRESSON, Z.ALSAYED, L.YU, *et al.*, “Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving,” *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, vol. 2, pp. 194–220, Sept. 2017.
- [5] S.SE, D.LOWE, and J.LITTLE, “Local and Global Localization for Mobile Robots using Visual Landmarks,” *International Conference on Intelligent Robots and Systems*, pp. 414–420, Nov. 2001.
- [6] N.BUNIYAMIN, N.SARIFF, W.A.J.WAN NGAH, *et al.*, “Robot global path planning overview and a variation of ant colony system algorithm,” *INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION*, vol. 5, pp. 9–16, 2011.
- [7] A.A.ABDULLA, H.LIU, N.STOLL, *et al.*, “A Backbone-Floyd Hybrid Path Planning Method for Mobile Robot Transportation in Multi-Floor Life Science Laboratories,” *IEEE International Conference on Multisensor*

- Fusion and Integration for Intelligent Systems (MFI)*, pp. 406–411, Sept. 2016.
- [8] D.M.TAVARES and S.J.BACHEGA, “Proposal for an AGV Communication System Using a Cellbot Framework,” *10th IEEE/IAS International Conference on Industry Applications*, pp. 1–6, 2012.
- [9] L.E.KAVRAKI, P.ŠVESTKA, J.C.LATOMBE, *et al.*, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces.,” *IEEE transactions on Robotics and Automation*, vol. 12, pp. 566–580, Aug. 1996.
- [10] R.KRESCH and D.MALAH, “Skeleton-based morphological coding of binary images.,” *IEEE Transactions on Image Processing*, vol. 7, pp. 1387–1399, Oct. 1998.
- [11] H.BLUM, “A transformation for extracting new descriptors of shape.,” *Models for the perception of speech and visual form*, vol. 19, pp. 362–380, May 1967.
- [12] J.C.SERRA, “Skeleton decompositions.,” *Image Algebra and Morphological Image Processing III. International Society for Optics and Photonics*, pp. 376–386, 1992.
- [13] F.Y.SHIH, *Image Processing and Mathematical Morphology: Fundamentals and Applications*. Boca Raton: CRC Press, 2009.
- [14] O.M.ELRAJUBI, I.EL-FEGHI, and M. B. SAGHAYER, “Hit-or-Miss Transform as a Tool for Similar Shape Detection,” *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, vol. 8, pp. 376–386, 2014.
- [15] C. GODSIL and G. ROYLE, *Algebraic Graph Theory*. Berlin: Springer, 2001.
- [16] A.ELFES, “Using Occupancy Grids for Mobile Robot Perception and Navigation,” *Computer*, pp. 46–57, 1989.
- [17] D.NOVÁČEK, “Localization of Mobile Robot Using Multiple Sensors. Prague, 2019. Master thesis,” *Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Cybernetics*.
- [18] Y.DENG, Y.CHEN, Y.ZHANG, *et al.*, “Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment,” *Applied Soft Computing*, vol. 12, pp. 1231–1237, 2012.
- [19] J.YU and S.M.LaVALLE, “Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics,” *IEEE TRANSACTIONS ON ROBOTICS*, vol. 32, pp. 1163–1177, Oct. 2016.

- [20] A.NIEWOLA and L.PODSEDKOWSKI, “L\* Algorithm—A Linear Computational Complexity Graph Searching Algorithm for Path Planning,” *Journal of Intelligent Robotic Systems*, vol. 91, pp. 425–444, Dec. 2018.
- [21] M.B.SUBRAMANIAN, D.K.SUDHAGAR, and G.RAJARAJESWARI, “Intelligent Path Planning Of Mobile Robot Agent By Using Breadth First Search Algorithm,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, pp. 1951–1955, Mar. 2014.
- [22] *Image file reading and writing* [online]. Intel Corporation, Willow Garage, Itseez. [cit. 5. 3. 2020]. Dostupné z: [https://docs.opencv.org/4.2.0/d4/da8/group\\_\\_imgcodecs.html](https://docs.opencv.org/4.2.0/d4/da8/group__imgcodecs.html).
- [23] *Morphological Transformations* [online]. Intel Corporation, Willow Garage, Itseez. [cit. 5. 3. 2020]. Dostupné z: [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_morphological\\_ops/py\\_morphological\\_ops.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html).
- [24] *Morphological structuring element (strel)* [online]. MathWorks. [cit. 5. 3. 2020]. Dostupné z: <https://www.mathworks.com/help/images/ref/strel.html>.
- [25] *Skeletonize* [online]. Stéfan van der Walt. [cit. 10. 3. 2020]. Dostupné z: [https://scikit-image.org/docs/dev/auto\\_examples/edges/plot\\_skeleton.html](https://scikit-image.org/docs/dev/auto_examples/edges/plot_skeleton.html).
- [26] *Matplotlib: Visualization with Python* [online]. John D. Hunter. [cit. 5. 3. 2020]. Dostupné z: <https://matplotlib.org/>.
- [27] *CSV File Reading and Writing* [online]. Python Software Foundation. [cit. 20. 3. 2020]. Dostupné z: <https://docs.python.org/3/library/csv.html>.
- [28] *scipy.ndimage.morphology.binary\_hit\_or\_miss* [online]. Community library project. [cit. 2. 4. 2020]. Dostupné z: [https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.ndimage.morphology.binary\\_hit\\_or\\_miss.html](https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.ndimage.morphology.binary_hit_or_miss.html).
- [29] *scipy.ndimage.measurements.label* [online]. Community library project. [cit. 2. 4. 2020]. Dostupné z: <https://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.ndimage.measurements.label.html>.
- [30] Line End Sub-Types. In: *v6 Examples – Morphology of Shapes* [online]. ImageMagick, 2016 [cit. 3. 4. 2020]. Dostupné z: <http://www.imagemagick.org/Usage/morphology/linejunctions>.
- [31] Line Junction Sub-Types. In: *ImageMagick v6 Examples – Morphology of Shapes* [online]. ImageMagick, 2016 [cit. 3. 4. 2020]. Dostupné z: <http://www.imagemagick.org/Usage/morphology/linejunctions>.





# Dodatok A

## Obsah CD

### Obsah CD

- └─ BakalarskaPraca
  - └─ Bakalarska\_praca\_Adam\_Ukleh.pdf
- └─ 1.BuildingRoadMap
  - └─ Potrebne\_obrazky\_ku\_Zdrojovym\_kodom
    - └─ súbory s príponami .png a .pgm
  - └─ Textove\_Subory\_vytvorene\_Zdrojovymi\_kodmi
    - └─ súbory s príponami .csv
  - └─ Zdrojove\_kody
    - └─ súbory s príponami .py
- └─ 2.BuildingGraph
  - └─ Potrebne\_obrazove\_subory\_k\_Zdrojovym\_kodom
    - └─ súbory s príponami .png a .pgm
  - └─ Potrebne\_textove\_subory\_k\_Zdrojovym\_kodom
    - └─ súbory s príponami .csv
  - └─ Zdrojove\_kody
    - └─ súbory s príponami .py



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Ukleh** Jméno: **Adam** Osobní číslo: **453060**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra kybernetiky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Plánování pohybu mobilního robotu v průmyslovém prostředí**

Název bakalářské práce anglicky:

**Mobile Robot Motion Planning in Shopfloor**

Pokyny pro vypracování:

1. Seznamte se s metodami plánování pohybu mobilních robotů, zvláště pak s metodami dostupnými v ROS.
2. Vyberte nebo navrhnete metodu, která bude vhodná pro plánování pohybu dopravních vozíků v továrně.
3. Metodu implementujte a vyzkoušejte na reálném robotu.
4. Výsledky zhodnoťte, navrhnete další zlepšení.

Seznam doporučené literatury:

1. Elias Xidias, Paraskevi Zacharia and Andreas Nearchou: Path Planning and scheduling for a fleet of autonomous vehicles, Robotica 2018
2. Shraavan Krishnan, Govind Aadithya R, Rahul Ramakrishnan, Vijay Arvinth, Sivanathan K: A Look at Motion Planning for Autonomous Vehicles at an Intersection 2018
3. Z. Shiller ; Y.-R. Gwo: Dynamic motion planning of autonomous vehicles IEEE Transactions on Robotics and Automation (Volume: 7, Issue: 2, Apr 1991)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Vladimír Smutný, Ph.D., robotické vnímání CIIRC**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.01.2019**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Vladimír Smutný, Ph.D.  
podpis vedoucí(ho) práce

doc. Ing. Tomáš Svoboda, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta