

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/I/2019/36086129633547268

IS MIKRO A MAKRO NAVIGÁCIE PRE ZRAKOVO POSTIHNUTÝCH

Diplomová práca

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/I/2019/36086129633547268

IS MIKRO A MAKRO NAVIGÁCIE PRE ZRAKOVO POSTIHNUTÝCH

Diplomová práca

Študijný program: Informačný Manažment (Jednoodborové štúdium,
Inžiniersky II. st., denná forma)

Študijný odbor: Informačný Manažment

Školiace pracovisko: Katedra aplikovanej informatiky

Vedúci záverečnej práce: Ing. Jaroslav Kultán, PhD.

Bratislava 2019

Bc. Juraj Lipovský

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol
všetku použitú literatúru.

Dátum:

.....

(podpis študenta)

Pod'akovanie

Ďakujem Ing. Jaroslavovi Kultanovi, PhD. za odborné konzultácie a pomoc aj napriek neskorým nočným hodinám a ochotu riešiť problémy cez Skype pri vypracovaní tejto diplomovej práce.

Abstrakt

LIPOVSKÝ, Juraj: IS Mikro a Makro Navigácie pre zrakovo postihnutých

Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra aplikovanej informatiky.

Konzultant: Ing. Jaroslav Kultán, PhD.

Bratislava: FHI EU, 2019, 50 s.

Cieľom diplomovej práce je návrh a vytvorenie integrovaného informačného systému určeného pre navigáciu nevidiacich a zrakovo postihnutých po meste, .

Prvá kapitola sa zameriava analýze existujúcich navigačných systémov a ich historického vývoja, rozdelenie a vymedzenie termínov „mikro navigácia“ a „makro navigácia“.

V ďalšej časti sa charakterizuje hlavný cieľ tejto diplomovej práce s rozdelením na tri podciele.

Tretia kapitola zachytáva nástroje a metódy riešenia, analyzuje existujúce riešenia v danej problematike, či už pri mikro alebo makro navigácii. Nachádza sa v nej analýza a výber vhodných hardwarových aj softwarových riešení pre tento systém vrátane ich porovnaní.

Štvrtá kapitola popisuje výsledky práce spojenej s vývojom a návrhom tohto integrovaného systému, opis použitých algoritmov, vizuálny návrh zariadenia, konkrétny popis zariadenia, jeho portov a techniky, rovnako ako popis a analýzu softvérových riešení a diskusiu o ďalšom rozvoji zariadenia v budúcnosti.

V piatej kapitole sa zameriavame na opis vízie projektu a možnostiach rozvoja daného informačného systému v budúcnosti, pri zohľadnení výsledkov výskumu a následného testovania a posúdenia všetkých kladov aj záporov spracovaného riešenia. V tejto kapitole je opísaný záver práce ako aj zhrnutie vykonanej práce.

Záverečná kapitola sa zaoberá použitou literatúrou a zdrojmi.

Výsledkom riešenia danej problematiky je že, aplikácia využíva už existujúce komunikačné systémy, umožňuje však zadávanie cieľa a postupnú navigáciu do cieľa. Okrem toho sleduje vzdialenosť ku prekážkam a poskytuje zvukový signál o ich vzdialenosti.

Kľúčové slová: mobilné navigačné prostriedky, meranie vzdialeností, smart prístroje, orientácia v meste, infračervené body, rekonštrukcia scény.

Abstract

LIPOVSKÝ, Juraj: IS Micro and Macro Navigation for blind

University of Economics in Bratislava. Faculty of Business Informatics; Department of Applied Informatics.

Supervisor: Ing. Jaroslav Kultán, PhD.

Bratislava: FHI EU, 2019, Bratislava: FHI EU, 2019, 50 p.

The aim of the thesis is to design and create an integrated information system designed for navigation of blind and visually impaired people in the city.

The first chapter focuses on the analysis of existing navigation systems and their historical development, the division and definition of the terms "micro navigation" and "macro navigation".

In the next part, the main goal of this bachelor thesis is divided into three sub-goals.

The third chapter captures the tools and methods of the solution, analyses the existing solutions in the given issue, whether in micro or macro navigation. It contains analysis and selection of suitable hardware and software solutions for this system, including their comparison.

The fourth chapter describes the results of the work related to the development and design of this integrated system, the description of the algorithms used, the visual design of the device, the specific description of the device, its ports and techniques, as well as the description and analysis of software solutions.

In Chapter 5, we focus on describing the vision of the project and the possibilities for future development of the information system, taking into account the results of the research and subsequent assessment of all the pros and cons of the processed solution.

The fifth chapter describes the conclusion of the work with a summary.

The final chapter deals with the literature and sources used.

The result of the solution of this issue is that the application uses already existing communication systems, but it allows entering the target and gradual navigation to the target. In addition, it monitors the distance to obstacles and provides an audible signal of their distance.

Keywords: mobile navigation aids, distance measurement, smart devices, city orientation, infrared points, scene reconstruction.

Obsah

Čestné vyhlásenie.....	8
PodĎakovanie.....	9
Abstrakt.....	7
Abstract	8
Obsah	9
Úvod	12
1. Analýza existujúcich navigačných riešení.....	14
1.1. Rozdelenie mikro a makro navigácie	14
Makro navigácia	14
1.1.1. Mikro navigácia	16
1.1.2. Cieľ práce	18
2. Výber nástrojov a metód riešenia	19
3.1. Analýza existujúcich technických nástrojov.....	19
3.1.1. Existujúce makro-navigačné systémy	19
3.2. Výber hardvérových prostriedkov	22
3.2.1. Ultrazvukové senzory	22
3.2.2. LIDAR	23
3.2.3. Stereo vĺzia	24
3.2.4. Sledovanie infračervených bodov	24
3.2.5. Procesorová jednotka	26
3.3. Porovnávací diagram.....	27
3.4. Výber softvérových prostriedkov.....	27
Microsoft Kinect SDK.....	28
Maximálna hodnota	28
Vysoký exponent.....	28

Intervalový histogram	29
4. Výsledky práce.....	30
4.1. Vývoj softvéru a testovacieho prostredia	30
3.4.4. Vizualizácia údajov v matici.....	30
Ukladanie hĺbkovej mapy na disk.....	31
Testovacie prostredie.....	32
4.1.1. Text-to-Speech modul.....	34
4.1.2. 4.2. Testovanie navigácie	34
4.1.4. Testovanie vnútri	35
4.2.1. Testovanie vonku	36
4.2.2. 4.3. Výsledky výskumu	37
Analýza existujúcich technických a programových prostriedkov	37
4.3.1. Metodológia postupu využívania možného prostriedku	37
4.3.2. Výber technických prostriedkov pre realizáciu.....	38
4.4. Harmonogram	39
5. Vízia projektu.....	40
5.1. Skenovania okolia	40
5.2. Vyhodnocovacie algoritmy	40
5.3. Vývoj zariadenia	42
5.4. Komunikácia s používateľom	42
Záver.....	43
Obrázková príloha	44
Zoznam kódov	49
5.5. Testovanie priemerom.....	49
5.6. Testovanie maximálnou hodnotou	49
5.3. Testovanie vysokým exponentom	50

5.4.	Kompresia a dekompresia súborov s hĺbkovou mapou.....	50
5.5.	Integrácia Microsoft Kinect SDK	51
Zoznam tabuliek		52
5.1.	Výber hĺbkovej kamery	52
5.2.	Výber Mini PC.....	54
Použitá literatúra, zdroje.....		55

Úvod

Psychológovia označujú zrak ako zdroj 80% vnímania okolitého sveta. Slepí a zrakovo postihnutí ľudia považujú za nesmierne zložité a komplikované riešiť každodenné problémy a úlohy, čo si ľudia nepatriaci do tejto skupiny častokrát len ťažko predstavujú. Vnímanie okolia zrakom môžeme považovať za špecifickú informáciu prijatú očami, presúvanú očným nervom a spracovanú v mozgu človeka. Vedná disciplína zvaná Informačné Technológie sa zaoberá práve informáciami, ich prenosom, spracovaním, uchovávaním a. i., preto, by to mala byť práve táto vedná disciplína, ktorá dokáže vyriešiť poruchy prijímania vizuálnej informácie z okolia a jej spracovania a transformovať ju na takú formu, ktorú dokážu zrakovo postihnutí jedinci prijať a efektívne využiť. To znamená vytvorenie integrovaného systému spájajúceho priestorovú orientáciu, aktívne vyhýbanie sa prekážkam a nebezpečenstvám a navigáciu v najefektívnejšej trase k požadovanému cieľu.

Avšak dnes, v dobe obrovského rozmachu informačných technológií a informačných systémov, preberajú na seba časť každodenných úloh jeho používateľov a tým dovoľujú sústrediť sa na komplikovanejšie, zložitejšie a zodpovednejšie činnosti. Tieto výhody vedia hendikepovaní ľudia oceniť dvojnásobne, keďže asistenciu pri svojom každodennom živote vyžadujú zďaleka na vyššej úrovni. Podľa môjho názoru, jedna z najdôležitejších úloh vyspelej spoločnosti je postarať sa o jej časti alebo členov, ktorí to sami nedokážu. Ak dokážu Informačné Technológie priniesť hmatateľné zlepšenia a výsledky v živote členov našej spoločnosti, nedokážem nájsť dôvod, prečo by nemali byť vyvinuté pre tieto potreby aj ako súčasť procesu Informatizácie spoločnosti.

Zrakové vnímanie okolitého sveta považujeme za konkrétny a presne prebiehajúci proces prijatia, spracovania a vyhodnocovania informácie počínajúc ľudským okom, pokračujúc očnými nervami, až po stredný mozog, ktorý je zodpovedný za prvé spracovanie zraku a sluchu. Zrakovo postihnutých a nevidiacich ľudí môžeme v najrýdzejšej podobe klasifikovať ako jedincov s poruchou, v niektorom v týchto častiach procesu. V tejto práci sme sa snažili pomôcť predovšetkým ľuďom s poruchou už v počiatočných štádiách prijímania samotnej informácie a využiť predovšetkým informačné technológie na transformáciu týchto informácií do podoby, ktorú je znevýhodnený používateľ schopný ďalej spracovať. Dôvod, prečo sa zameriavame predovšetkým na ľudí s poruchou očí a očných nervov, je vysvetlený ďalej v práci.

Práca má jednoduchú štruktúru a začína teoretickým úvodom do problematiky navigácie, jej históriou, využitím a postupným vývojom naprieč dejinami. Pokračuje vytýčením cieľov, teda akýchsi míľnikov, ktoré je potrebné zvládnuť pred samotným dosiahnutím konečného cieľu. Metódy a nástroje, ktoré sme k danému vývoju použili, sú rovnako popísané v tretej kapitole. Nasleduje zhrnutie výsledkov a poznatkov nadobudnutých počas výskumu.

1. Analýza existujúcich navigačných riešení

Pre uvedenie akéhokoľvek produktu na trh, je potrebné poriadne sa porozhliadať po už funkčných a zavedených riešeniach v našej problematike. Tento postup nám nielen zaručí odpoveď na zmysluplnosť prípadne nadbytočnosť a redundanciu nášho výskumu a práce, ale aj pomôže k istej dávke inšpirácie, na základe ktorej, môžeme náš výsledný produkt zlepšovať a vyvíjať. V prvej kapitole sa preto budeme venovať prieskumu trhu existujúcimi riešeniami, ale aj ich podrobnou a detailnou analýzou. V tejto kapitole sa rovnako nebudeme venovať len celkovým riešeniam našej problematiky navigácie pre nevidiacich a zrakovo postihnutých, ale aj čiastkovým riešeniam, nevyhnutným k docieleniu celkového riešenia.

1.1. Rozdelenie mikro a makro navigácie

Pre lepšie pochopenie a spracovanie problematiky je potrebné vytýčiť si nasledujúce pojmy, teda navigáciu a orientáciu v priestore rozdeliť na dve skupiny. Skupiny sú odlišené podľa veľkosti priestoru, v ktorom sa orientujeme a presnosti s akou potrebujeme, aby nás systém navigoval. Tieto pojmy budú v ďalších častiach práce kľúčové a to práve kvôli podstate a analógii ich prepojenia, terminológii rozlíšenia typu navigácie a komplexnosti systému ako metodologickej pomôcky.

1.1.1.

Makro navigácia

Tento typ navigácie a priestorovej orientácie zahrňuje systémy, praktiky a metodológie využívané pri navigácii vo veľkých priestoroch, najčastejšie mimo budov a stavieb, teda v otvorených vonkajších priestranstvách. Prirovnať by sme ju mohli ku bežne používaným systémom práce s mapami pri navigovaní cesty do svojho cieľa v rámci mesta, krajiny, štátu, kontinentu ale aj pri moreplavbe. Túto funkciu dnes vo veľkej miere zastúpili elektronické systémy vybavené digitalizovanými mapami veľkého rozsahu a zároveň vysokej podrobnosti, pričom pri určovaní čo najpresnejšej polohy využívajú údaje vysielané zo satelitov systému GPS alebo GLONASS, upresnené sieťami GSM a Wi-Fi (v rámci ich dostupnosti). Na presné určenie polohy sú používané rôzne metódy postupného výpočtu.

Pilotáž

Pilotáž (piloting) alebo optická navigácia spolu s trianguláciou (triangulation) je navigácia v pobrežných a/alebo vnútrozemských vodách s častým vizuálnym určovaním pozície podľa geografických a hydrografických navigačných znakov. Označuje sa aj ako ^{1.1.1.1}triangulácia, pretože poloha plavidla (1. vrchol trojuholníka) sa zameriava pomocou dvoch vizuálnych navigačných znakov (2. a 3. bod trojuholníka) určovaním (opačného) azimutu navigačných znakov voči plavidlu. Spolu s vizuálnymi znakmi sa používajú aj ďalšie metódy, ako napríklad meranie hĺbky.

Okrem určovania polohy zahŕňa pilotáž aj vytyčovanie kurzu, napríklad so započítavaním otočných charakteristík plavidla, prílivu a odlivu, riečnych alebo morských prúdov, vytyčovanie a zameriavanie kotviska, sledovanie a dodržiavanie plavebnej signalizácie a pod. V súčasnosti je dôležitým doplnkom pilotáže hlavne radarová navigácia.

Astronomická navigácia

^{1.1.1.2.} Astronomická navigácia (celestial navigation) zahŕňa presné meranie pozície (výšky nad obzorom) astronomických telies, meranie času, zakresľovanie astronomických pozičných línií vypočítaných za pomoci astronomických tabuliek a almanachov. Používa sa najmä na otvorenom mori ako doplnok elektronickej (najmä satelitnej) navigácie.

^{1.1.1.3.}

Rádio navigácia

Rádio navigácia (radionavigation) je navigácia za pomoci rádiových vĺn, ktoré vysielajú rádio majáky a pozemné rádiové navigačné stanice. Rádiové vlny podliehajú pri svojom šírení rušivým fyzikálnym vplyvom atmosféry a zemského povrchu, čo ovplyvňuje dosiahnuteľnú presnosť a dosah signálu. Rádiové zisťovanie smeru (radio direction finding, RDF) je určovanie relatívneho alebo absolútneho azimutu od plavidla k rádio majáku pomocou smerovej antény. ^{1.1.1.4.} Chyba námeru dosahuje za dobrých podmienok maximálne 2° na vzdialenosť do 150 nm.

Hyperbolické systémy

Hyperbolické rádio navigačné systémy (hyperbolic systems) umožňujú určenie polohy plavidla meraním časových rozdielov medzi rádiovými impulzmi vyslanými v rovnakom čase z minimálne troch vysielacích staníc. Porovnaním časového rozdielu vždy medzi signálmi z dvoch staníc je možné zakresliť pozičnú líniu ktorá má tvar hyperboly. Meraním troch staníc

je možné určiť polohu plavidla v priesečníku aspoň dvoch pozičných línií s presnosťou 0,25 nm. Typickými hyperbolickými systémami sú napríklad LORAN C alebo OMEGA.

Radarová navigácia

Radarová navigácia (radar navigation) používa radar nainštalovaný na plavidle na určovanie vzdialeností a azimutov objektov so známou polohou. Z týchto údajov sa potom určuje poloha plavidla. Tento proces sa líši od použitia radaru ako systému na lokalizáciu prekážok a predchádzanie kolíziám.

Satelitná navigácia

Satelitná navigácia (satellite navigation) sa používa na určovanie pozície plavidla na základe signálu z umelých družíc Zeme. Najznámejším satelitným navigačným systémom je systém GPS (Global Positioning System). Družice systému GPS vysielajú okrem pozičného signálu aj presný časový signál. Systém umožňuje zameranie polohy prijímača s presnosťou $\pm 10 - 15$ m. V tejto práci sme tiež používali satelitnú navigáciu pre vyriešenie problematiky navigovania vo veľkých priestoroch nevidiacich a slabozrakých, konkrétne išlo o implementáciu aplikácie Sygic Offline GPS Navigation. Navigáciu od firmy Sygic sme si vybrali z dôvodu, že pri porovnávaní spomedzi kandidátov ako Google Maps, Waze, HERE a Sygic, si najlepšie počínala vo forme offline navigácie, teda tento softvér nepotreboval žiadnu konektivitu so serverom poskytovateľa služby a všetky úlohy sa vykonávali priamo na zariadení používateľa. Toto zvládajú aj vyššie spomenuté, konkurenčné služby, avšak zďaleka nie na úrovni ako navigácia od firmy Sygic.

Mikro navigácia

Systém navigácie a priestorovej orientácie zameraný na výrazne menšie vzdialenosti ako tomu bolo pri makro navigácii nazývame mikro navigácia. Pre tento typ priestorovej orientácie nemá presne určený operačný rádius ani používaný priestor, ide len o systémy využívané pri orientovaní v danej ceste po jej najkratšej trase s dôrazom na vyhýbanie prekážkam, kolíziám a nebezpečenstvám. V našej problematike zrakovo postihnutých ľudí je na túto činnosť najčastejšie využívaná tzv. slepecká palička, teda biela palica, ktorú zrakovo postihnutý drží v jednej ruke a jej pomocou zisťuje prekážky v ceste. Iný príklad využitia mikro

navigácie v praxi je pri parkovaní automobilu. Jeden z hlavných cieľov a princípov vývoja tejto práce bolo aj úspešné zvládnutie problematiky mikro navigácie pre aktívne informovanie používateľa systému o hrozbách kolízie s okolitými objektami. V mojej bakalárskej práci sme sa zameriavali na použitie ultrazvukových senzorov za účelom detekcie objektov a prekážok v okolí, v tejto práci sme úroveň povýšili na rôzne iné technológie schopné zrekonštruovať okolitú scénu a vhodným spôsobom transformovanú informáciu odovzdať slabozrakému používateľovi.

2. Cieľ práce

Cieľom práce je vytvoriť aplikáciu na pomoc nevidiacim, ktorá bude zahŕňať pomoc pri presune na veľké vzdialenosti – makroorientáciu a pomoc pri obchádzaní blízkych prekážok – mikroorientáciu. Spočívali vo vytvorení aplikácie, pomocou ktorej by slabozrakí a nevidiaci ľudia boli schopní samostatného pohybu a navigácie, bez nutnosti akejkoľvek ďalšej pomoci. Samotnú problematiku sme si preto hneď na začiatku rozdelili na viacero menších úloh .

Prvou úlohou je analyzovať existujúce technické a programové prostriedky využívané pri v rôznych problémoch orientácie. V tejto časti je potrebné podrobne analyzovať silné a slabé stránky existujúcich zariadení, vzájomne ich porovnať a na základe komplexného porovnávania nájsť niektoré problémy, ktoré by sme chceli odstrániť.

Druhou úlohou je vytvoriť metodológiu tvorby a postupu využívania možného prostriedku, vytvoreného na základe existujúcich mobilných zariadení alebo špecifických technických prostriedkov.

Treťou úlohou je vybrať vhodné technické prostriedky potrebné na realizáciu uvedených zariadení v súlade s vytvorenou metodikou realizácie nových zariadení. Hlavným prínosom tejto časti je popis vybraných technických zariadení – procesorov, periférnych zariadení, programových prostriedkov.

Štvrtou úlohou je vytvoriť model zariadenia pomáhajúceho nevidiacemu v mikro a makro navigácii na základe moderných poznatkov a technológií.

Na základe uvedených cieľov je potrebné vytvoriť jednotlivé časti v tejto práci, v ktorých bude uvedené riešenie daných podcieľov.

3. Výber nástrojov a metód riešenia

V predchádzajúcej kapitole sme si zadefinovali jednotlivé ciele potrebné k riešeniu našej problematiky, rovnako ich rozdelili na menšie podciele a popísali ich význam. Pre ich splnenie je však potrebné vybrať nástroje a metódy, ktoré budeme pre naše potreby využívať. V tejto kapitole teda budeme postupne prechádzať hardvérovými aj softvérovými prostriedkami, potrebnými ku správne navigovaniu nevidiacich a zrakovo postihnutých.

3.1. Analýza existujúcich technických nástrojov

V rámci každého projektu týkajúceho vytvorenia a integrácie akéhokoľvek typu produktu, je nutné si hneď na začiatku spraviť prieskum trhu po existujúcich riešeniach. To sa robí z dôvodu vyhýbania sa duplicitnej práci, prípadne okamžitému neúspechu z dôvodu veľkého náskoku konkurencie. Prieskum trhu tiež slúži ako akýsi ukazovateľ dopytu a ponuky v danom segmente, čím pomáha ako istý typ ukazovateľa dodatočného smerovania projektu prípadne inšpirácie. V našej problematike navigovania slabozrakých je trh v tomto momente oveľa menší a nenasýtený, v porovnaní s inými segmentami napríklad GPS navigácie, prípadne rekonštrukcie okolitých scén ako samostatný odbor. V tejto podkapitole sa preto budeme venovať existujúcim možnostiam, vytvorených pre nevidiacich a slabozrakých, či už pre ich skvalitnenie života ale hlavne doterajším možnostiam ich navigácie.

3.1.1.

Existujúce makro-navigačné systémy

V tejto podkapitole analýze existujúcich technických nástrojov sa budeme ďalej venovať makro navigačným prostriedkom slúžiacim k navigácii nevidiacich a slabozrakých vo veľkých priestoroch, ale aj pre zlepšenie kvality v iných aspektoch života.

BlindSquare

Je to svetovo najpopulárnejšia mobilná aplikácia pre slepých a zrakovo postihnutých. Popisuje okolie, zaujímavé body a križovatky ulíc pri cestovaní. BlindSquare používa hlasový syntetizátor pri navigovaní a ovláda sa použitím audio menu pri použití ľubovoľného headsetu alebo reproduktora, to znamená, že užívateľ sa nemusí vôbec dotýkať obrazovky mobilného zariadenia pri samostatnom cestovaní. Výhodou je možnosť kombinovať aplikáciu s inou

navigačnou aplikáciou a veľmi veľká globálna rozšírenosť používania. Nevýhodou je podpora zatiaľ len iOS mobilného operačného systému.

Siri/S Voice/ Google Assistant, hlasový analyzátor

Použitie hlasového asistenta na platforme umelej inteligencie je jedným z posledných a najmodernejších prvkov, využívajúcich zber a analýzu dát, ktorý vývojári integrovali do mobilných telefónov s operačným systémom iOS a Android. Podobní asistenti boli kedysi integrovaní skôr ako ukážka technologického pokroku a prezentácia súčasného stavu umelej inteligencie pre riešenie každodenných problémov používateľov, avšak časom sa zistilo, že ich využívanie ocenia najmä postihnutí či rôzne hendikepovaní používatelia zariadení.

„Vidíme Google Assistant ako dôležitý vývoj. Je ešte príliš skoro, avšak už teraz môžeme pozorovať obrovský pokrok. Ľudia posúvajú hranice využívania ich mobilného zariadenia. Rozširuje sa množina využívania mobilného telefónu, čo je to, na čo sa sústreďíme.“ – Sundar Pichai, CEO Google Assistant.

3.1.1.3. KAPSYS SmartVision

Kapsys SmartVision je smartfón s upraveným systémom Android, ktorý ocenia používatelia so zrakovým hendikepom. Má dotykový displej, telefónnu klávesnicu, hlasové funkcie a kontrastnú grafiku prostredia. Kapsys SmartVision patrí medzi tie zariadenia, ktoré sa pre svoje zameranie sa na konkrétnu cieľovú skupinu a špeciálne funkcie nazývajú kompenzačnými. Ich cieľom je pomáhať ľuďom s rôznymi hendikepmi a potrebami pri realizovaní konkrétnych činností. SmartVision sa stará o pohodlnejšiu prácu s telefónom, pomáha teda pri komunikácii, vyhľadávaní či navigácii. Kapsys SmartVision má kompaktné rozmery a jednoduchý dizajn podriadený účelu. Dotykový displej má uhlopriečku 3,5 palcov využíva farebný a dotykový IPS panel. Displej je prekrytý priehľadným plastom, ktorý je od zobrazovacej plochy oddelený väčšou medzerou. Bude sa pod ňou hromadiť prach. IPS panel poskytuje dobré farby aj pozorovacie uhly, ale nedosahuje kvalít AMOLED panelov, ktoré má vo svojich zariadeniach Samsung. IPS je stále iba LCD, teda pixely nesvietia, sú iba podsvietené. V exteriéri, ak je podsvietenie slabšie ako okolité svetelné podmienky, to môže byť problém.

Systém Android je vybavený špeciálnym prostredím od francúzskej spoločnosti Kapsys. To zahŕňa nielen úpravu grafiky a dizajnu systému, ale aj ďalší špeciálny softvér; napríklad ABBY Finereader, navigačný softvér Kapten či hlasový syntetizátor Laura.

Hlasový asistent v tomto prípade neplní pokyny od používateľa, ale je v prvom rade hlasovou spätnou väzbou. Laura číta to najpodstatnejšie, čo sa práve nachádza na displeji a usmerňuje tak aj celkom nevidiaceho používateľa. Laura číta po slovensky veľmi zrozumiteľne.

Súčasťou hlasových funkcií telefónu je aj možnosť diktovania, teda prepis hovoreného slova do digitálneho textu. Na využitie tejto možnosti však musí byť telefón pripojený na internet (napr. cez Wi-Fi).

Aplikácia ABBY Finereader dokáže čítať texty, na ktoré je zameraný fotoaparát telefónu. Iná aplikácia využíva fotoaparát, aby oznámila aj názov farby, na ktorú je práve namierený, prípadne dokáže po zvolení konkrétnej farby zo zoznamu oznámiť jej nález v prostredí.

Fotoaparát je vybavený funkciami, ktoré slabozrakí potrebujú pri čítaní. Obrázok je možné pomocou aplikácie Lupa v priamom prenose prevádzať do vysokého kontrastu, negatívu a do rôznych farebných úprav tak, aby bol pre používateľa rozpoznateľný. Zrakových porúch je totiž veľké množstvo a každá sa môže kompenzovať iným spôsobom.

Dôležitým detailom je skutočnosť, že sa na tento Android dajú inštalovať aplikácie z obchodu Google Play.

3.1.1.4.

TapTapSee

Ide o mobilnú aplikáciu využívajúcu kameru smartfónu, špeciálne navrhnutú pre slepých a zrakov o postihnutých užívateľov iOS mobilných operačných systémov, poháňaná CloudSight API. Aplikácia používa kameru zariadenia na fotografovanie objektov, ich identifikáciu a opis pomocou hlasového syntetizátora pre používateľa.

V TapTapSee používateľ dva krát klikne na obrazovku, tým odfotí vizuálnu podobu dvoj alebo trojdimenzionálneho objektu pred kamerou v ľubovoľnom uhle. Ten má následne presne analyzovaný a definovaný objekt v priebehu pár sekúnd. Hlasový analyzátor potom identifikáciu odfotografovaného objektu presne opíše používateľovi.

DotWalker

DotWalker je aplikácia pre Android, ktorá sa venuje špeciálnym potrebám cestovania zrakovo postihnutých užívateľov. Táto aplikácia používa čítačku obrazovky Talkback. Samotný DotWalker používa priamy výstup reči, zvukové upozornenia, vibrácie a text s vysokým kontrastom, ktorý možno podľa potreby zväčšiť.^{3,1,15}

Na poskytnutie informácií o polohe je využívaný senzor GPS. Geomagnetický snímač sa používa na funkcie kompasu a akcelerometra na spustenie niektorých činností používateľa.

Aplikácia je k dispozícii v dvoch verziách. Bezplatná verzia prichádza s obmedzenou funkčnosťou, napríklad spracovanie len do 30 bodov a zakázaný prístup k funkciám súvisiacim s webom. Platená verzia sa nachádza v službe Google Play ako DotWalker Pro.

3.2. Výber hardvérových prostriedkov

Hlavná časť výskumu tejto práce pramení z vhodnej rekonštrukcie okolitej scény pre potreby mikro navigácie (viď kapitola 1.1.1.), preto začneme s výberom vhodných hardvérových prostriedkov na skenovanie a mapovanie najbližšieho okolia používateľa tohto systému (viď Obrázok 3). Pre každý prostriedok doplníme výhody, nevýhody a vhodnosť použitia v našej práci.

3.2.1.

Ultrazvukové senzory

Ako prvé musíme spomenúť masovo rozšírené pre svoju jednoduchosť a spoľahlivosť ultrazvukové senzory (viď Obrázok 1). Tieto senzory dobre poznáme napríklad z parkovacích senzorov umiestnených v nárazníkoch moderných automobilov a sústava takýchto senzorov pomáha používateľovi merať a dodržiavať vzdialenosti auta od prekážok pri parkovaní s dostatočnou presnosťou. Fungujú na princípe vyslania vlnenia v spektre ultrazvuku, ktoré sa od pevných prekážok odráža. Následne senzor zachytí odrazenú vlnu, pričom obsluhujúci softvér pozná čas vyslania a opätovného zachytenia vlny. Za predpokladu, že poznáme rýchlosť tejto vlny a čas medzi vyslaním a odoslaním, je jednoduché si vypočítať prejdenú vzdialenosť. Tento výpočet vykonáva riadiaca jednotka v automobiloch, prípadne zvládne ho veľká skupina mikroprocesorových jednotiek pre svoju nenáročnosť. Výstup zobrazený na Obrázku 2.

Výhody:

- jednoduchosť aplikácie a použitia,
- nízka cena,
- malá pravdepodobnosť interferencie a rušenia.

Nevýhody:

- schopnosť zmerať vzdialenosť len v jednom smere.

Na základe vyššie spomenutých vlastností sme usúdili, že použitie týchto senzorov by bolo síce vhodné na jednoduché úkony, avšak pre komplexné riešenie mikro navigácie a priestorovej orientácie sú príliš jednoduché a málo presné. Hlavným problémom je absencia schopnosti rozlišovať viac informácií ako len jednoduchú vzdialenosť, teda museli by sme použiť sústavu týchto senzorov a upraviť frekvenčný rozsah každého jedného, kvôli rušeniu. Takéto riešenie by však bolo rozmerovo veľmi náročné a neelegantné.

LIDAR

3.2.2. Ide o pokročilejšiu technológiu oproti jednoduchým ultrazvukovým senzorom. LIDAR funguje na princípe pohyblivého laserového lúča, najčastejšie o vlnovej dĺžke 600-1000 nm, pričom toto zariadenie postupne osvetľuje cieľovú plochu. Senzor následne sníma čas a vlnovú dĺžku odrazeného lúča a na základe rozdielnych hodnôt oproti vyslaným, je vypočítaná a zrekonštruovaná 3D scéna, kam lúč dopadol. Fotka LIDARu na Obrázku 3. Najčastejšie sa táto technológia používa napríklad v geodézii, mapovaní veľkých plôch z leteckých záberov, autonómnych vozidlách a vo vojenskom priemysle v laserom navádzaných raketách.

Výhody:

- presné meranie vzdialenosti viacerých bodov scény v krátkom čase,
- malá pravdepodobnosť interferencie a rušenia,
- použitie za takmer všetkých svetelných podmienok.

Nevýhody:

- vysoká energetická náročnosť oproti ostatným alternatívam,
- mapuje prostredie len v jednej rovine.

Po preskúmaní všetkých kladov aj záporov sme dospeli k názoru, že použitie tohto typu technológie je stále nedostatočné, najmä kvôli záberu len jednej roviny zo scény, teda

absencia skenovania okolia ako celku, ale aj hlučnému a energeticky náročnému pohonnému ústrojenstvu väčšiny dostupných LIDAR-ov.

Stereo vĺzia

Tento typ skenovania okolia má za úlohu čo naj dôvernejšie napodobiť a kopírovať funkciu páru očí u živočíchov. Jedná sa o sústavu dvoch RGB kamier v jednej línii, so známym uhlom záberu a známou vzdialenosťou jednej od druhej. Obraz je ďalej spracovávaný v procesorovej jednotke, kde má softvér za úlohu porovnať posun kľúčových prvkov výstupného obrazu jednej kamery oproti druhej. Čím viac je jeden prvok posunutý v obraze jednej kamery oproti druhej, tým je ku sústave kamier bližšie. Táto metóda nám dokáže efektívne zrekonštruovať komplexnú scénu, avšak má veľký problém s absolútnym meraním vzdialeností. Jej výstupom je skôr porovnanie jednotlivých prvkov s informáciou o ich vzájomnej vzdialenosti. Zapojenie kamier znázornené na Obrázku 4.

Výhody:

- mapovanie scény ako celku,
- postačia bežne dostupné a lacné RGB kamery.

Nevýhody:

- veľmi vysoké nároky na výpočtovú kapacitu procesorovej jednotky,
- závislosť od svetelných podmienok.

Napriek zdanlivej úspešnosti tejto technológie sa nejedná o úspešného kandidáta v našom teste a to kvôli ohromne vysokej výpočtovej náročnosti samotného algoritmu určeného na porovnávanie prvkov v obraze. Pri našich testoch bolo na spracovávanie Full HD obrazu potrebné zaobstarať stolový počítač s grafickým čipom NVIDIA GTX 1080, pokiaľ sme chceli zabezpečiť real-time použitie 24-30 fps. Diskusie o aplikácie na mobilné zariadenia teda odložíme o pár rokov, potom však táto technológia vyzerá naozaj sľubne.

Sledovanie infračervených bodov

Použitie tejto technológie objavilo veľké uplatnenie v hernom priemysle, konkrétne u herných konzol Kinect. Zariadenie vypočíta hĺbku scény na základe premietanej matice bodov v infračervenom spektre. Body sú premietané projektorom umiestneným vedľa

infračervenej kamery a sú premietané v známom tvare a počte. Kamera odrazené body následne sníma a softvér vyhodnotí hĺbku obrazu na základe posunu bodov a vzdialenosti susedov. Jednoducho povedané, čím viac sú body od seba navzájom vzdialené, tým sa odrazili od objektu, ktorý je ďalej (viď Obrázok 2). Tento postup pri aplikácii produktu Microsoft Kinect V2 nám zaručil výsledok vo forme matice bodov o rozmere 512 * 424 a informácii o hĺbke v intervale od 0,5 m do 8,0 m a presnosťou na milimetre. Maticu bodov vidíme na Obrázku 5.

Výhody:

- presná rekonštrukcia komplexnej scény.

Nevýhody:

- infračervené body sú ľahko rušené priamym slnečným žiarením.

Aj keď je táto technológia vhodná primárne na vnútorné použitie, kvôli interferencii zo strany priameho slnečného žiarenia, Microsoft aplikoval do svojho produktu Kinect V2 funkciu takzvaných „aktívnych“ infračervených lúčov, teda projektor vysiela tieto infračervené body v konkrétnej frekvencii blikania, ktorý je následne ľahšie odlíšiteľný od ostatného šumu. Produkt sme samozrejme testovali v reálnej prevádzke vo vonkajšom prostredí, výsledky boli nadmieru dobré, takže výsledkom nášho porovnávajúceho testu je technológia sledovania infračervených bodov a konkrétne aplikovaná na zariadení Microsoft Kinect V2. Výstup z tejto hĺbkovej kamery je znázornený na Obrázku 7.

Aj keď sa výber technológie spracovania infračervených bodov mohol zdať ako vhodný pre vývoj v tejto fázy projektu, pre uistenie sa v tvrdeniach sme pristúpili k porovnávaniu medzi najpoužívanejšími typmi hĺbkových kamier. Tieto varianty využívajú či už spomenuté infračervené body, ale aj technológiu stereoskopie. Porovnávacia Tabuľka 1.1. sa nachádza v Prílohe, červenou sme zvýraznili kritériá vo variantoch, ktoré sú vyslovene neakceptovateľné pre potreby tohto projektu, zelenou je zvýraznený víťazný kandidát.

Pre lepšiu predstavu, je potrebné ešte uviesť, kde všade sa podobné technológie skenovania okolia alebo objektov pomocou hĺbkových kamier využívajú. S využitím podobných technológií som sa stretol najmä pri výskume spojeným s touto prácou, neskôr pri testovaní a vyvíjaní projektu navigácie pre nevidiacich. Konkrétne technológiu infračervených bodov tiež využíva firma Apple vo svojich iPhoneoch od modelu X, na odomykanie pomocou tváre, tiež známe ako Face ID. Tie totiž okrem známych kontúr tváre dokážu nasnímať aj topológiu

tváre používateľa, čo vedie k vyššej presnosti zabezpečenia samotného mobilného telefónu. Skenovanie a rekonštrukcia 3D scén našla svoje využitie aj pri veľkom rozmachu autonómnych vozidiel, ktoré v dnešnej dobe zažívajú obrovský úspech. Tieto vozidlá potrebujú pre presnú mikro navigáciu pri svojej ceste poznať svoje bezprostredné okolie podobne ako vodič vozidla sleduje svoje zorné pole a prispôsobuje tak jazdu. Avšak s hĺbkovou kamerou nemusíme nutne rekonštruovať len okolitú scénu, ale aj objekty. Tento princíp našiel uplatnenie v priemysle využívajúcom 3D tlačiarne na produkciu atypických typov objektov. Existujúci objekt sa naskenuje pomocou vyššie spomenutých technológií, môže sa ešte dodatočne upraviť pomocou vhodného softvéru a neskôr sa zreplikuje využitím 3D tlačiarne. Jedným z ďalších využití sú napríklad možnosti rozšírenej reality, využívané pri simuláciách v priemysle, alebo pre zábavu u konečných spotrebiteľov. Je totižto oveľa presnejšie rozšíriť realitu o dodatočné objekty, pokiaľ sme okolitú scénu naskenovali hĺbkovou kamerou a poznáme tak umiestnenie reálnych objektov.

3.2.5. Procesorová jednotka

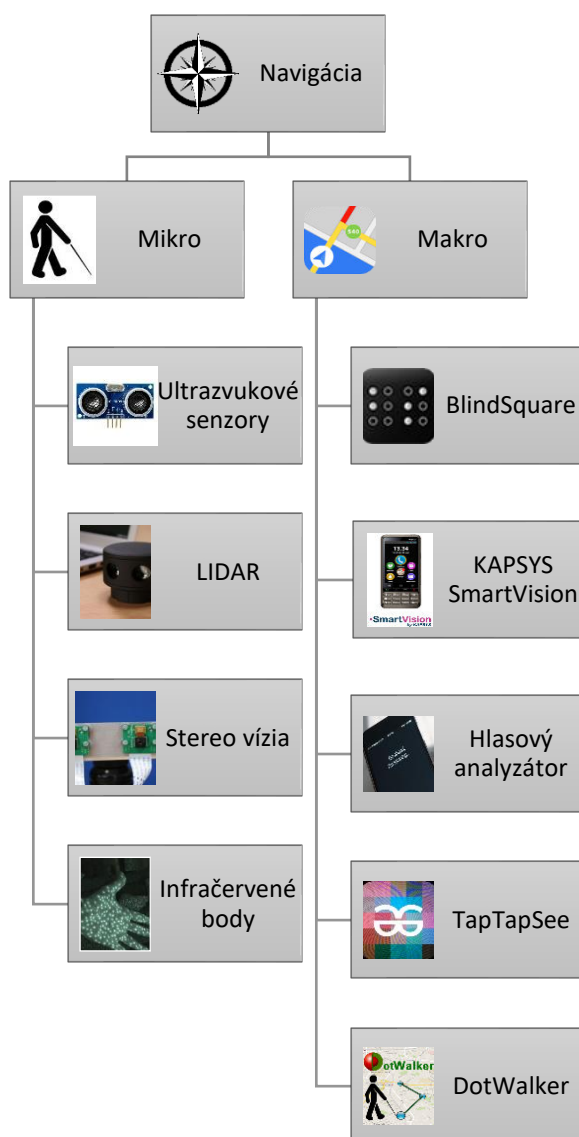
Výstup z každého jedného typu senzoru, opísaného v predošlej podkapitole je len súbor surových údajov a dát, s minimálnym spracovaním. Jedným z výstupov a pridaných hodnôt tejto práce je premena informácie, ktorú nevidiaci a slabozrakí ľudia nedokážu prijať a spracovať, teda vizuálna informácia o stave a povahe okolia, na takú formu, ktorú dokážu prijať a pochopiť. Na túto úlohu musíme ako medzikus použiť prenosný počítač, ktorý bude informácie spracovávať a vhodne transformovať. Za týmto účelom sme museli najprv vhodného kandidáta vybrať spomedzi dostupných a najpoužívanejších mini počítačov. Medzi kritériá sme zaradili nákupnú cenu, rozmery, CPU daného počítača, odber a vstupno-výstupné porty. Výsledky porovnávaní sa nachádzajú v Tabuľke 1.2..

Víťazom porovnávaní sa stal Mini PC – Latte Panda (viď Obrázok 8), a to z hlavných dvoch dôvodov. Prvý, je vybavený USB 3.0 portom, pre vysokorýchlostnú komunikáciu s Kinectom a druhý, pracuje s CPU od Intelu, čo nám dovoľuje spustiť na danom PC plnohodnotnú kópiu Windows 10 a tým nestrácať čas problémami s kompatibilitou medzi Microsoft Kinect SDK (popísané v ďalšej podkapitole) a operačným systémom. Objednali sme priamo od výrobcu za 150 USD a po pár dňoch bolo doručené. Výrobca umožňuje dokúpiť

rovnako aj „Starter Kit“, teda napájacie zdroje, krabičku a sústavu chladičov s aktívnym chladením pre bezproblémový chod PC.

3.3. Porovnávací diagram

Pre lepšiu ilustráciu prikľadáme diagram zobrazujúci využitie mikro a makro navigácie za pomoci jednotlivých technológií a softvérov.



3.4. Výber softvérových prostriedkov

V predchádzajúcej kapitole sme preberali vhodné hardvérové prostriedky, alebo konkrétne senzory a ich sústavu, ktoré nám pomôžu skenovať scénu a okolie používateľa. V tejto kapitole sa budeme sústrediť na softvér, ktorý sme pre potreby tejto práce

naprogramovali a slúži na obsluhu hardvéru a výpočet hrozieb kolízie používateľa a okolitých objektov pri jeho navigácii. Ako programovací jazyk bol vybraný jazyk C++ z dôvodu jeho rýchlosti, ale aj priamej podpore využitia C++ SDK pre Kinect, rovnako aj množstvo knižníc pre spracovanie matíc a obrazu. Našou úlohou v tomto prípade bolo vytvorenie algoritmu, ktorý nájde možnú hrozbu v získanom hĺbkovom obraze z Kinectu. Algoritmus musí zanedbať malý zhuk pixelov s nízkou vzdialenosťou, pretože s najväčšou pravdepodobnosťou by išlo v tomto prípade o informačný šum, avšak musí byť schopný aktuálne hrozby nevynechávať a správne lokalizovať. Aby sme našli tento optimálny bod, začali sme s testovaním rôznych porovnávacích algoritmov. Tieto algoritmy opíšeme v nasledujúcej kapitole ako poslednú podkapitolu pred opísaním výsledkov práce.

Microsoft Kinect SDK

3.4.1. Samotný Kinect potreboval ku svojej správnej funkcii SDK, teda jednoducho povedané súbor funkcií, ktoré zjednodušia prácu s týmto produktom a hlavne nebudeme nútení si všetky funkcie programovať sami od začiatku, ale samotní vývojári z Microsoftu nám poskytnú tento nástroj. Obsahuje množstvo funkcií, keďže sa jedná primárne o doplnok ku hernej konzole, na sledovanie ľudského tela a jeho častí, my sme však pre potreby tohto projektu využívali len súbory funkcií na čítanie hĺbkových matíc a ich vlastností. Po zapracovaní SDK do projektu sme sa mohli pustiť do vyvíjania porovnávacích algoritmov.

3.4.2.

Maximálna hodnota

Ako prvý sme skúsili algoritmus, ktorý nájde zhuk bodov s maximálnou hodnotou, teda mal by byť schopný nájsť hoci aj malý objekt, pokiaľ sa dostane veľmi blízko. V praxi by to znamenalo, že by sme porovnávali body s vysokou hodnotou so susednými, a pokiaľ by tieto tiež nadobúdali podobné hodnoty, znamenalo by to blízky objekt v našej scéne. Tento postup sa však neosvedčil, keďže sme zistili, že niektoré objekty nadobúdajú tvary, pri ktorých nie sú susedné pixely z pohľadu hĺbkovej mapy príbuzné.

Vysoký exponent

Vyskúšali sme postup zvýhodňovania bodov s vysokou hodnotou a to konkrétne exponovaním vysokým exponentom. V tomto prípade sme mali v úmysle zväčšiť rozdiely

medzi vzdialenosťami, pre jednoduchšie následné vyhodnocovanie hrozieb, avšak neboli sme schopní odlíšiť rušenie infračervených bodov cudzími zdrojmi od naozajstných hrozieb.

Intervalový histogram

Ako optimálne riešenie sa ukázalo rozdeliť si obraz do sektorov a v rámci každého sektora si vytvoriť intervalový histogram hodnôt jednotlivých pixelov. Pokiaľ histogram reprezentujúci body bližšie ako jeden meter, bol dostatočne vysoký, algoritmus označil tento sektor ako smer možného nebezpečenstva. Tento postup nám rovnako zabezpečil aj možnosť proporcionálneho rozdelenia stúpajúcej hrozby z jednotlivých sektorov, teda nemuseli sme sektory označovať binárnou logikou, ale mohli sme sledovať stúpanie a klesanie jednotlivých hrozieb.

Na záver tejto podkapitoly môžeme zhrnúť naše poznatky z výberu technických a programových prostriedkov do jasného výberu vhodného kandidáta na nasledujúcu prácu, testovanie a porovnávanie výsledkov.

4. Výsledky práce

4.1. Vývoj softvéru a testovacieho prostredia

Po výbere vhodných technických a programových prostriedkov sme sa mohli pustiť do vývoja a aplikácie poznatkov. Za programovací jazyk sme si vybrali C++, vysokú podporu od Kinect SDK. Prvá úloha bolo správne inicializovať Kinect a spustiť komunikáciu medzi PC a Kinectom. Na toto nám poslúžilo SDK od Microsoftu, aby sme nemuseli bádať ako toto vyriešiť, tzv. od nuly, ale využili sme teda zabudované funkcie pre správnu inicializáciu komunikácie Kinectu. Keď bol Kinect spustený a komunikácia fungovala správne, začali sme si pýtať od Kinectu snímky hĺbkovej mapy, ako surové dáta určené na spracovanie. Tento postup je zobrazený v prílohe so zdrojovými kódmi 1.5 a na jeho konci máme v pamäti počítača uloženú hĺbkovú mapu s rozmerom 512 * 424, 24 krát za sekundu v rozsahu od 500 do 8 000 milimetrov.

4.1.1. Vizualizácia údajov v matici

Po kontrole správnosti, či je hĺbková mapa správna som narazil na problém. V pamäti sa síce nachádzala matica s údajmi, avšak tento údaj je pre správnu kontrolu nedostačujúci. Pre človeka je najlepší typ kontroly stále vizualizácia, preto sme sa rozhodli túto mapu vizualizovať. Na to nám perfektne poslúžila knižnica OpenCV, čo je knižnica tried funkcií a metód, určených pre spracovávanie a analýzu obrazu. „CV“ v názve knižnice znamená „Computer Vision“, teda ide o knižnicu primárne určenú pre počítačové videnie, čiže zameranie funkcie perfektne sedí pre naše potreby. Po zavolaní knižnice a vykreslení matice na obrazovku sme mohli teda odkontrolovať správnosť výsledku a kvalitu hĺbkovej mapy generovanej Kinectom. V tomto štádiu môžeme teda uzavrieť túto podkapitolu, s vedomím, že dokážeme inicializovať Kinect pomocou SDK od Microsoftu, vypýtať si hĺbkovú mapu od daného zariadenia a skontrolovať správnosť údajov v hĺbkovej mape vizualizovaním danej matice na obrazovke. Ďalším krokom bude teda práca s danou maticou, jej uloženie na disk a kompresia.

Ukladanie hĺbkovej mapy na disk

V predošlej podkapitole sme úspešne zvládli prácu s Kinectom a výstupom je kvalitné skenovanie scény okolia s presnosťou na milimetre. Avšak predtým, ako sa pustíme do vyhodnocovania hĺbkových máp za cieľom hľadania prekážok v ceste nevidiaceho, je nutné ^{4.1.2} nastaviť testovacie prostredie pre správne vyhodnotenie úspešnosti algoritmov, ktoré majú za úlohu tieto hrozby lokalizovať. Aby boli testy relevantné, podmienky medzi jednotlivými testami musia byť rovnaké, pričom jediná zmena medzi testami musí byť samotný algoritmus, ktorý chceme otestovať. Za týmto účelom teda musíme údaje o hĺbke naskenované z okolia uložiť vo forme sekvencií na disk a potom na týchto sekvenciách spúšťať testy. Spomenuté sekvencie si môžeme jednoducho predstaviť ako videá s nahrávkami okolia, pričom namiesto farieb je vizualizovaná hĺbka danej scény, podobne ako sme to robili v predošlej podkapitole za účelom kontroly správnosti práce Kinectu.

Za týmto účelom sme vyvinuli zapisovanie údajov odoslaných z Kinectu na disk. Matica sa zapisovala štandardným spôsobom, bez akýchkoľvek obštrukcií, ako celistvý blok $512 * 424$ pixelov. Keďže jeden pixel obsahoval údaje o vzdialenosti v milimetroch a maximálny dosah Kinectu je 8 000 milimetrov, nemohli sme jeden pixel reprezentovať len jedným bajtom, pretože $2^8 = 1\ 024$, teda maximálna hodnota, ktorá môže byť zapísaná do bloku jedného bajtu je 1 024. Pokiaľ využijeme dva bajty, teda $2^{16} = 65\ 536$ čo reprezentuje maximálnu hodnotu, ktorá sa zmestí do dvoch bajtov, teda 16 bitov. Jednotlivé matice údajov teda boli zapisované v blokoch $512 * 424 * 2$ bajty. V tomto bode sme ale narazili na problém. Pokiaľ vezmeme do úvahy, že Kinect je schopný novú hĺbkovú maticu vyprodukovať 24-krát za sekundu, dosiahneme rýchlosť zapisovania na disk popísanú nižšie.

$$\frac{512 * 424 * 2 * 24}{1024^2} = 9,9375\ MB * s^{-1}$$

Tento dátový tok je síce plne zvládnuteľný dnešnými technológiami, no pri predstave, že pri nahrávaní sekvencií budeme plniť disk rýchlosťou takmer 10 MB/s, sme sa radšej zamysleli nad možnosťami kompresie týchto dát. Pri skúmaní ako vyzerajú tieto matice v surovom stave, sme pristúpili na „RLE“ dátovú kompresiu.

RLE je veľmi jednoduchá, bezstratová dátová kompresia, ktorá beží na dátach, ktoré sú charakteristické tým, že obsahujú veľké množstvo rovnakých údajov v nepretržitých a celistvých sekvenciách a je výhodnejšie uložiť ich reprezentáciu a počet, ako pôvodný reťazec. Najčastejšie použitie nájdeme napríklad v grafike, ikonách, animáciách a obrázkoch. Čiže na disk sme začali miesto surových matíc zapisovať zreťazené sekvencie opakujúcich sa znakov, formou: počet znakov, typ znaku. Týmto štýlom sme dosiahli zlomkovú veľkosť dát na disku oproti pôvodnému typu zápisu a to hlavne pokiaľ sme nahrávky robili vo vonkajšom prostredí, kde sa veľmi veľa krát opakovala nulová hodnota, teda pixely boli príliš ďaleko a vzdialenosť nebolo možné zmerať. Pripúšťame však možnosť, že týmto štýlom by bolo možné aj nechtiac zväčšiť objem dát zapisovaných na disk, a to v prípade, že každý pixel by mal unikátnu hodnotu, oproti svojim susedom. V tom prípade by každý pixel reprezentoval o 4 bajty (veľkosť premennej typu integer na 64-bitových platformách) viac dát. Kompresia má teda zmysel, len pokiaľ má sekvencia v matici aspoň 3 rovnaké znaky za sebou. Praktickým testovaním sme však prišli k záveru, že kompresia je výhodná a dáta na disku zaberať oveľa menej miesta ako v prípade zápisu v surovom stave. Zložitosť algoritmu kompresie a dekompresie je lineárna, keďže všetky operácie zvládneme vykonať jedným prechodom cez dátovú štruktúru. Úryvok zo zdrojového kódu sa nachádza v kódovej Prílohe 1.4. Na konci tejto podkapitoly sme schopní zaznamenávať výstup z Kinectu a zapisovať tieto dáta na disk po kompresii, ktoré takto zaberať oveľa menej miesta. V nasledujúcej kapitole budeme ťažiť z výhod zapisovania na disk a vytvoríme testovacie prostredie pre vyhodnocovanie úspešnosti algoritmov, ktoré majú za úlohu nájsť a lokalizovať hrozby kolízie v týchto hĺbkových mapách.

4.1.3.

Testovacie prostredie

Po vhodnom zápise reprezentácie matice na disk, sme mohli začať s vytváraním testovacieho prostredia. V ňom budeme testovať úspešnosť algoritmov hľadajúcich a lokalizujúcich hrozby v hĺbkových mapách okolia a porovnávať ich s absolútnou pravdou, ktorú sme vytvorili z ručne zapísaných anotácií v súbore zvlášť. Tieto údaje sme získali sledovaním nahrávok sekvencií ako výsledok predošlej podkapitoly. Absolútna pravda v tomto anotačnom súbore je reprezentovaná ako interval snímok a sektor, v ktorom je nebezpečenstvo odstupňované stupňami 1 až 4.

Testovacie prostredie teda funguje nasledovne:

- program načíta súbor s uloženou absolútnou pravdou, ktorý sme dopredu pripravili,
- následne otvorí príslušný súbor s uloženou sekvenciou hĺbkových máp, ktorý sme dopredu nahrali,
- potom pokračuje a načíta jednu snímku za druhou, vyhodnotí hrozby pomocou všetkých vyhodnocovacích algoritmov a výsledky porovná s absolútnou pravdou uloženou v anotačnom súbore, a to tak, že s využitím poznatku, v ktorom snímku sa práve nachádzame, nájde príslušný interval v súbore a k nemu príslušné sektory matice, v ktorých by sa príslušné nebezpečenstvo malo nachádzať.
- výsledky porovnávania si uloží do pamäte a po skončení nahranej sekvencie vytvorí analýzu úspešnosti podľa príslušného algoritmu.

Vyhodnocovacie algoritmy sme popisovali v kapitole 3.3., časti kódu sa nachádzajú v prílohe so zdrojovými kódmi 1.1., 1.2., 1.3.. Na základe údajov z tohto testovacieho prostredia bol víťazom algoritmus využívajúci intervalový histogram ako jadro svojho fungovania. Tento algoritmus bol schopný nájsť hrozby s úspešnosťou obecné >90%, čím sme teda mohli dokázať fungovanie algoritmus aj konkrétnymi a absolútnymi výsledkami a nie len pocitom, že algoritmus funguje správne. Do budúca toto prostredie poskytuje skvelý potenciál pre ďalší vývoj z hľadiska jednoduchej kontroly nového prístupu k vyhodnoteniu a lokalizácie hrozieb. Nový prístup totiž budeme schopný okamžite skontrolovať, prípadne do anotačného súboru pridať viacero sledovaných aspektov, ako len blížiac sa hrozby, ale aj blížiac sa nerovnosti povrchu v podobe obrubníkov a jám, ale aj rôzne iné informácie extrahovateľné z hĺbkovej mapy, prípadne kombinácie hĺbkovej a RGB mapy. Testovacie prostredie vyvíjajú všetky podobné rozsiahle projekty pre urýchlenie celkového vývoja a obrovským rozšírením možností do budúca.

Výsledkom práce popísanej v tejto podkapitole je výber vhodného algoritmu schopného prečítať hĺbkovú mapu okolia z Kinectu, vyhodnotiť a lokalizovať potenciálne hrozby kolízie s objektami a túto potenciálnu hrozbu reprezentovať intenzitou (Obrázok 10). Ďalším krokom bude túto získanú informáciu transformovať a reprezentovať formou, ktorú

bude mať možnosť nevidiaci a slabozrakí používateľ systému pochopiť a využiť na svoju navigáciu.

Text-to-Speech modul

Predtým ako sme sa mohli posunúť k testovaniu v reálnych podmienkach, zostávalo vyriešiť zdanlivo posledný problém. Tým je vhodná reprezentácia výstupných informácií. Ako prvý nápad prišiel reprezentovať tieto informácie zvukom a ako vhodná štruktúra zvukových informácií mohol poslúžiť prirodzený ľudský jazyk. Do programu sme preto implementovali TTS (Text-to-Speech) modul od spoločnosti Microsoft. Tento modul pracuje tak, že vstupný reťazec znakov je transformovaný do hovorovej reči na výstupe daného modulu. Použitie v našom projekte je veľmi jednoduché, vývojári z Microsoftu modul pripravili výborným spôsobom, stačí ho pridať do projektu a po skonštruovaní tried len nakrmiť metódy želanými reťazcami. Vstupné reťazce sme pripravili tak, že na začiatok sa priradí horizontálny smer nebezpečenstva (vpravo, vľavo, v strede), potom vertikálny smer (hore, dole, v strede) a na koniec intenzita nebezpečenstva (bližšie ako 2,5 metra, bližšie ako 1,5 metra, bližšie ako 1 meter). Po krátkom otestovaní, sme zistili, že rozčlenenie priestoru pred používateľom bude musieť byť na oveľa menšie sektory ako 3×3 , pretože dané sektory sú jednoducho príliš veľké. Scénu sme teda rozdelili na 8 horizontálnych a 4 vertikálne sektory, čím sme dostali 32 samostatných sektorov.

Na konci tejto podkapitoly sa náš projekt nachádza v bode, kde dokážeme zrozumiteľnou rečou varovať používateľa pred nebezpečenstvom a hrozbou kolízie s objektami v jeho bezprostrednom okolí, ktoré boli zistené použitím algoritmov popísaných v kapitole 3.3.4., pričom nebezpečenstvo je odstupňované a varovania sú prioritizované na základe vzdialenosti, ale aj smeru zrážky (čelné majú vyššiu prioritu). V tomto štádiu je našou úlohou prejsť do testovania v reálnej prevádzke a zistiť ako sa tento mikro navigačný systém bude chovať mimo vývojového laboratória.

4.2. Testovanie navigácie

V nasledujúcej podkapitole máme naše zariadenie pripravené, softvér funkčný a rovnako pripravený na testovaciu fázu nášho projektu. Táto fáza má za úlohu odhaliť okrem dodatočných nedostatkov, ktoré sme počas predošlých fáz zanedbali, aj kvalitu navigácie a celkový prínos, ktorý tento projekt prináša.

Zariadenia však bolo potrebné pred spustením testovania v reálnej prevádzke mierne prispôbiť. V prvom rade, sme museli vyriešiť problém so zdrojom napájania, ktorý musel byť prenosný. Pribalený adaptér ku Kinectu mal výstupných 12 voltov, čo bolo operatívne napätie, pri ktorom Kinect pracuje. Za svoj zdroj energie sme teda vybrali olovený batériový článok z motocykla, čo nám poskytlo dostatočnú kapacitu na niekoľko hodín testovania a zvládnutelnú hmotnosť pri nosení v rucksaku. Hmotnosť akumulátora sa pohybovala na úrovni 5 kg, čo nebola najpríjemnejšia hodnota, avšak pre testovacie účely išlo o veľmi nepodstatnú nepríjemnosť a batéria fungovala spoľahlivo počas všetkých fáz testovania, aj pri mínusových teplotách v decembrovom počasí, keď sme testovali vonku. Kabeláž Kinectu sme preto prespájkovali na konektory pripojiteľné k motocyklovému akumulátoru a v prípade nechceného zlého zapojenia pridali aj poistku, ktorá mala slúžiť ako ochrana pred prepólovaním.

Ako počítač, ktorý mal za úlohu spracovať výsledky z Kinectu sme zvolili Latte Panda (výber a popis PC spracovaný v kapitole 3.2.5. a odfotený na Obrázku 8). Na tento počítač sme nainštalovali kópiu operačného systému Windows 10 a náš softvér vo forme projektu v programe Visual Studio. Napájanie PC bolo realizované tiež z batérie a bolo potrebné medzi túto batériu a Kinect vložiť DC-DC menič, z dôvodu zníženia napätia z 12 voltov na 5 voltov, čo je operačné napätie Latte Panda, keďže je napájaný pomocou Micro USB konektora.

Všetok hardvér je v tomto bode pripravený na testovanie a uložený v rucksaku testera, softvér bez problémov beží a sme pripravený na prvé fázy testovania.

4.2.1.

Testovanie vnútri

Ako prvé sme sa pustili do testovania vo vnútorných podmienkach (viď Obrázok 6). Testovacieho pracovníka sme vybavili hardvérom a pripravili testovaciu prekážkovú dráhu, ako prvú vo forme úzkej chodby plnej objektov, s ktorými hrozí potenciálne nebezpečenstvo.

Cieľom tohto testovania bolo zistiť správne umiestnenie a natočenie Kinectu na tele používateľa, odstupňovania hrozieb čo najintuitívnejším spôsobom ale aj samozrejme prvotné odskúšanie celého systému skenovania okolia pomocou technológie infračervených bodov.

Po krátkom testovaní sa dostavili prvé zmysluplné výsledky, zistili sme, že umiestnenie Kinectu bude najvhodnejšie v oblasti hrudníka a to z dôvodu lepšej detekcie objektov na úrovni zeme. Kinect si vo vnútornom prostredí počínal veľmi dobre, testovací

pracovník po chvíli tréningu bol schopný aktívnemu vyhýbaniu sa prekážkam, bol schopný lokalizovať potenciálnej hrozbe vo svojej ceste a prispôsobovať si tak svoju trajektóriu. Zvolili sme preto komplikovanejšie a viac „neurčité“ prostredie, teda väčšiu miestnosť s objektami, čím sme výrazne sťažili priestorovú orientáciu testovacieho pracovníka. V tomto prípade už navigácia nebola taká elegantná ako pri prechádzke rovnou chodbou, avšak stále bol pracovník schopný svoju cestu neskřížiť s akýmkoľvek objektom.

Po úspešnom zvládnutí testovania vnútri budov, sme zhodnotili, že Kinect je vhodným pomocníkom, pri správnom umiestnení na tele používateľa, dosah 8 metrov je dostačujúci a náš softvér hrozby vyhodnocuje správne. Ďalšou fázou teda bude testovanie vo vonkajších podmienkach pri reálnej prevádzke medzi ľuďmi.

Testovanie vonku

4.2.2. V tejto podkapitole sa budeme zameriavať na popis udalostí počas testovania nášho systému na navigáciu pre nevidiacich vo vonkajšom prostredí. Výsledky z testovania vnútri boli veľmi dobré, preto sme sa rozhodli čo najskôr presunúť von. Zostávalo vybrať už len vhodné testovacie prostredie a keďže sme chceli v tejto fáze zachytiť, pokiaľ je to možné, čo najviac scenárov, presunuli sme sa do centra mesta, konkrétne Bratislavské Vianočné trhy. Vo vnútorných podmienkach bolo použitie infračervených bodov k rekonštrukcii 3D scén naozaj skvelým riešením.

Po presune na miesto testovania sme testovacieho pracovníka vybavili všetkým potrebným hardvérom a systém zapli (viď Obrázok 9). Uloženie káblov nekoordinovane po tele testovacích pracovníkov, ako aj pochybný obsah batohu na chrbte viedol k častým nedôverčivým pohľadom okoloidúcich, pre vtedajšiu vypätú situáciu na scéne európskych vianočných trhov sme preto experimenty ukončili pred intervenciou štátnych bezpečnostných zložiek. Záver z tohto experimentu však len potvrdil naše očakávania, aktívna forma infračervených bodov veľmi dobre filtruje rušenia vonkajších vplyvov, avšak nedokáže si poradiť so slnečným žiarením. Slnko je a ešte dlho bude, veľkým súperom pre väčšinu senzorov. Problém nastával vo veľkom množstve objektov, ktoré sa v zábere nachádzali, filtráciu relevantných informácií sme pripísali na zoznam vylepšení v budúcnosti.

V tomto bode sme dokázali funkčnosť nami navrhnutého a vyvinutého systému pre navigáciu nevidiacich aj v reálnych podmienkach, počas zvýšenej koncentrácie ľudí na tak masovom podujatí, akými sú Vianočné Trhy v Bratislave (viď Obrázok 10).

4.3. Výsledky výskumu

Výsledky a zistenia výskumu boli nadmieru príjemné. Zistili sme, že využitie infračervených bodov za účelom rekonštrukcie 3D scén je výborný nápad, aj keď aj napriek všetkým snahám vývojárov od Microsoftu je stále veľkým problémom priame slnečné žiarenie, hlavne počas jasných slnečných dní s vysokými teplotami. Pre analýzu obrazu sme integrovali ešte „Text to Speech“ modul, z dôvodu využitia najmä slabozrakými a nevidiacimi používateľmi. Tento si počínal veľmi dobre a varoval používateľa o smere a intenzite hrozby kolízie. V tejto kapitole by sme chceli zhrnúť aj všetky ciele zadefinované v kapitole „2. Cieľ práce“ a zhodnotiť ich plné alebo čiastkové splnenie.

Analýza existujúcich technických a programových prostriedkov

4.3.1.

Tejto problematike sme sa intenzívne venovali hlavne v kapitole „1. Analýza existujúcich navigačných riešení“, kde sme si zadefinovali rozdiel pojmov mikro a makro navigácia a ich využitie v našej problematike. Rozpísali sme technické prostriedky používané na makro navigáciu, ich historický vývoj a aplikovanie do praxe v súčasnosti. Analýza technických prostriedkov pokračovala v kapitole 3.1, kde sme porovnali najpoužívanejšie formy skenovania scén pomocou technologických pomôcok, a v kapitole 3.2 sme spracovali softvérové a programové nástroje nutné k spracovaniu výstupov a signálov z týchto zariadení.

4.3.2.

Metodológia postupu využívania možného prostriedku

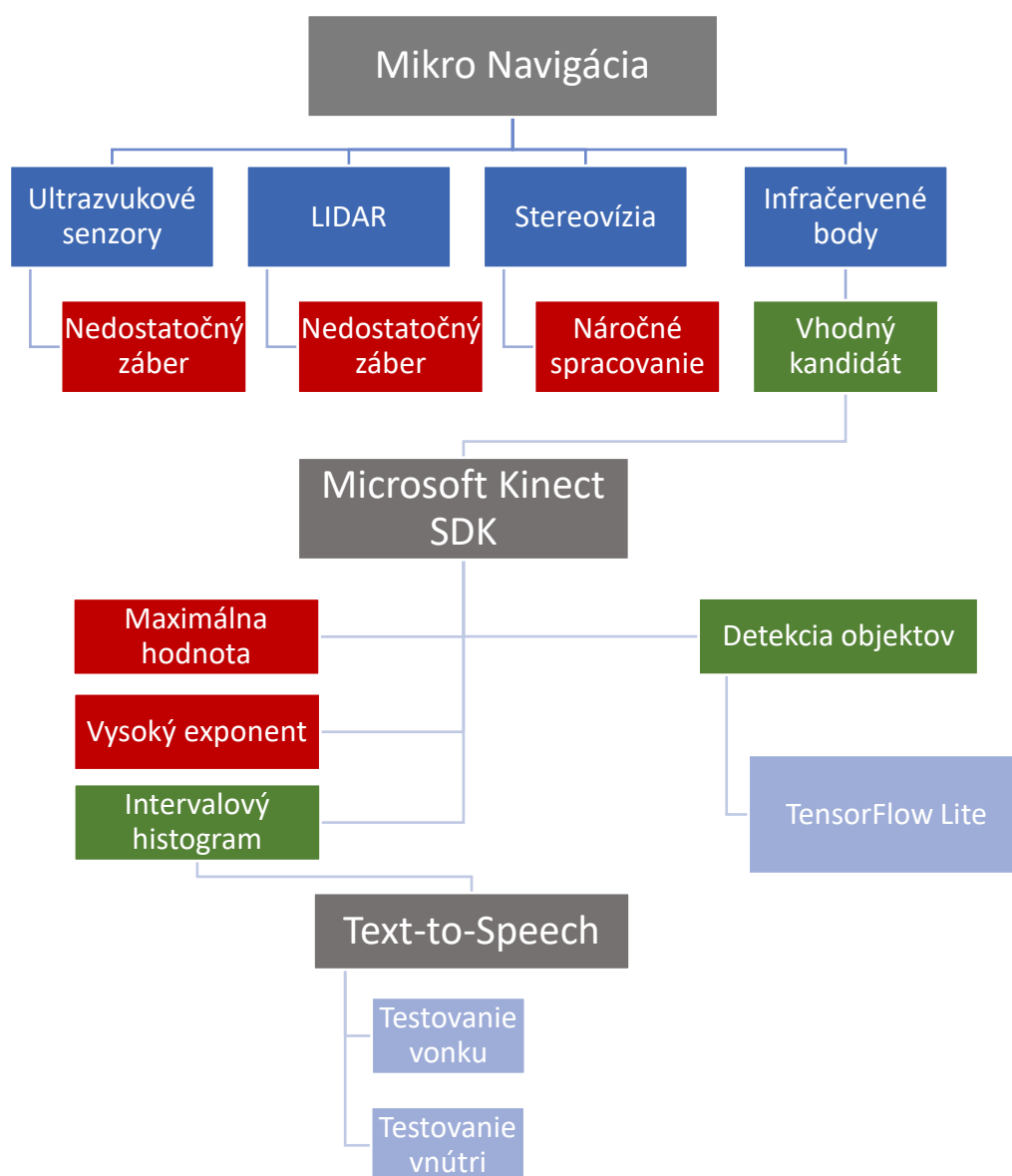
Postupy používané pri využívaní vybraných technických pomôcok mohli byť vytvorené až na základe dôkladnej analýzy vybraného technického prostriedku. Tieto metodológie a postupy sme si teda nemohli zadefinovať len po úvodnom výskume a porovnaní kandidátov na vhodné zariadenie, ale až po testovaní v reálnych podmienkach. Na základe týchto reálnych poznatkov sme boli schopní postupne vylepšovať úvodný návrh zapojenia, programovania a využitia zariadenia pre čo najoptimálnejšie výsledky.

Výber technických prostriedkov pre realizáciu

Porovnaniu a výberu hardvéru vhodnému k realizácii výsledného produktu sme osobitný rozbor v kapitole 3.1., kde sme si podrobne rozobrali princíp fungovania jednotlivých senzorov vhodných k skenovaniu a rekonštrukcii okolitých scén, popísali ich^{4.3.3} možné výhody a nevýhody v danej problematike a aplikovaní použitia a vybrali z pomedzi kandidátov jedného optimálneho. Tým sa stal produkt Microsoft Kinect V2, ktorý sa stal aj predmetom testovania a nasadzovania v reálnej prevádzke (viď Obrázok 4) pre potreby získania poznatkov o skutočnej aplikácii zariadenia ako takého ale aj testovania technológie.

4.4. Harmonogram

Pre lepšiu predstavu prikkladáme harmonogram a zhrnutie vykonanej práce v prehľadnejšej grafike.



5. Vízia projektu

Na záver praktickej časti by sme chceli zhrnúť zopár úvah o budúcnosti a ďalších možnostiach rozvoja tohto projektu. Úvahy sme vykonali na základe podrobnej analýzy a dlhodobej diskusie výsledkov výskumu a vývoja v tejto oblasti, ako aj pri práci v podobných projektoch so zameraním na počítačové videnie. Vízia sme rozdelili na viacero podkapitol, podľa oblasti zlepšenia jednotlivých segmentov projektu.

5.1. Skenovania okolia

Do budúcnosti plánujeme výskum v oblasti vyššie spomínanej stereoskopie, teda použitie množiny RGB kamier k rekonštrukcii scén. Ide o efektívnejšie riešenie, menej náchylné na rušenia vonkajšími vplyvmi oproti momentálne využíwanej technológii sledovania infračervených bodov. Svoju daň za výhody si vyberá vysoká výpočtová náročnosť pri použití tejto technológie, keďže zložitosť algoritmov použitých na výpočet hĺbkovej mapy z dvojice RGB obrázkov rastie exponenciálne s rastúcim rozlíšením daných obrázkov. Z tohto dôvodu nás momentálne obmedzuje technológia dnešnej doby, a na vhodné použitie tejto technológie musíme počkať s príchodom výkonnejších procesorových jednotiek.

Aj v tomto prípade prichádza technologický vývoj do štádia, kedy napodobňuje vývoj organických živočíchov v prírode, ktorý svoje genetické vlastnosti dostali vďaka výsledkom evolučných algoritmov.

5.2. Vyhodnocovacie algoritmy

Pre vyhodnotenie hrozby sme v našej práci používali algoritmy, ktoré vytvárali z hĺbkovej mapy istú heuristiku o nebezpečných sektoroch, teda smeroch blížiacich sa objektov. Tento typ vyhodnotenia bol veľmi rýchly a efektívny, čo nám dovoľovalo spustiť daný algoritmus aj na malých a energeticky nenáročných zariadeniach. Ide však o príliš jednoduchý princíp, ktorý nám neumožňuje získať dodatočné informácie o okolí používateľa.

V súčasnosti veľký posun môžu zaznamenať aj neurónové siete v oblasti identifikácie objektov. Pri použití neurónových sietí za účelom identifikácie a klasifikácie objektov hrozacej kolízie môžeme získavať informácie nielen o vzdialenosti daných objektov ale aj ich povahe. Tieto informácie vieme následne využívať nielen na mikro navigáciu a vyhýbanie sa prekážkam

ale aj ku nájdeniu samotného cieľa, prípadne zaujímavostí počas cesty a tým razantne zvýšiť kvalitu života nevidiacich používateľov systému.

V oblasti neurónových sietí sme robili výskum aj v rámci tohto projektu a trénovali sme vlastné modely neurónových sietí. Výsledky nás veľmi príjemne prekvapili, modely mali vysoké úspešnosti a po drobných optimalizáciách aj vysoký výkon na prenosných zariadeniach. Používali sme konkrétne nástroj pre trénovanie neurónových sietí od firmy Google, konkrétne TensorFlow a jeho mobilnú verziu TensorFlow Lite. Po nazbieraní dostatočne veľkého datasetu s anotáciami (dáta obsahovali ručne dopísané informácie o svojom obsahu a charaktere) sme tento dataset rozdelili na trénovací a testovací. Následne sme spustili tréning a evaluáciu daného modelu. Náročnosť tréningu bola veľmi vysoká, preto sme ho v rámci úspory prostriedkov a času presunuli na cloud, kde bežal na prenajatých špičkových grafických kartách. Aj na týchto kartách bolo trvanie tréningu často v desiatkach hodín, pričom veľa záleží na použitej architektúre neurónovej siete. Po dotrénovaní a dosiahnutí požadovanej úspešnosti na testovacom datasete sme model exportovali do mobilnej verzie TensorFlow Lite a testovali na mobilných zariadeniach. Model bol schopný kvalitne detekovať a klasifikovať objekty na základe informácie z kamery mobilného telefónu a rovnako bol schopný aj určiť pravdepodobnosť istoty detekcie.

Problém bol v platformovej kompatibilite daného nástroja pre využitie neurónových sietí TensorFlow, ktorý bol určený primárne na Linuxové operačné systémy. To nám zabránilo rýchlo a efektívne napojiť našu neurónovú sieť na náš projekt a previazať teda vyhodnocovanie hrozieb pomocou hĺbkovej kamery s ich klasifikáciou pomocou neurónovej siete. Tento platformový problém sme vyriešili ukladaním záznamu sekvencie z testovania na disk a opätovným spustením na inom operačnom systéme. Ako riešenie do praxe je to síce nevyhovujúce, ale v rámci tohto študentského projektu sme si perfektne osvojili prácu s umelou inteligenciou a neurónovými sieťami a dokázali funkčnosť riešenia v danej oblasti.

Do budúcnosti teda určite plánujeme v rámci rozvoja tohto projektu zakomponovať vlastný model neurónovej siete do vrstvy systému, ktorá vyhodnocuje surové dáta z kamier a senzorov na relevantné a zmysluplné informácie, ktoré dokážu skvalitniť navigáciu nevidiaceho a slabozrakého používateľa.

5.3. Vývoj zariadenia

Budúci vývoj rovnako dokáže zlepšiť obrovské medzery ukrývajúce sa v rozmeroch a energetickej náročnosti daných zariadení. V súčasnosti má Kinect V2 rozмеры cca 25 * 6,6 * 5,7 centimetrov, čo je ďaleko presahujúce rozмеры elegantného zariadenia, ktoré je užívateľ ochotný nosiť na svojom tele pri každodenných úlohách. Princíp technológie nie je komplikovaný. Vo svojej podstate ide o projektor infračervených bodov a infračervenú kameru. Tieto zariadenia nevyžadujú veľké rozмеры, dôkazom je aj použitie rovnakej technológie v hornej časti displeja iPhone X, na technológiu odomykania zariadenia tvárou známu ako „Face ID“. Kinect však svoje rozмеры nadobudol hlavne kvôli vyššej prioritě na chladenie, dlhodobý výkon a použitie doma, niekde na policičke, čo nevyžaduje tak vysoké nároky na rozмеры a energetickú náročnosť zariadenia. Podobný senzor u iPhone zase prišiel o dosah 8 metrov ako tomu je v prípade Kinectu. V našom projekte teda pri diskusii o budúcnosti vývoja musíme nájsť kompromis medzi týmito požiadavkami a dostať rozмеры zariadenia a aj energetickú náročnosť na optimálnejšiu úroveň.

5.4. Komunikácia s používateľom

Za najväčšiu výzvu v budúcnosti tohto projektu, však považujeme samotnú interakciu so slabozrakým používateľom tohto produktu. Hlasové ovládanie prináša množstvo zmätočných informácií, často redundantného a zbytočného charakteru. Ich filtrácia by nemusela odovzdávať používateľovi zase všetky relevantné informácie. Pri skúmaní riešení, sme však narazili na jeden zásadný objav na poli medicíny, konkrétne ide o zistenie, že nervové zakončenia na koreni jazyka, sú svojím spôsobom „paralelné“ s očnými nervami. Za predpokladu, že vidíme predovšetkým mozgom, nie očami, je možné stimulovať tieto nervy jednoduchou sústavou elektród umiestených na jazyku a tým vytvárať slabozrakému jednoduchý obraz priamo na mozgu. Testovania prebiehali zatiaľ na akademickej pôde v University of Pittsburgh Medical Center a dopadli výborne. Používateľ rozoznával konkrétne kontúry objektov, mohol bez použitia akéhokoľvek ďalšieho zmyslu priamo interagovať s týmito objektami a zvládol dokonca niektoré športové hry alebo jazdu na bicykli.

Záver

Možnosť práce práve na projektoch usilujúc sa o zlepšenie života ľuďom so zdravotným postihnutím, v spojení s možnosťou výskumu „počítačového videnia“, bola pre mňa veľkým lákadlom a výzvou, kvôli ktorej som sa do tohto projektu zapojil.

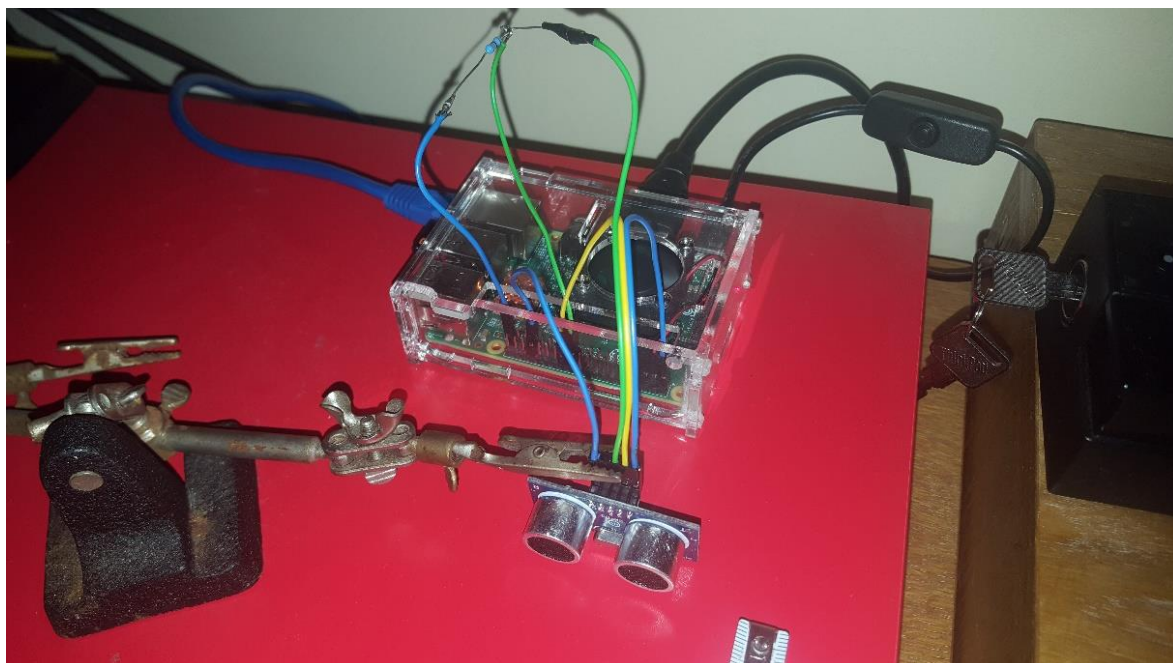
Cieľom tejto práce nebolo teda len teoretické opísanie možností, ktoré by nevidiaci a slabozrakí ľudia mohli vo svojom živote využívať, avšak aj reálna práca so samotnými technickými a programovými prostriedkami, za účelom skutočného testovania v reálnych podmienkach. Počas vývoja projektu sme si osvojili mnohé programovacie techniky ako napríklad prácu s externými knižnicami a SDK, a komunikácia s periférnymi zariadeniami. To by samozrejme nebolo možné bez zvládnutia programovacieho jazyka C++ na pokročilej úrovni.

Najväčší posun som však zaznamenal pri objavovaní a skúmaní techník „Computer Vision“ teda počítačového videnia. Práve v tejto oblasti vidím obrovský potenciál v budúcnosti, ako možnosť prispôsobenia Informačných Technológií do sveta ako ho poznáme, bez nutnosti jeho zmeny, napríklad práve v oblasti navigácie autonómnych vozidiel, či ako tomu bolo v tomto prípade zdravotníckych pomôcok.

Najpokročilejšie techniky počítačového videnia využívajú ako jadro niektorých svojich algoritmov práve neurónové siete, vďaka ich schopnosti veľmi klasifikovať dáta len na základe presne nešpecifikovanej podobnosti s inými, na ktorých sme tieto modely neurónových sietí vytrénovali. Ide o postup inšpirovaný prírodou a javmi nachádzajúcimi sa v prírode, kde živé tvory posudzujú objekty alebo situácie len na základe predošlých skúseností (prípadne vrodených inštinktov, ale to je na diskusiu v inej téme). Takýmito modelmi neurónových sietí sme sa na záver výskumu zaoberali a trénovali si svoje vlastné, čo nám dodalo cenné poznatky aj z oblasti dátového inžinierstva ale aj umelej inteligencie.

Celkovo hodnotíme tento študentský projekt na veľmi vysokej úrovni z hľadiska rozšírenia obzorov v mnohých segmentoch Informačných technológií, získali sme skúsenosti v mnohých odvetviach od elektrotechniky, softvérového inžinierstva, algebraické štruktúry a maticové operácie, dátové inžinierstvo až po umelej inteligencie. V budúcnosti má tento projekt teda vysoké uplatnenie nie len ako užitočná zdravotnícka pomôcka, ale aj ako študentský projekt slúžiaci úvodu do mnohých úrovní Informačných Technológií.

Obrázková príloha



Obrázok 1 - Zapojenie ultrazvukového senzora

```
*Python 2.7.9 Shell* pi@raspberrypi: ~
File Edit Shell Debug Options Windows Help

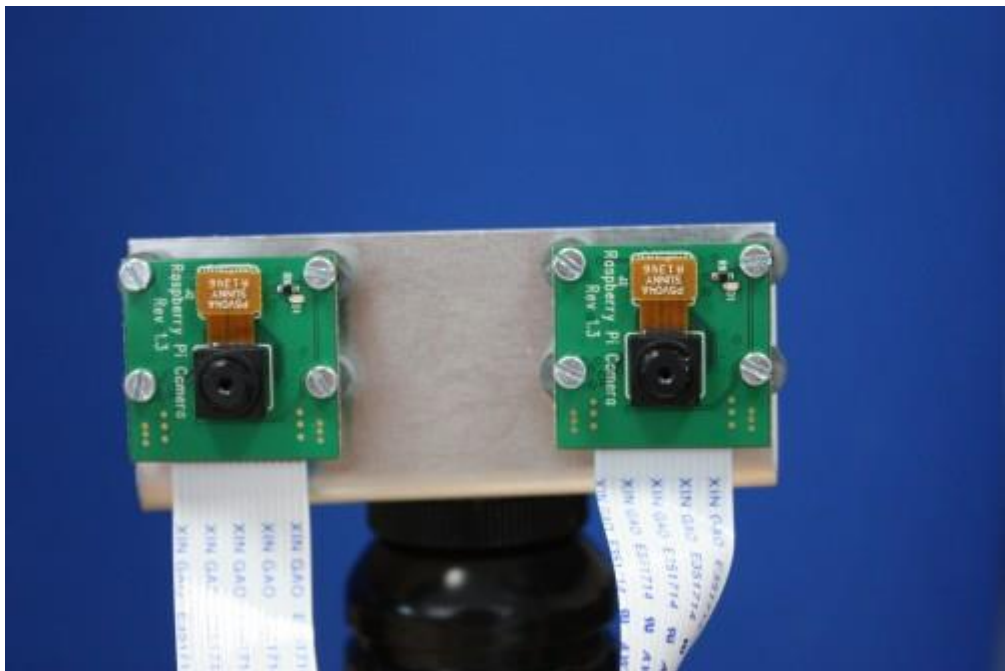
Cas = 0.00232217311859 ... tomu odpoveda vzdialenost : 39 cm
Cas = 0.00280253887177 ... tomu odpoveda vzdialenost : 47 cm
Cas = 0.0028792142868 ... tomu odpoveda vzdialenost : 48 cm
Cas = 0.00297021865845 ... tomu odpoveda vzdialenost : 50 cm
Cas = 0.00331938266754 ... tomu odpoveda vzdialenost : 56 cm
Cas = 0.00401480197906 ... tomu odpoveda vzdialenost : 68 cm
Cas = 0.00486071109772 ... tomu odpoveda vzdialenost : 82 cm
Cas = 0.00379979610443 ... tomu odpoveda vzdialenost : 64 cm
Cas = 0.00387754440308 ... tomu odpoveda vzdialenost : 65 cm
Cas = 0.00427038669586 ... tomu odpoveda vzdialenost : 72 cm
Cas = 0.00424747467041 ... tomu odpoveda vzdialenost : 72 cm
Cas = 0.00432288646698 ... tomu odpoveda vzdialenost : 73 cm
Cas = 0.00424377918243 ... tomu odpoveda vzdialenost : 72 cm
Cas = 0.00424203872681 ... tomu odpoveda vzdialenost : 72 cm
Cas = 0.00422101020813 ... tomu odpoveda vzdialenost : 71 cm
Cas = 0.00426816940308 ... tomu odpoveda vzdialenost : 72 cm
Cas = 0.00438282489777 ... tomu odpoveda vzdialenost : 74 cm
Cas = 0.00442357063293 ... tomu odpoveda vzdialenost : 75 cm
Cas = 0.00438618659973 ... tomu odpoveda vzdialenost : 74 cm
Cas = 0.00433151721954 ... tomu odpoveda vzdialenost : 73 cm
Cas = 0.00434379577637 ... tomu odpoveda vzdialenost : 73 cm
Cas = 0.00477859973907 ... tomu odpoveda vzdialenost : 81 cm
Cas = 0.00855741500854 ... tomu odpoveda vzdialenost : 145 cm
Cas = 0.0183236122131 ... tomu odpoveda vzdialenost : 311 cm
Cas = 0.0152842521667 ... tomu odpoveda vzdialenost : 259 cm
Cas = 0.0159751653671 ... tomu odpoveda vzdialenost : 271 cm
Cas = 0.0163993835449 ... tomu odpoveda vzdialenost : 278 cm
Cas = 0.0160833120346 ... tomu odpoveda vzdialenost : 273 cm
Cas = 0.0165334939957 ... tomu odpoveda vzdialenost : 281 cm
Cas = 0.0126348018646 ... tomu odpoveda vzdialenost : 214 cm
Cas = 0.00543131828308 ... tomu odpoveda vzdialenost : 92 cm
Cas = 0.00525333881378 ... tomu odpoveda vzdialenost : 89 cm
Cas = 0.00639545917511 ... tomu odpoveda vzdialenost : 108 cm
Cas = 0.00694139003754 ... tomu odpoveda vzdialenost : 118 cm
Cas = 0.00728898048401 ... tomu odpoveda vzdialenost : 123 cm
Cas = 0.00810861587524 ... tomu odpoveda vzdialenost : 137 cm
Cas = 0.00810406208038 ... tomu odpoveda vzdialenost : 137 cm
Cas = 0.0066132068634 ... tomu odpoveda vzdialenost : 112 cm
Cas = 0.0064041852951 ... tomu odpoveda vzdialenost : 108 cm

Ln: 6 Col: 0
```

Obrázok 2 - Testovanie ultrazvukového senzora



Obrázok 3 - Zapojenie LIDARu



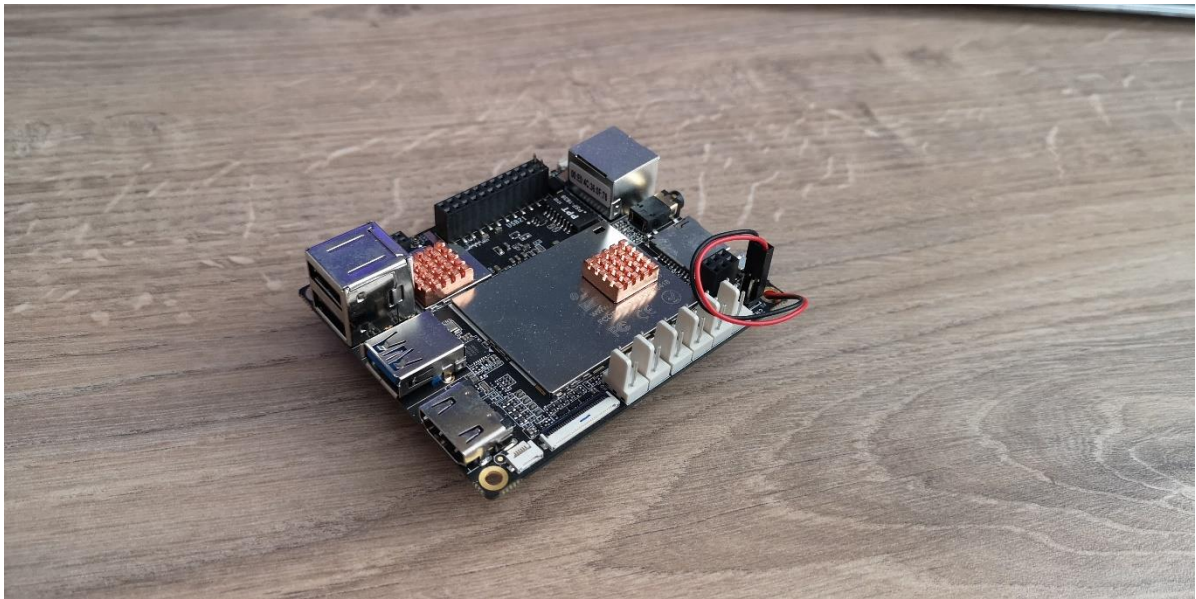
Obrázok 4 - Zapojenie stereo vize



Obrázok 5 - Testovanie technológie infračervených bodov



Obrázok 6 - Testovanie technológie infračervených bodov vo vnútri



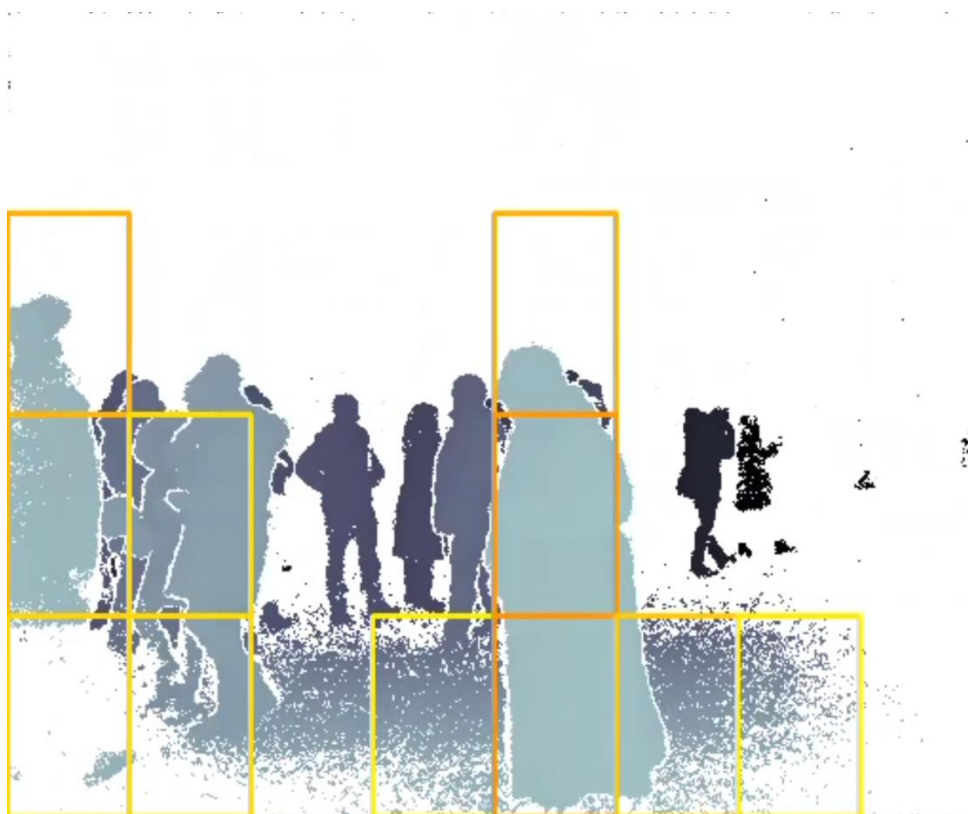
Obrázok 8 - Latte Panda Mini PC



Obrázok 7 - Porovnanie RGB a hĺbkovej mapy



Obrázok 9 - Testovanie vo vonkajších podmienkach



Obrázok 10 - Výsledná hĺbková mapa priamo z testovania vo vonkajších podmienkach

Zoznam kódov

5.5. Testovanie priemerom

```
sectorValue = 0;
sectorCount = 0;
BufferOps::wrapSector(depthBuffer, oneSector, testSectors[i].sectorNumber, widthOfFrame,
heightOfFrame);

for (int i{ 0 }; i < oneSector.size(); i++)
{
    sectorValue += oneSector[i];
    sectorCount++;
}

sectorAverage = sectorValue / sectorCount;
dangerLevel = normalizeDanger(sectorAverage);
if (frameCount >= testSectors[i].testFrom && frameCount <= testSectors[i].testTo)
{
    if (dangerLevel < testSectors[i].value)
    {
        outputResults << "Fail on Frame: " << frameCount << " - sector: " <<
testSectors[i].sectorNumber << " - correct: " << testSectors[i].value << " - actual: " <<
dangerLevel << std::endl;
        failedSectors++;
    }
}
```

5.6. Testovanie maximálnou hodnotou

```
maxValue = 0;
BufferOps::wrapSector(depthBuffer, oneSector, testSectors[i].sectorNumber, widthOfFrame,
heightOfFrame);

for (int j{ 0 }; j < oneSector.size(); j++) { if (oneSector[j] > maxValue) { oneSector[i] =
maxValue; } }

dangerLevel = normalizeDanger(maxValue);

if (frameCount >= testSectors[i].testFrom && frameCount <= testSectors[i].testTo)
{
    if (dangerLevel != testSectors[i].value)
    {
        outputResults << "Fail on Frame: " << frameCount << " - sector: " <<
testSectors[i].sectorNumber << " - correct: " << testSectors[i].value << " - actual:
" << dangerLevel << std::endl;
        failedSectors++;
    }
}
```

5.3. Testovanie vysokým exponentom

```
unsigned int exponent(std::vector<UINT16> sector)
{
    unsigned int sectorValue{ 0 }, relevantValue{ 0 };
    for (unsigned int i{ 0 }; i < sector.size(); i++)
    {
        if (sector[i] != 0)
        {
            sectorValue += 1 / (exp(sector[i]/1000));
            relevantValue++;
        }
    }
    if (relevantValue == 0) { return 0; }
    else { return normalizeDanger((sectorValue / relevantValue) * 1000); }
}

BufferOps::wrapSector(depthBuffer, oneSector, testSectors[i].sectorNumber, widthOfFrame,
heightOfFrame);
dangerLevel = exponent(oneSector);
if (frameCount >= testSectors[i].testFrom && frameCount <= testSectors[i].testTo)
{
    if (dangerLevel < testSectors[i].value)
    {
        outputResults << "Fail on Frame: " << frameCount << " - sector: " <<
testSectors[i].sectorNumber << " - correct: " << testSectors[i].value << " - actual:
" << dangerLevel << std::endl;
        failedSectors++;
    }
}
}
```

5.4. Kompresia a dekompresia súborov s hĺbkovou mapou

```
static std::size_t compress(const std::vector<UINT16>& depthBuffer,
std::vector<UINT16>& compressedBuffer)
{
    unsigned int count{ 1 };

    auto currentIt = compressedBuffer.begin();

    for (int i{ 0 }; i < depthBuffer.size(); i++)
    {
        if (i == (depthBuffer.size() - 1) || depthBuffer[i] !=
depthBuffer[i + 1])
        {
            *currentIt = count; currentIt++;
            *currentIt = depthBuffer[i]; currentIt++;
            count = 1;
        }
        else
        {
            count++;
        }
    }

    return currentIt - compressedBuffer.begin();
}

static void decompress(const std::vector<UINT16>& compressedBuffer,
std::size_t compressedBufferSize, std::vector<UINT16>& depthBuffer)
{
    auto position = depthBuffer.begin();
    unsigned int value;
```

```

        for (unsigned int i{ 0 }; i < compressedBufferSize; i += 2)
        {
            value = compressedBuffer[i + 1];
            std::fill(position, position + compressedBuffer[i],
value);
            position += compressedBuffer[i];
        }
    }
}

```

5.5. Integrácia Microsoft Kinect SDK

```

Kinect()
{
    GetDefaultKinectSensor(&kinect);
    kinect->Open();

    ComPtr<IDepthFrameSource> depthFrameSource;
    kinect->get_DepthFrameSource(&depthFrameSource);
    depthFrameSource->OpenReader(&depthFrameReader);

    ComPtr<IFrameDescription> depthDescription;
    depthFrameSource->get_FrameDescription(&depthDescription);
    depthDescription->get_Width(&widthOfDepth);
    depthDescription->get_Height(&heightOfDepth);
    Sleep(2000);
}

void getFrame(std::vector<UINT16>& depthFrame)
{
    ComPtr<IDepthFrame> tmpFrame;
    depthFrameReader->AcquireLatestFrame(&tmpFrame);

    if (tmpFrame == NULL)
    {
        return;
    }

    tmpFrame->CopyFrameDataToArray(static_cast<UINT>(depthFrame.size()),
&depthFrame[0]);
    tmpFrame->get_RelativeTime(&relativeTime);
}

```

Zoznam tabuliek

5.1. Výber hĺbkovej kamery

Varianty Kritériá	Occipital Structure	Orbbec Persee	RealSense SR300	Orbbec Astra	RealSense R200
Price	\$499 (bundle)	\$240	\$150	\$150	\$99
Tracking Method	IR	IR	IR	IR	IR
Range	0.4m – 3.5m	0.4m – 8m	0.2m – 1.2m	0.4m – 8m	0.5m – 3.5m
RGB Image	iOS Camera resolution	1280×720, 30 FPS	1920×1080, 30 FPS	1280×960, 10 FPS	1920×1080, 30 FPS
Depth Image	640×480 at 30fps, 320×240 at 60fps	640×480, 16 bit, 30 FPS	640×480, 60 FPS	640×480, 16 bit, 30 FPS	640×480, 60 FPS
Connectivity	Lightning on iOS, USB elsewhere	Ethernet	USB 3.0	USB 2.0	USB 3.0
Physical Dimensions	119.2×28×29 mm	172×63×56 mm	14×20×4 mm	160×30×40 mm	130×20×7 mm
Works outdoors	X	X	X	X	X
Skeleton tracking	X	X	X	✓ (only hand positions)	X
Facial tracking	X	soon	✓	X	✓
3D scanning	✓	X	✓	X	✓
Simultaneous apps	X	soon	✓	X	✓
Gesture Training	X	X	X	X	X
Gesture Detection	✓	X	✓	X	X
Toolkits	iOS, Unity3D, OpenNI	C++, Java, OpenNI, ROS	Java, JavaScript, Processing, Unity3D, Cinder	OpenNI	Java, JavaScript, Processing, Unity3D, Cinder
HW Requirements		Windows 7 or later x86-based processor @ 1.8+ ghz			

Varianty Kritériá	ZED Stereo Camera	RealSense F200	Kinect for Xbox One	DUO mini lx	Leap Motion
Price	\$449	\$99	\$100	\$695	\$100
Tracking Method	Stereo RGB cameras	IR	active IR	Passive IR	IR
Range	1.5m – 20m	0.2m – 1.2m	0.5m – 4.5m	0.3m – 2.4m	0.025m – 0.6m
RGB Image	configurable between 1280×480, 120 FPS and 4416×1242, 15 FPS	1920×1080, 30 FPS	1920×1080, 30 FPS	320×120, 360 FPS or 752×480, 56 FPS	N/A
Depth Image	640×480, 120 FPS or 2208×1242, 15 FPS	640×480, 60 FPS	512×424, 30 FPS	320×120, 360 FPS or 752×480, 56 FPS	20 to 200+ FPS
Connectivity	USB 3.0	USB 3.0	USB 3.0	USB 2.0	USB 2.0
Physical Dimensions	175×30×33 mm	150×30×58 mm	250×66×67 mm	52×25×11 mm	76×30×17 mm
Works outdoors	✓	✗	✓	✓	✗
Skeleton tracking	✗	✗	✓ (six skeletons)	✗	✗
Facial tracking	✗	✓	✓	✓	✗
3D scanning	✓	✓	✓	✓	✗
Simultaneous apps	✗	✓	✓	✗	✗
Gesture Training	✗	✗	✓ (Visual Gesture Builder)	✗	✓
Gesture Detection	✗	✓	✓ (hand open, closed, lasso)	✗	✓
HW Requirements	Quad-core 2,7GHz or faster 8GB NVIDIA GPU NVIDIA GTX1060 or higher USB 3.0 Windows 7, 8.1, 10, Ubuntu 16.04, L4T (Jetson)		64 bit processor 4 GB Memory DX11 Intel HD 4400 ATI Radeon HD 5400 series ATI Radeon HD 6570 ATI Radeon HD 7800 NVidia Quadro 600 NVidia GeForce GT 640 NVidia GeForce GTX 660		

5.2. Výber Mini PC

Kritériá Varianty	VoCore	Arduino INDUSTRIAL 101	Banana Pi M3	C.H.I.P	RASPBERRY PI 3 MODEL B	Latte Panda
Cena zariadenia (€)	15	35	100	9	40	150
Rozmery zariadenia (mm)	25,6 x 25,6	50,8 x 43,2	92 x 60	40 x 60	56 x 85	88 x 70
CPU	360 MHz, MIPS 24K	Atheros AR9331, 400 MHz	A83T ARM Cortex-A7 octa-core 1.8 GHz	1GHz Allwinner R8	1.2GHz 64- bit quad- core ARMv8 CPU	Intel Cherry Trail Z8350 Quad Core
Odber zariadenia @5V (A)	0,2	0,1	2,5	0,3	2,5	2
Pocet GPIO pinov	30	20	40	80	40	24

Použitá literatúra, zdroje

- [1] Juraj Lipovský – Tvorba mobilnej aplikácie pre nevidiacich – priestorová orientácia –
Bakalárska práca 2017, Použité: 1.12.2018
- [2] <https://en.wikipedia.org/wiki/Lidar>
Názov: Lidar, Použité: 2.12.2018
- [3] https://en.wikipedia.org/wiki/Computer_stereo_vision
Názov: Computer stereo vision, Použité: 2.12.2018
- [4] [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782041\(v=ieeb.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782041(v=ieeb.10))
Názov: Kinect for Windows version 2.0 SDK, Použité: 12.11.2018
- [5] <https://docs.opencv.org/3.0-beta/modules/refman.html>
Názov: OpenCV API Reference, Použité: 12.11.2018
- [6] <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/> -
Ultrasonic Sensor HC-SR04, Použité: 20.12.2018
- [7] <http://vocore.io/>
Názov: VoCore | Coin-sized Linux Computer, Použité: 12.3.2019
- [8] <http://www.arduino.org/>
Názov: Arduino - Open Source Products for Electronic Projects, Použité: 25.10.2016
- [9] <https://nextthing.co/pages/chip>
Názov: Get C.H.I.P. and C.H.I.P. Pro - The Smarter Way to Build Smart Things,
Použité: 25.10.2016
- [10] <https://www.raspberrypi.org/magpi/>
Názov: The MagPi Magazine - The official Raspberry Pi magazineThe MagPi Magazine
Použité: 25.10.2016
- [11] <https://www.forbes.com/sites/miguelhelft/2016/05/18/inside-sundar-pichais-plan-to-put-ai-everywhere/#14f62a144a2e>
Názov: Inside Sundar Pichai's Plan To Put AI Everywhere, Použité: 2.4.2019
- [12] <https://3digital.striz.sk/kapsys-smartvision-android-pre-slabozrakych-a-nevidiacich/>
Názov: Kapsys SmartVision, Android pre slabozrakých a nevidiacich, Použité: 3.4.2019

- [13] <https://www.inverse.com/article/47353-google-earnings-ceo-reveals-smart-assistant-phone-domination>
Názov: Google CEO Reveals How Its Smart Assistant Is Key To Smartphone Domination,
Použité: 3.4.2019
- [14] https://en.wikipedia.org/wiki/Run-length_encoding
Názov: Run-length encoding, Použité: 10.12.2018
- [15] SHAW Zed – Learn the Python the HARD WAY, d. v. 1.10.2013
ISBN-10: 0-321-88491-4
- [16] Andrew Koenig – Accelerated C++: Practical Programming by Example (C++ in Depth),
d. v. 14.8.2000
ISBN-13: 978-0201703535
- [17] Bjarne Stroustrup – The C++ Programming Language, d. v. 10. 7. 2013
ISBN: 978-0133522853
- [18] <https://sk.wikibooks.org/wiki/Moreplavba/Navig%C3%A1cia>
Názov: Moreplavba/Navigácia – Wikiknihy, Použité: 25.3.2017
- [19] https://sk.wikipedia.org/wiki/Glob%C3%A1lny_lokaliza%C4%8Dn%C3%BD_syst%C3%A9m
Názov: Globálny lokalizačný systém – Wikipédia Použité: 12.12.2016
- [20] <http://blindsquare.com/about/>
Názov: What is BlindSquare, Použité: 14.3.2018