Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jan Markočič

Računalniški vid za pomoč slepim in slabovidnim pri navigaciji

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Matej Kristan

Ljubljana 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V nalogi obravnavajte problem razvoja sistema za usmerjanje slepih in slabovidnih s pomočjo računalniškega vida. Sistem naj temelji na monokularni kameri, ki jo uporabnik nosi pritrjeno na telesu. Na podlagi pregleda literature izberite najprimernejšo metodo za izgradnjo trodimenzionalnega modela uporabnikove neposredne okolice iz zaporedja s kamero zajetih slik. Razvijte algoritem, ki bo sposoben hitrega procesiranja zgrajenega modela in določanja smeri, ki minimizira nevarnost trka uporabnika z oviro. Predlagajte tudi postopek podajanja informacije o smeri željenega gibanja, ki bo primeren za slepe in slabovidne. Implementirajte prototip sistema in analizirajte njegovo uspešnost na realnih primerih.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Jan Markočič, z vpisno številko **63100297**, sem avtor diplomskega dela z naslovom:

Računalniški vid za pomoč slepim in slabovidnim pri navigaciji

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mateja Kristana,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 7. septembra 2015 Podpis avtorja:

Zahvaljujem se profesorjem in asistentom na fakulteti za trud in podano znanje ter pomoč pri študiju. Posebna zahvala gre mentorju doc. dr. Mateju Kristanu za pomoč, spodbujanje in sodelovanje pri nastanku diplomskega dela. Zahvaljujem se tudi družini za podporo.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
SLAM	simultaneous localization and mapping	sočasna lokalizacija in mapiranje prostora
GPS	global positioning system	globalni sistem pozicioniranja
RFID	radio frequency identification	radio frekvenčna identifikacija
ROS	robot operating system	robotski operacijski sistem
TCP	transmition control protocol	protokol za nadzor prenosa
SIFT	scale-invariant feature transform	na skalo invariantna transformacija značilnice
SFML	simple and fast multimedia library	enostavna in hitra knjižnica za multimedijo

Povzetek

Slepi in slabovidni ljudje imajo veliko težav pri opravljanju vsakdanjih opravil. Tehnologija lahko pomaga pri reševanju problemov slepih. Še posebej velik problem predstavlja navigacija v nenadzorovanih okoljih, za kar obstaja veliko tehnoloških pripomočkov. Tehnologija je že toliko napredovala, da je velika procesorska moč na voljo ljudem po relativno nizki ceni. Novi algoritmi za rekonstrukcijo okolja omogočajo učinkovito zaznavanje ovir v realnem času. S pomočjo takšne tehnologije se robotika vzpenja, lahko pa jo uporabimo tudi za pomoč slepim. V tem delu predstavljamo sistem, ki na podlagi sekvence slik iz kamere razbere oddaljenost videnih predmetov. To stori s pomočjo direktnega monokularnega algoritma SLAM. Glede na zaznane ovire v vidnem polju kamere s predlagano metodo izračuna najbolj obetavno pot za uporabnika. Nato uporabniku svetuje, kam naj se premakne, da se ta ne zaleti. Uporabnik od sistema prejme zvočno informacijo prek stereo slušalk. V smeri, ki si jo sistem izbere, se oglaša zvok, katerega uporabnik triangulira in mu sledi. Testi so pokazali, da se uporabnik s sistemom v ovire zaleti šest-krat redkeje kot brez sistema, kar nakazuje na velik potencial našega pristopa.

Ključne besede: monokularna kamera, SLAM, ocenjevanje globin, izogibanje oviram, stereofonski zvok.

Abstract

Accomplishing many of the every-day activities present a significant challenge to visually impaired people. One of such challenges is navigation in an unsupervised environment, which might be well addressed with technological solutions. The recent technological advances have made computational devices with significant processing power a widely affordable commodity. In parallel, the advances in computer-vision-based systems allow scene reconstruction in real time paving the way for realistic systems for obstacle detection. As the field of robotics is quickly advancing with the help of such technologies, we can use it to help visually impaired people. In this work we propose a computer-vision-based system for navigation of visually-impaired in unsupervised environment. The system is composed of environment analysis sub-system and the navigation sub-system. A direct monocular SLAM is used to estimate a model of the user's environment. A new algorithm for analysis of this model and detection of potential obstacles is proposed. Based on the obstacle detection result, the system selects the most appropriate direction of motion to minimize the probability of collision. This information is provided to the user by a sound-based interface – a stereo headset. The direction of motion is provided to the user by placing sources of sounds in the virtual three-dimensional space that user perceives through the stereo headset. The system was evaluated in a real environment. The results have shown that the subjects wearing the system were six times less likely to collide with the obstacles in their environment than the subjects not wearing the system. Albeit basic, the experiments indicate a great potential of the

proposed system for navigation in realistic environments.

Keywords: monocular camera, SLAM, depth estimation, obstacle avoidance, stereophonic sound.

Poglavje 1

Uvod

1.1 Motivacija

V vsakdanjem življenju ljudje opravljajo veliko nalog, ki ne predstavljajo težav. Za slepega ali slabovidnega človeka pa je vsaka naloga kompleksnejša kot za zdravega, saj ne prejema oz. procesira vizualnih informacij, čemur je namenjeno med 30 in 40 odstotkov človekovih možganov [21]. Leta 2014 je bilo na svetu 285 milijonov ljudi s hudo prizadetim vidom, od katerih je 39 milijonov slepih in 246 milijonov slabovidnih [3]. V Ameriki je leta 2012 imelo službo le 37.7 odstotkov ljudi s hudo prizadetim vidom [26], saj ne morejo opravljati dela kot bi ga lahko zdrav človek. Tehnologija lahko tem ljudem pri njihovih težavah pomaga, saj obstajajo zelo natančni sistemi za zajem 3D informacije iz prostora. 3D sisteme s pridom uporabljamo v mobilni robotiki. Če se lahko robot samostojno premika po prostoru, lahko z istimi tehnologijami pridobivamo informacije o okolju in jih posredujemo slepemu človeku.

Sistem s pomočjo monokularne kamere in metode LSD-SLAM zajema slike okolja in računa razdalje do predmetov v vidnem polju. Zaznane predmete nato upošteva pri računanju najbolj obetavne poti za uporabnika, torej ga vodi mimo vseh zaznanih ovir. Pristop predpostavlja, da uporabnik približno ve, kje se nahaja in kam mora iti. Pri tem mu sistem pomaga točno

določiti smer hoje, da ne pride do trka z oviro. Za komunikacijo med sistemom in uporabnikom je mogoče uporabiti stimulatorje za enega ali več čutov, ki ostajajo uporabniku. V tem diplomskem delu to storimo s postavitvijo virtualnega 3D zvoka na lokacijo, kamor želimo, da se uporabnik premakne. Uporabnik sliši zvok prek stereo slušalk in naravno triangulira izvor zvoka.

1.2 Sorodna dela

Z razvojem tehnologije se pojavlja veliko sistemov namenjenih pomoči slepim ljudem pri navigaciji v prostoru. Ti uporabljajo najrazličnejše senzorje za pridobivanje informacij o okolju. Nekateri uporabljajo GPS [9, 29, 4], nekateri sonarje [1, 29], laserske senzorje [16, 36], kombinacijo senzorjev [29, 4] itd. Za posredovanje informacij človeku pa uporabljajo predvsem zvok [4, 29, 16, 24] in vibro-taktilne naprave [1] .

Pristopi se v splošnem delijo glede na to, ali so namenjeni notranji uporabi ali zunanji. Sistemi, ki uporabljajo GPS [9, 29, 4] za lokalizacijo, so omejeni na natančnost, ki jo tehnologija ponuja. Slepi ljudje imajo pri tem težavo, saj potrebujejo natančno informacijo o nahajališču cilja (npr. vhodnih vrat).

Za natančnejšo navigacijo si nekateri sistemi pomagajo z detekcijo in triangulacijo signalov oddajnikov bluetooth ali wi-fi, ki so postavljeni na določenih točkah po mestu [33].

Nekateri sistemi so narejeni za specifično uprabo, kot npr. BlindShopping [22], ki omogoča slepemu uporabniku navigacijo po trgovini s pomočjo pametnih telefonov ter prepoznavanje produktov s pomočjo tehnologie RFID in branjem QR kod.

Eden zanimivejših sistemov je Navigational Aids For The Visually Impaired (NAVI) [36], ki za delovanje uporablja Microsoftov Kinect, vibrotaktilni pas, slušalke in računalnik. Kinect meri razdalje do videnih objektov, sistem pa govori uporabniku, kam naj se premakne. Pri tem si pomaga z vnaprej nameščenimi markerji [17], da ugotovi, kje v prostoru se nahaja.

Večina sistemov, ki za obveščanje človeka uporabljajo zvok, to storijo z umetno generiranim govorom [29, 36]. Tako simulirajo pomočnika, ki slepemu uporabniku poroča o okolici. Za razliko od teh, nekateri sistemi, kot je Microsoft Independence Day [24], opozorijo uporabnika z virtualnim 3D zvokom, ki izvira iz izbrane točke, kot je npr. referenčna točka ali ovira. Pri tem pristopu se prepusti uporabniku nalogo, da lokalizira izvor zvoka. Človeku je reševanje tovrstnega problema naravno, saj s pomočjo stereo zvoka in triangulacije brez težav ugotovi, od kod prihaja zvok. Ta pristop smo tudi uporabili v diplomskem delu.

1.3 Prispevki

V diplomskem delu smo razvili in implementirali sistem, ki bi lahko slepim in slabovidnim pomagal pri navigaciji. Vhodni podatki sistema so zaporedje slik, ki jih servira kamera mobilnega telefona. Metoda LSD-SLAM [6] na podlagi slik izračuna oddaljenost predmetov v vidnem polju kamere. Razvili smo algoritem, ki upošteva položaje zaznanih predmetov in izračuna najbolj obetavno pot za uporabnika, po kateri bo prispel najdlje, brez zaletavanja v ovire. Informacijo o smeri, kam naj se uporabnik premakne, sistem posreduje prek stereo slušalk. Generira se virtualni zvok, kateremu se postavi izvor na določeno razdaljo v izbrani smeri. Uporabnik tako sliši, od kod zvok prihaja in mu enostavno sledi.

Sistem ne predstavlja nadomestila za belo palico, s katero si večina slepih pomaga pri zaznavanju ovir v okolici. Je pa z njim mogoče najti pot mimo ovir, ne da bi se jih dotaknili, in razbrati, npr., točno usmerjenost hodnika.

1.4 Struktura diplomske naloge

Diplomsko delo je v nadaljevanju razdeljeno na štiri poglavja. V Poglavju 2 je predstavljen glavni del teorije, ki jo sestavni deli sistema uporabljajo za delovanje, vključno z razlago delovanja najkompleksnejšega dela sistema,

metode LSD-SLAM [6]. Poglavje 3 opisuje, kako deluje del sistema, ki skrbi za izogibanje oviram. Vse o implementaciji rešitev in testiranju sistema je predstavljeno v Poglavju 4. Poglavje 5 pa je namenjeno sklepu in idejam o možnih nadgradnjah sistema.

Poglavje 2

Teoretični okvir

V tem poglavju so predstavljeni principi in metode, ki jih sistem uporablja pri delovanju. V Podpoglavju 2.1 je predstavljen robotski operacijski sistem (ROS) [28]. Osnove o teoriji nastanka slike v kameri so opisane v Podpoglavju 2.2. Podpoglavje 2.3 opisuje problem SLAM [5] in bazične metode, ki ga rešujejo. Metoda za direktno ocenjevanje strukture iz gibanja je opisana v Podpoglavju 2.4. Poglavje se konča s Podpoglavjem 2.5, kjer razložimo delovanje stereofonskega zvoka.

2.1 Robotski operacijski sistem

Odprtokodni robotski operacijski sistem (angl. robot operating system, ROS) [28] vsebuje vrsto knjižnic in orodij namenjenih razvoju programske opreme za robote. Vključuje vse od gonilnikov do najnaprednejših algoritmov različnih področij računalništva.

Razvijalci ROS-a svoje programe hranijo v obliki paketov, kar omogoča, da ostali razvijalci brez težav namestijo objavljene programe. To je vzrok za hitro širjenje znanja, nadgradnjo obstoječih algoritmov in predvsem razvoj metod, ki uporabljajo že razvite komponente. Tako se razvijalci lahko fokusirajo na svoje področje razvoja. Vsak paket lahko vsebuje več programov. Te se pa zažene kot vozlišče (angl. node), kar postavi vmesnik z ROS-om prek

servisov (angl. services) in tem (angl. topics).Vhodna in izhodna vozlišča lahko med sabo komunicirajo prek sporočil (angl. messages), za katere se definira format podatkov. Za komunikacijo se uporablja protokol TCPROS, ki koristi standardne TCP/IP [8] vtičnike za prenos podatkov.

Pred zagonom programa, ki deluje v okolju ROS, je potrebno pognati jedro ROS-a *roscore*. Ta omogoča vozliščem, da se registrirajo na listo vozlišč, prijavijo na željene teme in objavijo morebitne svoje teme. Ker *roscore* hrani listo zagnanih vozlišč, ve, katera so zagnana. Če dve vozlišči želita komunicirati med sabo, jih *roscore* postavi na zvezo in komunikacija lahko poteka.

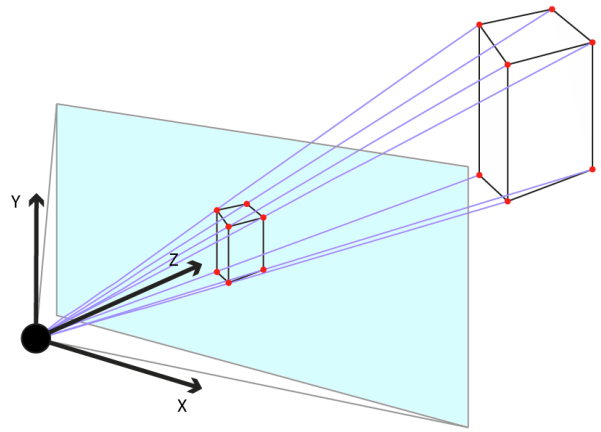
ROS vsebuje več uporabnih orodij, kot so sistem za beleženje dogajanja, program za snemanje in predvajanje sporočil, ki se objavljajo na temah (*rosbag*), vizualizacijski program (*rviz*) itd.

2.2 Model nastanka slike v monokularni kameri

Lego točk v prostoru lahko opišemo s tremi koordinatami. Kamera v posameznem trenutku prejme le dvodimenzionalno projekcijo videnega prostora na svoj senzor. Če želimo vedeti, v kateri smeri glede na kamero se nahaja viden predmet, moramo definirati transformacijo iz 3D koordinatnega sistema kamere na njeno 2D slikovno ravnino. V tem poglavju je opisano, kako nastane slika na senzorju kamere.

Če sredino kamere $\mathbf{C}$, ki je tudi izhodišče svojega koordinatnega sistema, povežemo s točkami v 3D prostoru pred njo, se povezave sekajo s slikovno ravnino (Slika 2.1). Presečišča tako upodobijo projekcijo 3D prostora na 2D ravnino. Kje točno na slikovni ravnini se pojavi projekcija $\mathbf{x}$ določene 3D točke $\mathbf{X}$, je odvisno od parametrov kamere, kot so goriščna razdalja f, nahajališče sredine kamere $\mathbf{C}$, projekcijski koeficienti itd.

Pri računanju transformacije iz 3D koordinatnega sistema kamere v njeno 2D slikovno ravnino določimo glavno točko $\mathbf{p}$ (angl. principal point) na sre-



Slika 2.1: Na sliki je prikazana projekcija predmeta na slikovno ravnino.

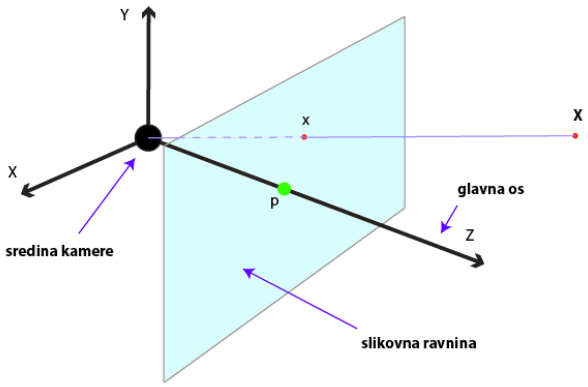
dini slikovne ravnine. Koordinatni sistem kamere normaliziramo s postavitvijo njene sredine **C** na način, da os z, t.i. glavna os (angl. principal axis), pod pravim kotom prebada slikovno ravnino v glavni točki **p** (Slika 2.2). Po modelu kamere s točkasto odprtino (angl. pinhole camera) lahko s pomočjo homogenih koordinat [19] za določeno točko $\mathbf{X} = [X_x, X_y, X_z]^T$ v 3D koordinatnem sistemu kamere izračunamo koordinate njene projekcije $\mathbf{x} = [x_x, x_y, x_z]^T$ na slikovni ravnini po formuli

$$\begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} = \begin{pmatrix} f \cdot X_X / X_Z \\ f \cdot X_Y / X_Z \\ 1 \end{pmatrix} = \begin{pmatrix} f \cdot X_X \\ f \cdot X_Y \\ X_Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_X \\ X_Y \\ X_Z \\ 1 \end{pmatrix}, \tag{2.1}$$

pri čemer je projekcijska matrika

$$\mathbf{P_0} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.2}$$

Točka **x** je v tem koraku zapisana v normaliziranem koordinatnem sistemu kamere, ki ima izhodišče v glavni točki **p**. Da bi pretvorili koordinate točke **x** v koordinatni sistem slike, ki ima izhodišče v kotu senzorja kamere, potre-



Slika 2.2: Na sliki je prikazana postavitev glavne točke **p** in ostalih elementov.

bujemo koordinati p_x in p_y glavne točke **p** v koordinatnem sistemu slike. Pri tem dobimo nadgrajeno formulo

$$\begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} = \begin{pmatrix} fX_X + X_Z p_x \\ fX_Y + X_Z p_y \\ X_Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_X \\ X_Y \\ X_Z \\ 1 \end{pmatrix}. \tag{2.3}$$

Za pretvorbo koordinat točke **x** iz metrov v enoto slikovnih točk moramo poznati velikost senzorja kamere $W_S \times H_S$ v metrih in velikost njegove pravokotne matrike slikovnih elementov $M_x \times M_y$. Pomagamo si z

$$m_x = M_x / W_s, \tag{2.4}$$

$$m_y = M_y / H_s, \tag{2.5}$$

s čimer dobimo funkcijo

$$\begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_X \\ X_Y \\ X_Z \\ 1 \end{pmatrix}, \tag{2.6}$$

in njeno okrajšano različico

$$\begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} = \begin{bmatrix} \alpha_x & 0 & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_X \\ X_Y \\ X_Z \\ 1 \end{pmatrix}. \tag{2.7}$$

Pri tem je potrebno upoštevati, da matrika slikovnih elementov senzorja ni vedno pravokotna. Če je ta zamaknjena, se v projekcijski matriki $\mathbf{P}_0$ pojavi element s, ki popravi zamaknjenost

$$\begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_X \\ X_Y \\ X_Z \\ 1 \end{pmatrix}. \tag{2.8}$$

Projekcijsko matriko je mogoče razdeliti na

$$\mathbf{P_0} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.9}$$

pri čemer je

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

kalibracijska matrika kamere, ki se jo uporablja pri projekciji točk iz 3D koordinatnega sistema kamere v 2D koordinatni sistem slike in poskrbi, da se merska enota spremeni v slikovne točke.

Pri kamerah, ki uporabljajo leče, se pojavlja tudi radialna distorzija. Poznamo več modelov radialne distorzije [32]. Ta spremeni sliko tako, da vsak slikovni element nelinearno premakne po premici proti sredini slike ali stran od nje. Dolžina premika je odvisna od oddaljenosti slikovnega elementa od sredine slike. Pri premiku slikovnega elementa se mu kot, ki je relativen na sredino slike, ne spremeni. Za popravek slike z radialno distorzijo se uporabljajo polinomske funkcije. Višja kot je stopnja polinoma, bolj natančno

se izračuna popravek, ampak izračun postane procesno zahtevnejši. Pred računanjem preslikave po modelu kamere s točkasto odprtino je potrebno nelinearnost odstraniti.

2.3 Gradnja zemljevida iz videoposnetka

Na področju robotike je istočasna lokalizacija in mapiranje (angl. simultaneous localization and mapping, SLAM) [5] problem, ki zajema sledenje poziciji robota v okolju in istočasno gradnjo zemljevida okolja. Pri gradnji zemljevida se v principu potrebuje znano pozicijo robota in pri računanju pozicije robota se potrebuje znan zemljevid okolice. Zato SLAM velja za enega najzahtevnejših problemov v robotiki. Rešitev problema pa predstavlja "sveti gral mobilne robotike" [5], saj bi s pomočjo te lahko dosegli resnično avtonomnost robota.

Zemljevidi, ki jih algoritmi SLAM generirajo, so lahko zelo različni. Robot potrebuje tak zemljevid, da mu bo omogočal navigacijo. Terenski roboti [12, 13], ki se samostojno premikajo po prostoru, pogosto gradijo dvodimenzionalni zemljevid tlorisa prostora. Za leteče robote [27]pa pride v poštev gradnja 3D zemljevida, saj se lahko premika neodvisno po vseh treh dimenzijah. Če bi se pa robot npr. premikal le po tirnicah oz. zastavljenih poteh, bi mu zadoščal topološki model zemljevida.

Premikajoči robot v posameznih trenutkih opazuje okolico in meri položaje zaznanih predmetov. Pri SLAM-u sta pozicija robota in zemljevid spremenljivki, kateri se računa istočasno. Pri tem pa se napake akumulirajo.

Pri reševanju problema SLAM je potrebno za vsak časovni korak t na podlagi meritev senzorjev $\mathbf{z_t}$ oceniti robotovo lokacijo $\mathbf{x_t}$ in zemljevid okolja $\mathbf{m_t}$. Izračunati je potrebno verjetnostno porazdelitev

$$P(\mathbf{m_t}, \mathbf{x_t}|\mathbf{z_{1:t}}), \tag{2.11}$$

Ki pove skupno verjetnost stanja mape $\mathbf{m_t}$ in robota $\mathbf{x_t}$ glede na vse opažene

meritve do trenutka t, $\mathbf{z_t}$. Z upoštevanjem Bayesovega pravila dobimo

$$P(\mathbf{x_t}|\mathbf{z_{1:t}}, \mathbf{m_t}) = \sum_{\mathbf{m_{t-1}}} P(\mathbf{z_t}|\mathbf{x_t}, \mathbf{m_t}) \sum_{\mathbf{x_{t-1}}} P(\mathbf{x_t}|\mathbf{x_{t-1}}) P(\mathbf{x_{t-1}}|\mathbf{m_t}, \mathbf{z_{1:t-1}})/N, \tag{2.12}$$

kjer je N število meritev. Na podoben način se lahko izračuna tudi verjetnost zemljevida

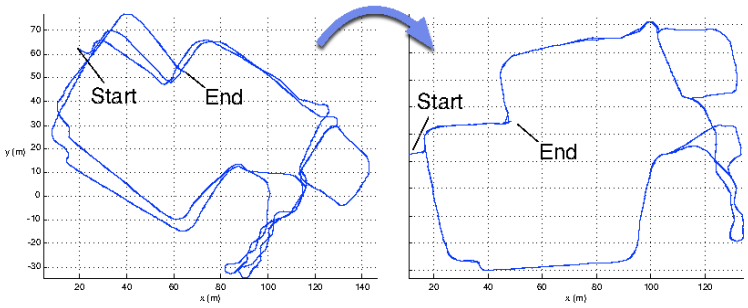
$$P(\mathbf{m_t}|\mathbf{x_t}, \mathbf{z_{1:t}}) = \sum_{\mathbf{x_t}} \sum_{\mathbf{m_t}} P(\mathbf{m_t}|\mathbf{x_t}, \mathbf{m_{t-1}}, \mathbf{z_t}) P(\mathbf{m_{t-1}}, \mathbf{x_t}|\mathbf{z_{1:t-1}}, \mathbf{m_{t-1}}). \tag{2.13}$$

Pri algoritmih SLAM ostaja problem, da se nedoločenost položaja robota povečuje skozi čas, saj se majhne napake posameznih meritev akumulirajo. Daljšo pot kot naredi robot, večja je akumulirana napaka. Za popravek je potrebna posebna metoda, ki meri podobnost trenutne okolice z okolico, ki jo je že spoznal oz. opazil v preteklih meritvah. Če postane ta podobnost dovolj velika, se sproži mehanizem zapiranja zanke (angl., loop closure), ki popravi robotu ocenjeno pozicijo in korigira zemljevid. Na Sliki 2.3 je prikazana pot robota, ki je večkrat prehodil isto pot. Pred zapiranjem zanke je vidna napaka meritev, ki se je skozi čas akumulira. Po zapiranju zanke pa robot upošteva, da so se prostori med premikanjem ponavljali. Zato svoje podatke popravi, kar je na desni polovici Slike 2.3 razvidno s popolnoma poravnano potjo. Lastnosti okolice se pogosto shrani s pomočjo značilnic SIFT [20], ki se jih primerja z vsemi starejšimi vnosi [25].

2.4 Direktno ocenjevanje strukture iz gibanja

Metoda za neposredno ocenjevanje strukture iz gibanja, imenovana Large-scale direct monocular SLAM (LSD-SLAM) [6, 7], je metoda, ki poleg zelo preciznega ocenjevanja položaja kamere iz sekvence slik zgradi 3D rekonstrukcijo okolice.

Metoda deluje v realnem času in upošteva vse informacije v slikah, saj dela neposredno z intenzitetami vseh slikovnih elementov. Metoda deluje



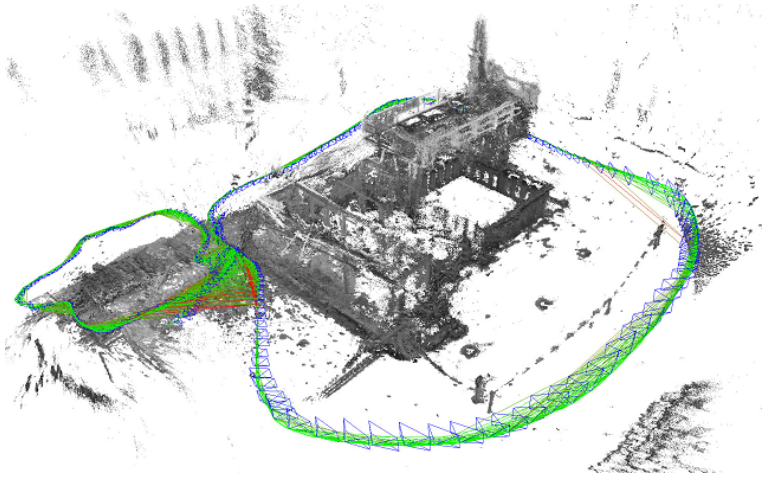
Slika 2.3: Na levem delu slike je prikazana pot robota pred zapiranjem zanke, na desnem delu pa pot robota po zapiranju zanke.

v realnem času. Za razliko od metod, ki delajo na podlagi značilnih točk [10], LSD-SLAM dela neposredno na intenzitetah slike, torej upošteva vse informacije v sliki. Razdaljo računa za vse slikovne elemente, ki nosijo dovolj vizualne informacije oz. so si dovolj različne od sosednjih. Zmožna je graditi velike 3D zemljevide okolice (Slika 2.4) zaradi robustnih metod, ki jih uporablja pri gradnji zemljevida.

V nadaljevanju so opisane glavne komponente metode LSD-SLAM. Ta najprej izračuna nov položaj in usmerjenost kamere (Podpoglavje 2.4.1), nato uporabi novo sliko iz kamere za nadgradnjo rekonstrukcije okolja (Podpoglavje 2.4.2).

2.4.1 Ocenjevanje položaja kamere

LSD-SLAM [6] uporablja neposredno prileganje slik za ocenjevanje položaja kamere. Prileganje izvaja med najnovejšo sliko, ki je na voljo, in zadnjo ključno sliko (angl. keyframe). Če se kamera preveč oddalji od znanega prostora - zgrajenega zemljevida, se ustvari nova ključna slika na podlagi zadnje prejete slike iz kamere. Vsaka od ključnih slik hrani ustrezno sliko iz kamere I_i, inverzno globinsko sliko D_i in varianco inverzne globine V_i. Ključne slike se uporablja, ker se tako napaka pri meritvah ne akumulira



Slika 2.4: Primer velikega zemljevida, ki je bil zgrajen z metodo LSD-SLAM. Na sliki je prikazana zgradba, okrog katere se je premikala kamera.

tako hitro.

Relativna pozicija ξ_{ji} nove slike I_j se glede na obstoječo ključno sliko $K_i = (I_i, D_i, V_i)$ izračuna z minimiziranjem fotometrične napake z normalizirano varianco. Pri tem si pomaga s šestdimenzionalnim vektorjem $\boldsymbol{\xi_{ji}}$, ki predstavlja transformacijo poze iz slike i v sliko j. Pri računanju fotometrične napake z določeno točko $\mathbf{p}$ in globino d uporabljamo 3D projekcijsko ovojno funkcijo

$$\omega(\mathbf{p}, d, \boldsymbol{\xi}) = \begin{pmatrix} x'/z' \\ y'/z' \\ 1/z' \end{pmatrix}, \tag{2.14}$$

kjer je vektor

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = exp(\boldsymbol{\xi}) \begin{pmatrix} p_x/d \\ p_y/d \\ 1/d \\ 1 \end{pmatrix}. \tag{2.15}$$

Minimizirano fotometrično napako se računa kot

$$E_p(\boldsymbol{\xi_{ji}}) = \sum_{\mathbf{p} \in \Omega_{D_i}} \left\| \frac{r_p^2(\mathbf{p}, \boldsymbol{\xi_{ji}})}{\sigma_{r_p(\mathbf{p}, \boldsymbol{\xi_{ji}})^2}} \right\|_\delta, \tag{2.16}$$

kjer je

$$r_p(\mathbf{p}, \boldsymbol{\xi_{ji}}) = I_i(\mathbf{p}) - I_j(\omega(\mathbf{p}, D_i(\mathbf{p}), \boldsymbol{\xi_{ji}})) \tag{2.17}$$

in

$$\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi_{ji}})} = 2\sigma_I^2 + \left(\frac{\partial r_p(\mathbf{p}, \boldsymbol{\xi_{ji}})}{\partial D_i(\mathbf{p})}\right)^2 V_i(\mathbf{p}). \tag{2.18}$$

Pri tem je Huberjeva norma označena z $\|\cdot\|_\delta$, t.j.,

$$\|r^2\|_\delta = \begin{cases} \frac{r^2}{2\delta} & \text{if } |r| \leq \delta \\ |r| - \frac{\delta}{2} & \text{sicer.} \end{cases}. \tag{2.19}$$

Oznaka $\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi_{ji}})}$ označuje varianco ostanka (angl. residual's variance).

Metoda si pomaga s propagiranjem negotovosti in z uporabo varianc inverznih globin V_i. Metoda predpostavlja, da ima opravka z Gaussovim šumom z varianco σ_I^2. Minimizacija se izvede z iterativno uporabo Gauss-Newtonove optimizacije [6]. Ta pristop eksplicitno upošteva spremenljiv šum pri računanju globin, kar je posebej pomembno pri direktnem monokunarnem SLAM-u, kjer se ta šum zelo razlikuje od slikovne točke do slikovne točke - odvisno kako dolgo so v vidnem polju kamere.

2.4.2 Rekonstrukcija okolice

LSD-SLAM ne računa stereo disparitet med dvemi najnovejšimi slikami, ki jih prejme iz kamere, ampak jo računa med najnovejšo sliko in najbližjimi ključnimi slikami. Metoda na podlagi disparitet slikovnih točk na stereo slikah gradi graf položajev (angl. pose-graph) ključnih slik. Za vsako ključno sliko izračuna (in nato posodablja) pol-gosto (angl. semi-dense) globinsko sliko (angl. depth map). Globinska slika je 2D matrika, ki za vsako slikovno točko ključne slike hrani izračunano globino.

Samo ključne slike vsebujejo globinsko sliko. Vse ostale slike služijo za izboljšavo globinske slike okoliških ključnih slik. Pri tem se uporablja veliko število učinkovitih stereo primerjav za regije slik, za katere je pričakovana stereo točnost dovolj visoka.

Izračun globinske slike

Ko metoda ustvari novo ključno sliko, ji zgradi lastno globinsko sliko. To naredi na podlagi globinske slike najprimernejše ključne slike iz okolice. Primernost ključne slike računa na podlagi relativne razdalje in kota med novo in opazovano ključno sliko, t.j.,

$$dist(\boldsymbol{\xi_{ji}}) = \boldsymbol{\xi_{ji}}^T \mathbf{W} \boldsymbol{\xi_{ji}}. \tag{2.20}$$

Pri tem diagonalna matrika $\mathbf{W}$ vsebuje ustrezne uteži.

Metoda novi ključni sliki inicializira globinsko sliko tako, da v njo projecira točke iz globinske slike najprimernejše ključne slike. Na novi globinski sliki nato popravi koordinate globinskih točk (angl. spatial regularization) in izloči nepravilno postavljene točke oz. osamelce (angl. outliers). Nato se globinska slika skalira, da si postavi povprečno inverzno globino (angl. inverse depth) na ena. Faktor skaliranja se vključi v matriko kamere, kjer je shranjen njen relativen položaj v prostoru, njena rotacija in sedaj tudi skala. Končno se nova ključna slika lahko uporablja za sledenje položaju prihodnjih slik. Za več informacij usmerjamo bralca v Podpoglavje 2.4.1.

Monokularni SLAM-i imajo pomanjkljivost, da ne morejo izmeriti absolutne skale - razmerja med izračunanimi razdaljami in realnimi razdaljami. Pri tem je problem, ker se skozi čas skala (zaradi napak pri meritvah) spreminja. Ta pojav je znan kot drsenje skale (angl. scale-drift), ki predstavlja enega glavnih izvorov napak pri meritvah [31]. LSD-SLAM rešuje ta problem s primerjavo faktorjev skaliranja (ki ju je izračunal ob generiranju ključnih slik) uporabljenih ključnih slik pri zapiranju zank (angl. loop-closure) v zemljevidu. Tako metoda detektira nepravilnosti drsenja skale in napako lahko odpravi.

LSD-SLAM ima mehanizem zapiranja zanke realiziran z lastno metodo neposrednega prileganja slik, ki upošteva pojav drsenja skale. Ta primerja in prilega dani ključni sliki, ki imata različne faktorje skaliranja. Da bi lahko ocenila skalirano transformacijo med ključnima slikama, upošteva poleg fotometričnega ostanka r_p (angl. photometric residual) tudi globinski ostanek

r_d (angl. depth residual), ki kaznuje razlike inverznih globin. Za oceno parametrov transformacije med slikami i in j, $\boldsymbol{\xi_{ji}}$, minimizira funkcijo

$$E_p(\boldsymbol{\xi_{ji}}) = \sum_{\mathbf{p}\in\Omega_{D_i}} \left\| \frac{r_p^2(\mathbf{p}, \boldsymbol{\xi_{ji}})}{\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi_{ji}})}} + \frac{r_d^2(\mathbf{p}, \boldsymbol{\xi_{ji}})}{\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi_{ji}})}} \right\|_\delta, \tag{2.21}$$

kjer sta r_p^2 in $\sigma^2_{r_p}$ definirana kot v enačbah (2.17) in (2.18). Naslednji formuli pa definirata globinski ostanek in njegovo varianco,

$$r_d(\mathbf{p}, \boldsymbol{\xi_{ji}}) = [\mathbf{p}']_3 - D_j([\mathbf{p}']_{1,2}), \tag{2.22}$$

$$\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi_{ji}})} = V_j([\mathbf{p}']_{1,2}) \left(\frac{\partial r_d(\mathbf{p}, \boldsymbol{\xi_{ji}})}{\partial D_j([\mathbf{p}']_{1,2})} \right)^2 + V_i(\mathbf{p}) \left(\frac{\partial r_d(\mathbf{p}, \boldsymbol{\xi_{ji}})}{\partial D_i(\mathbf{p})} \right)^2, \tag{2.23}$$

kjer $\mathbf{p}' = \omega_s(\mathbf{p}, D_i(\mathbf{p}), \boldsymbol{\xi_{ji}})$ predstavlja transformirano točko.

Optimizacija zemljevida

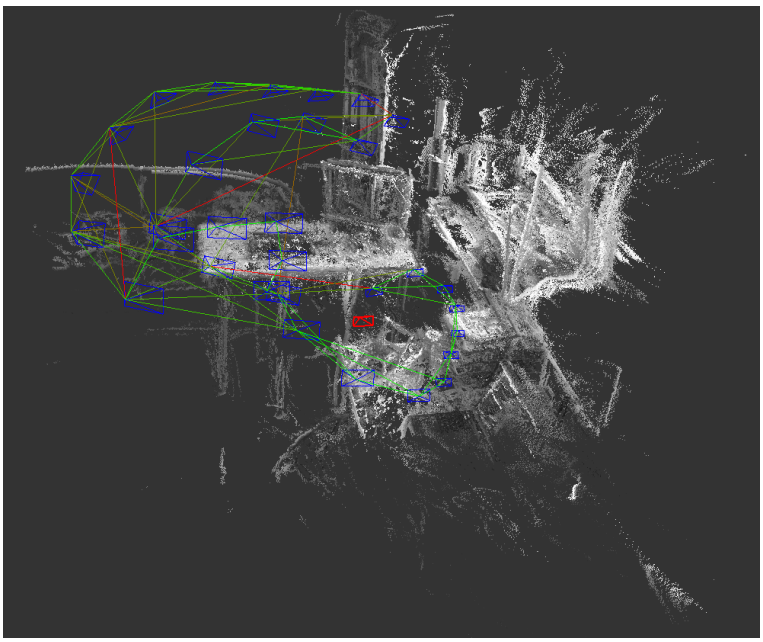
3D zemljevid je sestavljen iz skupine ključnih slik, povezav med njimi in njihovih globinskih slik. Zemljevid se skozi čas optimizira oz. posodablja. Na Sliki 2.5 so prikazane omenjene komponente. Bele pike sestavljajo oblak točk (angl. pointcloud), ki so postavljene na vseh površinah, za katere je LSD-SLAM uspel izračunati pozicijo. Ko se je kamera premikala skozi prostor, so se ustvarjale nove ključne slike. Poza vsake ključne slike je označena z modrim kvadratkom. LSD-SLAM je računal globine predmetom v prostoru na podlagi disparitet med ključnimi slikami, ki so med sabo povezane s črto.

Ko je v danem koraku za računanje globine izbrana ključna slika in je globinska slika že posodobljena, se proži komponenta optimizacije zemljevida. Definirajmo konkatenacijski operator poz $\circ$ z naslednjim izrazom:

$$\boldsymbol{\xi_{ki}} = \boldsymbol{\xi_{kj}} \circ \boldsymbol{\xi_{ji}} = log(exp(\boldsymbol{\xi_{kj}}) \cdot exp(\boldsymbol{\xi_{ji}})) \tag{2.24}$$

V tem koraku se minimizira funkcijo napake

$$E(\boldsymbol{\xi_{W1}} \ldots \boldsymbol{\xi_{Wn}}) = \sum_{(\boldsymbol{\xi_{ji}}, \Sigma_{ji})\in\varepsilon} (\boldsymbol{\xi_{ji}} \circ \boldsymbol{\xi_{Wi}}^{-1} \circ \boldsymbol{\xi_{Wj}})^T \Sigma_{ji}^{-1} (\boldsymbol{\xi_{ji}} \circ \boldsymbol{\xi_{Wi}}^{-1} \circ \boldsymbol{\xi_{Wj}}), \tag{2.25}$$



Slika 2.5: LSD-SLAM-ova rekonstrukcija prostora s ključnimi slikami (modri kvadrati) in povezavami med njimi. Rdeč kvadrat predstavlja trenutno lokacijo kamere.

kjer je W svetovni okvir (angl. world frame).

Metoda LSD-SLAM v koraku oprimizacije zemljevida poleg uporabe mehanizmov za zapiranje zank tudi detektira nepravilnosti drsenja skale in napake odpravi [6].

Inicializacija

Ko se LSD-SLAM inicializira, se prvi ključni sliki pripravi globinsko sliko z naključnimi globinskimi vrednostmi in veliko varianco. V prvih sekundah LSD-SLAM zaklene konfiguracijo globin. Če se v tem času dovolj premika kamero, LSD-SLAM s propagiranjem globinskih slik prihodnjih ključnih slik konvergira k pravilni konfiguraciji globin.

2.5 Stereofonski zvok

Če človek nosi slušalke, s katerimi sliši vsako uho svoj kanal generiranega zvoka, mu lahko induciramo iluzijo, da generiran zvok prihaja iz določene smeri. S 3D zvočnimi efekti lahko postavimo izvor zvoka kamorkoli v navideznem tridimenzionalnem svetu, saj je človek sposoben triangulirati lokacijo izvora zvoka. Pri tem sodelujejo zunanje, srednje in notranje uho ter možgani, ki ocenijo lokacijo izvora zvoka [2]. Ker se oblike glave in ušes pri ljudeh razlikujejo, ljudje zaznavajo izvore generiranih zvokov z različno natančnostjo.

Pri simuliranju 3D zvokov se predpostavlja, da so zvočni signali, ki pridejo do človekovih bobničev, enaki v primerih, ko človek sliši zvok iz okolice in iz slušalk [35]. Za dosego efekta eksternalizacije zvoka - da človek dojame, da zvok izvira zunaj glave - se uporablja metodo, opisano v navedeni strokovni literaturi [35].

Naj bo na zvočnik vezan električni signal $\mathbf{x_1}(t)$, na slušalke vezan signal $\mathbf{x_2}(t)$, signala $\mathbf{y_1}(t)$ in $\mathbf{y_2}(t)$ pa nastaneta ob prejemanju zvoka prek mikrofona, postavljena na mesto bobniča v človekovem ušesu, kjer je $\mathbf{y_1}(t)$ podvržen zvoku zvočnika, $\mathbf{y_2}(t)$ pa zvoku slušalk. Če želimo ustvariti navi-

dezno lokaliziran zvok, moramo izbrati tak $\mathbf{x_2}(t)$, da bo veljala enačba

$$\mathbf{y_2}(t) = \mathbf{y_1}(t). \tag{2.26}$$

Cilj je narediti linearni filter, ki bi transformiral vhodni signal $\mathbf{x_1}(t)$, tako da bi ga poslušalec s slušalkami slišal enako kot signal $\mathbf{x_2}(t)$ izvirajoč iz okolice.

Po procesiranju signalov $\mathbf{x_1}(t)$, $\mathbf{x_2}(t)$, $\mathbf{y_1}(t)$ in $\mathbf{y_2}(t)$ s Fourierovo transformacijo [34] dobimo rezultate $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{Y_1}$ in $\mathbf{Y_2}$, ki jih uporabljamo v enačbah

$$\mathbf{Y_1} = \mathbf{X_1LFM}, \tag{2.27}$$

$$\mathbf{Y_2} = \mathbf{X_2HM}, \tag{2.28}$$

kjer je $\mathbf{L}$ prenosna funkcija (angl. transfer function) zvočnika v odprtem prostoru, $\mathbf{F}$ predstavlja funkcijo prenosa zvoka glede na človekovo glavo, $\mathbf{M}$ je prenosna funkcija mikrofona, $\mathbf{H}$ pa prenosna funkcija, ki predstavlja relacijo med glavo in ušesom. Če se nastavi $\mathbf{Y_1} = \mathbf{Y_2}$ se lahko izračuna

$$\mathbf{X_2} = \mathbf{X_1LF/H}. \tag{2.29}$$

Tako dobimo splošno prenosno funkcijo $\mathbf{T} = \mathbf{LF/H}$. Iz tega sledi, da je signal $\mathbf{x_1}(t)$ po omenjenem procesiranju enak signalu $\mathbf{x_2}(t)$, torej bi nov signal človeku zvenel enako na stereo slušalkah, kot če bi originalen izviral iz okolice. Postopek se izvede za vsako uho po sebej.

Potrebno je poudariti, da je omenjen pristop zgolj teoretičen. Raziskave kažejo, da s to metodo slušalke ne popolnoma simulirajo zvoka, kot bi se ta slišal iz okolice. Eksperimenti, izvedeni v delu [35], kažejo, da je mogoče dobro razbrati pozicijo izvora zvoka po horizontali, po vertikali pa je to težje.

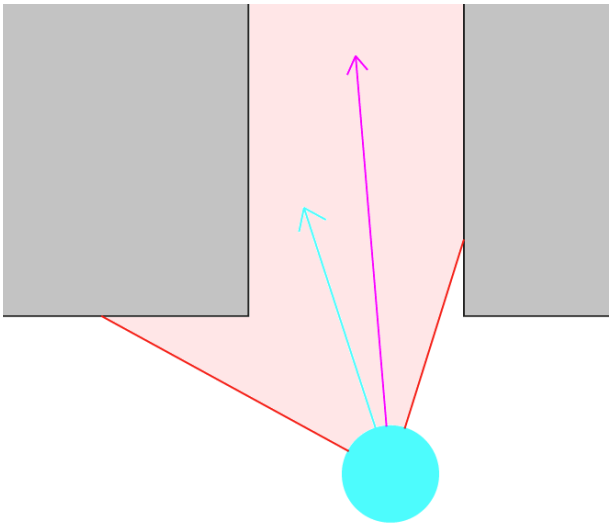
Poglavje 3

Sistem za izogibanje oviram

V okviru diplomskega dela smo razvili sistem za pomoč slepim in slabovidnim pri orientaciji in navigaciji v nestrukturiranem in nenadzorovanem okolju. Sistem s pomočjo kamere opazuje prostor, slike v realnem času obdeluje, da pridobi informacije o oddaljenosti videnih predmetov, in na podlagi vseh informacij izračuna najbolj obetavno pot za človeka, ki drži kamero v roki. Da bi lahko slep človek uporabljal tak sistem, smo povratno informacijo realizirali v obliki tridimenzionalnega zvoka, kateremu se izvor postavi na določeni razdalji v smeri, ki jo je sistem izbral. Uporabnik mora uporabljati slušalke, da lahko razloči smer, iz katere prihaja navidezni zvok.

3.1 Algoritem za izogibanje oviram

Zgradili smo algoritem za izogibanje oviram, ki uporablja delno rekonstrukcijo okolja, katero proizvede metoda LSD-SLAM [6] (Poglavje 2.4). Naloga algoritma je, da najde najbolj obetavno smer, kamor se lahko uporabnik premakne in da ga o tem obvesti. Pri tem ne upošteva neposredno uporabnikove želje, kam hoče iti. Zato predpostavlja, da uporabnik približno ve, kam je usmerjen in kam je namenjen. Npr. če se uporabnik znajde na začetku hodnika, mora vedeti, da je tam hodnik. Ker je slepemu človeku težko vedeti, če je pravilno usmerjen (vektor gledanja vzporeden s stenami hodnika),



Slika 3.1: Moder krog s puščico predstavlja uporabnika sistema in njegovo usmerjenost. Sistem predmetom v vidnem polju, označenim z rdečo barvo, meri oddaljenost in na podlagi te računa najbolj obetavno smer za uporabnika, označeno z vijolčno barvo.

mu lahko sistem pride zelo prav, saj mu sporoči pravilno usmerjenost (Slika 3.1). Uporabnik mora le usmeriti kamero tako, da je v njenem vidnem polju nadpovprečno oddaljen del prostora, kamor se želi premakniti. Ob tem sistem upošteva tudi ovire. Če se pojavi ovira pred uporabnikom, bo sistem zaznal, da obstaja bolj obetavna pot mimo ovire in bo novo smer posredoval uporabniku. Omejitve sistema postanejo očitne, ko najbolj oddaljene regije v vidnem polju kamere predstavljajo smer, ki ni primerna za uporabnikovo pot. Sistem ga bo usmeril mimo zaznanih ovir, ampak ostale površine so lahko za hojo vprašljive.

Algoritem za vhod vzame globinsko sliko, ki predstavlja oddaljenost predmetov v vidnem polju kamere. Format globinske slike je isti, kot jo uporablja metoda LSD-SLAM. Podatke o lokaciji in usmerjenosti kamere algoritem ne upošteva, saj v posameznem trenutku niso relevantni za določanje trenutno najbolj obetavne poti. Za sistem je dovolj, da v posameznem trenutku po-

zna oddaljenost predmetov v vidnem polju kamere, saj je smer izračunane najbolj obetavne poti relativna na uporabnikovo smer gledanja. Algoritem bi sicer lahko tudi nadgradili s pomočjo podatkov o lokaciji in usmerjenosti kamere. Nadgradnjam je namenjeno Poglavje 5.1.

Človekovo okolje je mogoče preslikati v dvodimenzionalni prostor, saj se z nogami vedno drži tal. Zato lahko višinsko komponento jemljemo kot konstanto. S to poenostavitvijo v mislih smo za uporabnika predvideli le horizontalne smeri kot možnosti za nadaljevanje poti. Algoritem opazuje globinsko sliko samo po horizontali, v okolici njene sredine. Z upoštevanjem zgornje in spodnje meje tako ne upošteva predmetov, ki so pod nogami uporabnika (npr. tla), in predmetov, ki so nad uporabnikom (npr. strop). Ta dejstva nakazujejo na naslednjo predpostavko sistema, da uporabnik drži kamero vzporedno s tlemi (oz. usmerjeno enako kot ima usmerjeno glavo, odvisno tudi od uporabnikove želje). Pomanjkljivost tega pristopa se izrazi, ko se uporabnik približa oviri, ta pa izstopi iz vidnega polja kamere. V tem primeru se ovira ne upošteva pri računanju poti, zaradi česar lahko pride do trka med uporabnikom in oviro.

V nadaljevanju algoritem po horizontali razdeli obravnavan del globinske slike $\boldsymbol{\Delta}$ na N delov in za vsakega preveri, katera globinska vrednost je v tem kvadrantu najmanjša. S tem pridobi enodimenzionalni vektor globin $\boldsymbol{\delta}$ nižje resolucije (dolžine N), ki se ga v nadaljevanju hitreje obravnava kot bi se celotni nabor globinskih vrednosti. Poleg tega vsak element vektorja $\boldsymbol{\delta}$ vsebuje informacijo, kako oddaljen je najbližji predmet v svojem delu globinske slike v celotnem pasu, od vrhnje zastavljene meje a do spodnje b. Vrednosti globinskemu vektorju $\boldsymbol{\delta}$ se dodeljuje z

$$\boldsymbol{\delta}_i = \min_{x_{\min} \leq x < x_{\max}} (\min_{b<y<a} (\boldsymbol{\Delta}_{x,y})), \tag{3.1}$$

kjer se $x_{\min}$ in $x_{\max}$ računa z

$$x_{\min} = w \cdot \frac{i}{N}, \tag{3.2}$$

$$x_{\max} = w \cdot \frac{i+1}{N}, \tag{3.3}$$

pri čemer w predstavlja širino matrike globinske slike. Ker za vsak element vemo, kje na sliki se nahaja, si lahko s to informacijo pomagamo pri obveščanju uporabnika o oddaljenosti predmetov pred sabo (Slika 3.3).

Srednja vrednost μ, ki jo algoritem izračuna za vektor N-tih globin, predstavlja prag za ločevanje bolj oddaljenih predmetov od bližnjih (Slika 3.2). Srednjo vrednost μ računa po formuli

$$\mu = \frac{\sum_{i=0}^{N} \boldsymbol{\delta}_i}{N}. \tag{3.4}$$

Pri tem se ne upoštevajo točke, katerim se globina pri meritvi ni spremenila oz. je ostala enaka inicializirani. Za takšne točke algoritem ne pozna globine, saj je ni mogel izračunati. Nedefinirane vrednosti globin se pojavljajo, ko so v vidnem polju kamere homogene površine, ki nimajo dovolj vizualnih informacij (npr. robovi), da bi jim program lahko sledil in s tem ocenil globino. Algoritem za izogibanje oviram se tako izogne nevšečnim rezultatom in nikoli ne usmerja uporabnika v predele prostora, za katere ne pozna oddaljenosti.

V naslednji fazi se algoritem spet sprehodi globinskem vektorju $\boldsymbol{\delta}$. Tokrat, na podlagi oddaljenosti od kamere, elemente $\boldsymbol{\delta}_i$ kategorizira med nadpovprečno oddaljene in ostale po formuli

$$I(\boldsymbol{\delta}_i) = \begin{cases} 1 & \text{če je } \boldsymbol{\delta}_i > \mu, \\ 0 & \text{sicer.} \end{cases} \tag{3.5}$$

Algoritem poišče najdaljše zaporedje nadpovprečno oddaljenih elementov ($I(\boldsymbol{\delta}_i) = 1$). Ko ga najde, mu izračuna indeks sredinskega elementa

$$\mu_I = i_z - \frac{n_I}{2}, \tag{3.6}$$

kjer je i_z indeks zadnjega elementa v zaporedju, n_I pa predstavlja dolžino najdaljšega zaporedja nadpovprečno oddaljenih elementov. Sredina zaporedja μ_I posredno predstavlja izbrano smer

$$s = \frac{N}{2} - \mu_I. \tag{3.7}$$

Izhodišče globinske slike $\boldsymbol{\Delta}$ se nahaja v zgornjem levem kotu, zato negativne vrednosti smeri s pomenijo, da mora uporabnik spremeniti smer v levo, ob

pozitivnih vrednostih smeri s pa mora uporabnik spremeniti smer v desno. Postopek je opisan v Algoritmu 1.

Generira se zvok, katerega se postavi na določeno oddaljenost od kamere v izbrani smeri. Koordinate se mu določi s pomočjo pomožnega vektorja $\mathbf{v_{tmp}}$, ki kaže v smer, od koder prihaja zvok. Ta je definiran kot

$$\mathbf{v_{tmp}} = \begin{pmatrix} (x' \cdot f_{xi} + c_{xi}) \cdot \boldsymbol{\delta}_{\mu_I} \\ 0 \\ \boldsymbol{\delta}_{\mu_I} \end{pmatrix}, \tag{3.8}$$

kjer je smer v enoti optičnih pikslov definirana kot

$$x' = \frac{2 \cdot s}{N} \cdot w, \tag{3.9}$$

in veljajo enačbe inverznih lastnosti kamere

$$f_{xi} = 1/f_x, \tag{3.10}$$

$$f_{yi} = 1/f_y, \tag{3.11}$$

$$c_{xi} = -c_x/f_x, \tag{3.12}$$

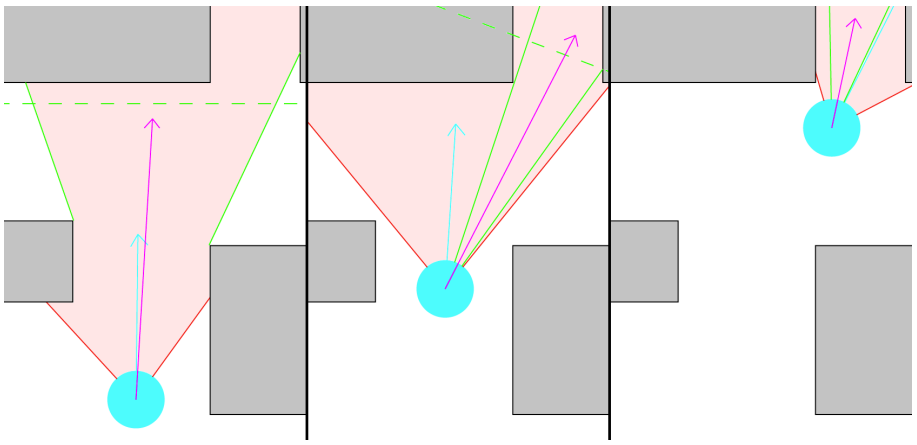
$$c_{yi} = -c_y/f_y, \tag{3.13}$$

pri čemer se uporablja dano fokusno razdaljo kamere po osi x f_x in y f_y ter optični sredini po osi x c_x in y c_y. Pomožni vektor $\mathbf{v_{tmp}}$ normaliziramo in skaliramo z določenim radiusom r, da dobimo pozicijo izvora zvoka

$$\mathbf{v_s} = \frac{\mathbf{v_{tmp}}}{\|\mathbf{v_{tmp}}\|} \cdot r. \tag{3.14}$$

S pomočjo stereo zvoka sistem uporabniku sporoči izbrano smer. Uporabnik nato oceni lego izvora zvoka in se lahko začne premikat v ustrezni smeri. V primeru negotovosti pa si lahko pomaga z obračanjem oz. premikanjem in s tem potrjuje konsistenco lege izvora zvoka.

Naš pristop je kljub reduciranem naboru podatkov, na podlagi katerih se odloča za smer, precej učinkovit. Uporabnika vedno usmerja v največje proste predele prostora, ki so nadpovprečno oddaljeni. Če v nekem trenutku nek prehod zaradi praga še ni viden, bo zaznan pa čez nekaj korakov, ko



Slika 3.2: Na sliki je prikazan primer uporabe sistema, kjer uporabnik sistema (modri krogec, usmerjen kamor kaže modra puščica) hodi skozi prostor. Sistem v vidnem polju (rdeče pobarvana regija) zaznava ovire in jim računa oddaljenost. Nato izračuna njihovo povprečje, ki predstavlja prag ločevanja bližnjih od oddaljenih predmetov, označen z zeleno prekinjeno črto. Metoda za nadaljevanje poti izbere tisto smer, ki predstavlja sredino nadpovprečno največje zaznane oddaljene regije. Izbrana smer je označena z vijolčno puščico.

se mu bo uporabnik približal (Slika 3.2). Ob upoštevanju predpostavke, da uporabnik pozna približno smer, mu jo bo sistem pravilno popravljal. Uporabnik lahko uporabi sistem tudi za ugotavljanje, kje se nahajajo prehodi iz prostora v prostor samo s tem, da s kamero pogleda okrog sebe. Paziti mora le, da metoda LSD-SLAM uspešno posodablja svoj zemljevid prostora.

Alternativa opisani metodi bi bila, da bi sistem pošiljal uporabnika v smer najbolj oddaljene zaznane regije. V tem primeru bi lahko nekonsistenca (šum na podatkih) lege izvora zvoka zmedla uporabnika. Poleg tega bi pogosto prihajalo do primerov, ko bi sistem pošiljal uporabnika tik mimo zaznanih ovir do najbolj oddaljene zaznane točke, pri čemer bi se uporabnik z veliko verjetnostjo zaletel v bližnji predmet.

Algoritem 1 Izogibanje oviram

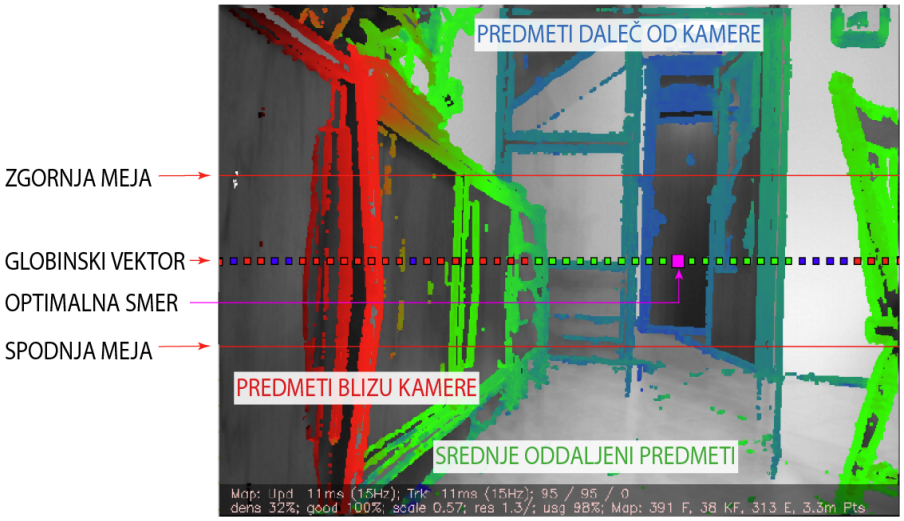
Vhodi: Vektor *najblizjeGlobine*, ki hrani N globin po horizontali.

Izhod: Lokalna koordinata izbrane poti X.

```
1: prejsnjiZelen ← false
2: zaporednih ← 1
3: najvecOddaljZaporednih ← 0
4: najboljsiKvadrant ← −1
5: for all globina in najblizjeGlobine do
6:     if globina ≠ neskoncno then
7:         if globina > depthAvg then
8:             if prejsnjiZelen then
9:                 zaporednih ← zaporednih + 1
10:            else
11:                zaporednih ← 1
12:                prejsnjiZelen ← false
13:        else
14:            if ¬prejsnjiZelen then
15:                zaporednih ← zaporednih + 1
16:            else
17:                if zaporednih > najvecOddaljZaporednih then
18:                    najvecOddaljZaporednih ← zaporednih
19:                    najboljsiKvadrant ← globina.id − zaporednih/2
20:                zaporednih ← 1
21:                prejsnjiZelen ← false
22:    else
23:        if prejsnjiZelen and zaporednih > najvecOddaljZaporednih then
24:            najvecOddaljZaporednih ← zaporednih
25:            najboljsiKvadrant ← globina.id − zaporednih/2
26:        zaporednih ← 1
27: if prejsnjiZelen and zaporednih > najvecOddaljZaporednih then
28:    najvecOddaljZaporednih ← zaporednih
29:    najboljsiKvadrant ← globina.id − zaporednih/2
30:    zaporednih ← 1
31: if najboljsiKvadrant ≠ −1 then
32:    X = najboljsiKvadrant · imgWidth/(N − 1)
```

3.2 Vizualizacija delovanja

Za lažji razvoj sistema ter vizualno analizo delovanja algoritma smo LSD-SLAM-ovo vizualizacijo nadgradili. Vizualno smo preverjali, katero smer je sistem izbral. S tem smo tudi preverjali pravilnost postavitve 3D zvoka. Kot je prikazano na Sliki 3.3, smo zgornjo in spodnjo mejo obravnavane površine označili z rdečo črto. Podatke o najmanjši oddaljenosti predmetov za vsak kvadrant smo prikazali z barvnim kvadratkom na ustreznem mestu. Z majhnimi kvadratki zelene barve smo označili kvadrante, kjer so najmanjše globine nadpovprečno velike. Tisti kvadrant, ki velja za najprimernejšega za določitev smeri za uporabnika, smo označili z večjim kvadratkom vijolične barve. V tej smeri smo tudi postavili izvor zvoka. Modri kvadratki predstavljajo regije, za katere sistem ne pozna globine in je zato ne upošteva. Ostale kvadrante pa smo označili z majhnimi rdečimi kvadratki. Tem je skupno, da nimajo nadpovprečnih globin, torej predstavljajo regije, kjer se nahajajo bližje ležeči predmeti.



Slika 3.3: Vizualizacija globinskega vektorja, ki hrani najbližjo globino vsakega kvadranta. Vijoličen kvadratek predstavlja najbolj obetavno pot, ki v tem primeru kaže skozi vrata. Kvadratki na sliki označujejo center posameznega kvadranta, ki sega od zgornje meje do spodnje.

Poglavje 4

Eksperimentalno vrednotenje

Na začetku pričujočega poglavja opisujemo, kako smo implementirali posamezne dele sistema. Sledi opis postopka kalibracije kamere, s čimer smo dosegli boljše delovanje metode LSD-SLAM [6] opisane v Poglavju 2.4. Po tem opisujemo, kako se metoda LSD-SLAM obnaša v praksi, nakar predstavimo izvedene meritve natančnosti sistema z rezultati. Poglavje se konča z opisom izvedenih testov za določanje učinkovitosti sistema.

4.1 Implementacija sestavnih delov sistema

Sistem smo razvijali v okolju Linux. Za komunikacijo med deli sistema smo uporabljali robotski operacijski sistem (anlg. robot operating system, ROS) [28]. Tako smo se odločili zaradi uporabe metode LSD-SLAM, ki je na razvijalčevi spletni strani na voljo le v izbranem okolju. Podrobnejši opis implementacije sestavnih delov sistema sledi v nadaljevanju.

4.1.1 Zajem slik

Vizualno informacijo dobi sistem iz monokularne kamere. Ta odločitev je primerna zaradi praktičnih in finančnih razlogov, saj bi se lahko v končni fazi sistem v celoti poganjal na pametnem telefonu. Poleg tega so na mobilnih telefonih kamere precej kvalitetne, kar je tudi razlog, da smo to za

naš sistem uporabili. Na pametni telefon smo naložili aplikacijo IP Webcam [18], ki zajema slike iz kamere, odpre izbrana vrata (angl. port) in na njih oglašuje slike. Pri tem smo uporabili mobilni telefon LG G3, katerega poganja procesor Quad-core 2.5 GHz Krait 400. Kamera telefona omogoča slikanje z največjo ločljivostjo 4160×3120 slikovnih točk, video posnetki pa imajo lahko največjo ločljivost 3840×2160 pri frekvenci osveževanja 30 Hz.

Na računalniku, povezanem v isto lokalno omrežje kot telefon, se je izvajal program, ki se je povezal na odprta vrata aplikacije IP Webcam na pametnem telefonu in prenašal slike v lokalni pomnilnik ter jih prek ROS-ovih funkcionalnosti za sporočanje oglaševal na določen topic. Slike so se prenašale s hitrostjo približno 15 slik na sekundo. Kamero smo vedno premikali počasi, da nadaljnje metode niso imele težav zaradi nizke frekvence osveževanja slike. Na ta način smo v pravilnem formatu pripravili okno v svet za metodo LSD-SLAM.

4.1.2 Izračun globinske slike

Za računanje razdalij do predmetov v vidnem polju kamere smo potrebovali metodo, ki zmore to početi v realnem času. Da bi bil sistem primeren za čim različnejša okolja, tako za notranjo kot tudi zunanjo uporabo, in bi se izvajal v realnem času, smo se odločili za metodo imenovano Large-Scale Direct monocular SLAM (v nadaljevanju LSD-SLAM) [6, 7] opisano v Poglavju 2.4. Metoda deluje po principu pol-goste (angl. semi-dense) vizualne odometrije. Pristop kombinira točnost in robustnost gostega vizualnega SLAM-a ter učinkovitost metod, ki bazirajo na sledenju značilnih točk. Metoda LSD-SLAM je procesorsko zelo učinkovita. Poganja jo lahko tudi procesor pametnega telefona [30]. Metoda LSD-SLAM uporablja sekvenco slik iz kamere za računanje oddaljenosti predmetov v vidnem polju. S pomočjo grafa ključnih slik, za katere pozna položaj v prostoru, rotacijo, skalo in globinsko sliko (depth map), gradi in izpopolnjuje 3D zemljevid prostora. Zaradi robustnosti metode in uporabljenih pristopov za korekcijo napak, lahko metoda LSD-SLAM gradi zemljevide velikega obsega. Delovanje metode LSD-SLAM

je natančneje opisano v Poglavju 2.4.

4.1.3 Algoritem za izogibanje oviram

Da bi sistem vedel, kako usmerjati uporabnika, ne da bi se ta zaletel, smo razvili algoritem, ki skrbi za izogibanje oviram. Kot je opisano v Poglavju 3, algoritem za izogibanje oviram v slehernem trenutku analizira globinsko sliko zadnje ključne slike.

Metodo LSD-SLAM smo nastavili tako, da v notranem prostoru našega laboratorija kreira nove ključne slike, ko se kamera premakne za več kot 7 cm oz. rotira za več kot 5 stopinj. S tem smo dosegli, da algoritem za izogibanje oviram obravnava globinske podatke s frekvenco približno 5 Hz, kar je v praksi dovolj pogosto za počasi premikajočega uporabnika.

Algoritem globinsko sliko razdeli po horizontali na $N = 50$ delov, torej se ustvari vektor globin s petdesetimi elementi. Tako imamo dovolj gosto posejane meritve globin, da se med obračanjem kamere uporabnik ne zmede zaradi preskoka smeri, iz katere prihaja umetno generiran 3D zvok. Hkrati pa je dolžina vektorja globin dovolj majhna, da se ob procesiranju njegovih podatkov ne izgubi veliko časa.

Ko ima algoritem izračunano najbolj obetavno smer za uporabnika, mu to sporoči prek zvočnega signala. Za generiranje 3D zvoka smo se poslužili knjižnice imenovane Simple and fast multimedia library (SFML) [11]. Ta na podlagi določitve položaja za izvor zvoka le-temu izračuna ustrezen zvočni signal za vsak stereo kanal. Pri tem upošteva smer, od kod prihaja zvok glede na usmerjenost kamere in oddaljenost izvora zvoka od kamere. Uporabnik sliši vsak kanal v ustreznem ušesu in ugotovi, od kod prihaja zvok.

Ko se uporabnik premika je optimalno, da drži kamero pred sabo na določeni razdalji. To je pomembno, ker z ustrezno oddaljenostjo kamere od telesa robustificiramo delovanje metode LSD-SLAM, saj onemogočamo čisto vrtenje kamere. Če se človek obrne z iztegnjeno roko naprej, se bo tej v veliki meri spremenil položaj, ne pa samo usmeritev. Poleg tega mora biti kamera usmerjena enako kot glava uporabnika, ker pri generiranju zvoka se

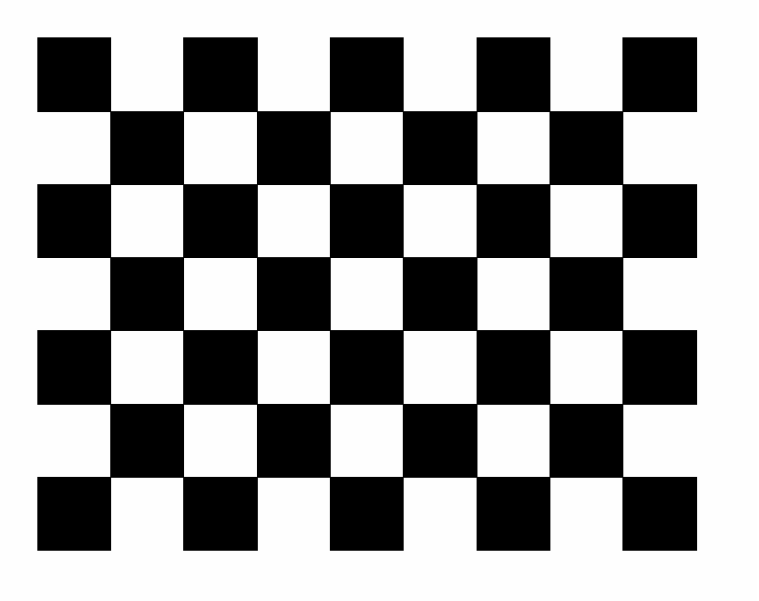
upošteva usmerjenost telefona, ne pa njegove glave. Če uporabnik pravilno drži telefon, ima dobre pogoje za pravilno oceniti smer izvora generiranega zvoka. Naučiti se mora le, kako se odzvat na spremembe smeri zvoka. Najbolj učinkovito je, da se uporabnik za zvokom vedno obrača med hojo naprej, ker sistem lahko spremeni smer takoj, ko bližnja ovira izstopi iz vidnega polja kamere.

4.2 Kalibracija sistema

Metoda LSD-SLAM je v dobrih pogojih lahko zelo natančna pri gradnji zemljevida, zato potrebuje kalibracijske podatke o uporabljeni kameri. Da bi metoda LSD-SLAM delovala optimalno, smo kameri določili, da zajema slike velikosti 720×480 slikovnih elementov.

Za kalibracijo kamere smo uporabili ROS-ov paket *camera calibration* [15], ki predstavlja orodje, s katerim je mogoče na enostaven način izmeriti kalibracijske podatke o kameri. Kalibracijski program prek kamere opazuje šahovnico, ki ima določeno število ($m \times n$) presečišč kvadratov in določeno velikost (d) kvadratkov. Kot vhod prejme podatka o velikosti šahovnice in ime teme, kjer so objavljene slike. V našem primeru smo uporabili šahovnico, ki je imela $m = 8$ presečišč kvadratkov po širini in $n = 6$ presečišč kvadratkov po višini (Slika 4.1). Da je lahko bila prikazana na zaslonu, smo velikost kvadratkov določili z $d = 3.8$ cm.

S pomočjo paketa *camera calibration* smo izmerili potrebovane lastnosti kamere: fokusna razdalja kamere po osi x ($f_x = 606.3$), fokusna razdalja kamere po osi y ($f_y = 605.7$), optična sredina kamere po osi x ($c_x = 357.1$), optična sredina kamere po osi y ($c_y = 239.7$) in koeficienti distorzije ($k_1 = 9.22 \cdot 10^{-2}$, $k_2 = -12.8 \cdot 10^{-2}$, $p_1 = 1.39 \cdot 10^{-3}$ in $p_2 = 5.58 \cdot 10^{-4}$). Fokusni razdalji in optični sredini se upošteva po odpravitvi radialne distorzije pri računanju preslikave 3D točk v 2D koordinatni sistem slike, kjer se uporablja



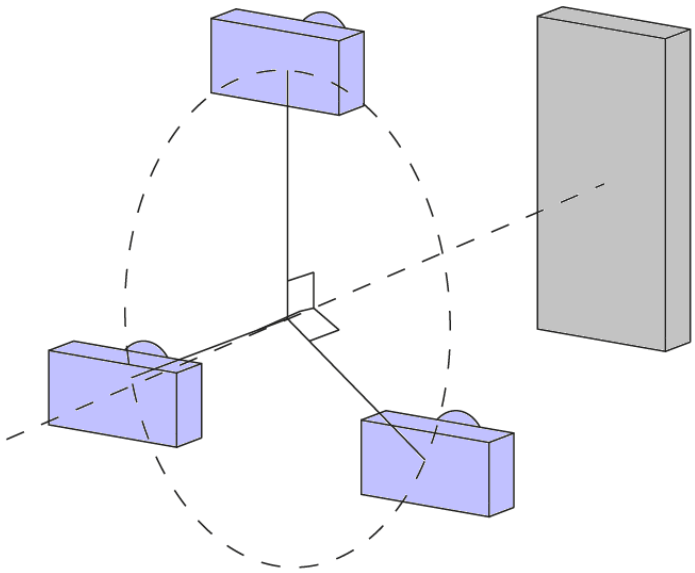
Slika 4.1: Šahovnica, ki smo jo uporabili za kalibracijo kamere.

kalibracijsko matriko

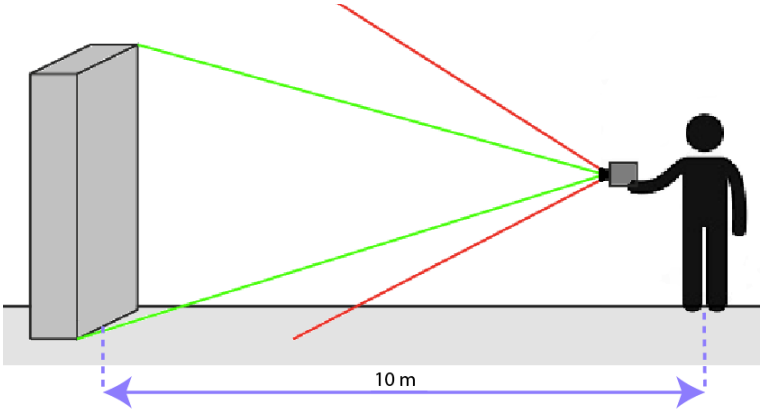
$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.1}$$

Izmerjene podatke smo prenesli v konfiguracijsko datoteko LSD-SLAM-a, ki jo ob zagonu prebere in upošteva pri delovanju.

Da je LSD-SLAM deloval pravilno, je bilo kljub kalibraciji pomembno, kako smo premikali kamero. Za dosego čim boljših rezultatov je potrebno metodi LSD-SLAM pripravljati slike iz primernih perspektiv, da lahko prek teh z najmanjšo napako računa razdalje. Najboljšo rekonstrukcijo dobimo takrat, ko je dispariteta očitna. To pa pomeni, da moramo premakniti kamero pravokotno na smer gledanja. Kamero premaknemo levo, desno, gor ali dol, lahko v krožnem gibu (Slika 4.2). Paziti pa moramo, da čim manj spremenimo smer gledanja kamere, saj vrtenje kamere okoli navpične ali vodoravne osi povzroča nedoločenost. Če kameri samo spremenimo smer gledanja, ne da bi jo premaknili, sistem zelo hitro odpove.



Slika 4.2: Slika prikazuje ilustracijo krožnega giba kamere, ko meri razdaljo do ovire.



Slika 4.3: Pri meritvi razdalje smo kamero držali vedno na isti razdalji od opazovanega ravnega predmeta, katerega sprednja površina je bila pravokotna na smer gledanja kamere. Za izračunat oddaljenost predmeta smo upoštevali le tisti del slike, ki je opisovala predmet. Ta del vidnega polja kamere je na skici označen z zeleno črto.

4.3 Meritev natančnosti sistema

Za oceno kvalitete ocenjevanja globine, smo testirali metodo LSD-SLAM z naslednjim postopkom (Slika 4.3):

- Predmet smo postavili 10 m stran od izhodiščne točke X_1, od koder smo izvedli meritev z LSD-SLAM-om. Razdaljo smo izmerili z metrom.

- S kamero smo naredili nekaj krožnih gibov, kot je opisano v Poglavju 4.2, s ciljem, da LSD-SLAM čim natančneje izmeri razdaljo do predmeta iz izhodiščne točke X_1. Pri vsaki meritvi smo upoštevali samo vrednosti globinske slike $\mathbf{\Delta}$, ki so opisovale opazovani predmet. Globinske vrednosti smo povprečili in s tem dobili eno meritev d_i - relativno razdaljo od kamere do opazovanega predmeta. Oceno resnične razdalje smo izračunali s povprečjem delnih meritev

$$\hat{d} = \frac{1}{N} \sum_{i=1:N} d_i, \tag{4.2}$$

kjer d_i predstavlja i-to meritev, N pa število ponovitev. V našem eksperimentu smo izvedli $N = 5$ meritev.

- Meritve smo povprečili. Skalo S smo izračunali po formuli

$$S = \frac{d_{real}}{\hat{d}}, \tag{4.3}$$

pri čemer je d_{real} izmerjena razdalja z metrom.

- Z ocenjeno skalo smo v metre preračunali vse N razdalje. Na teh razdaljah smo potem lahko izračunali varianco in povprečno vrednost. Izmerjene razdalje smo v metre pretvorili po formuli

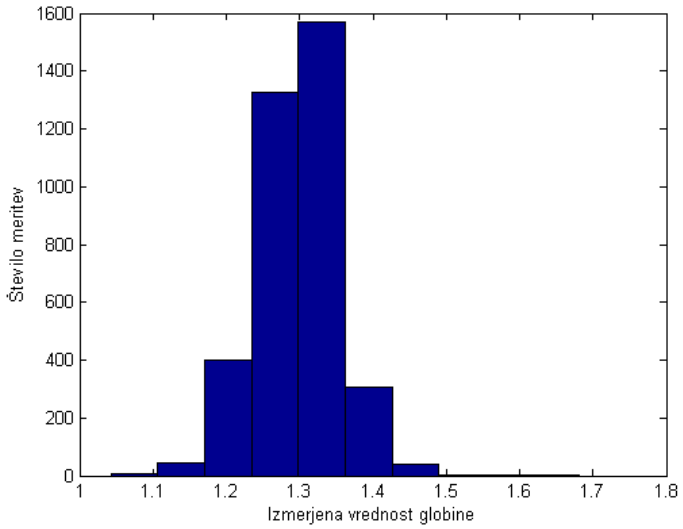
$$d_{Ri} = d_i \cdot S, \tag{4.4}$$

kjer d_{Ri} predstavlja meritev d_i pretvorjeno v metre.

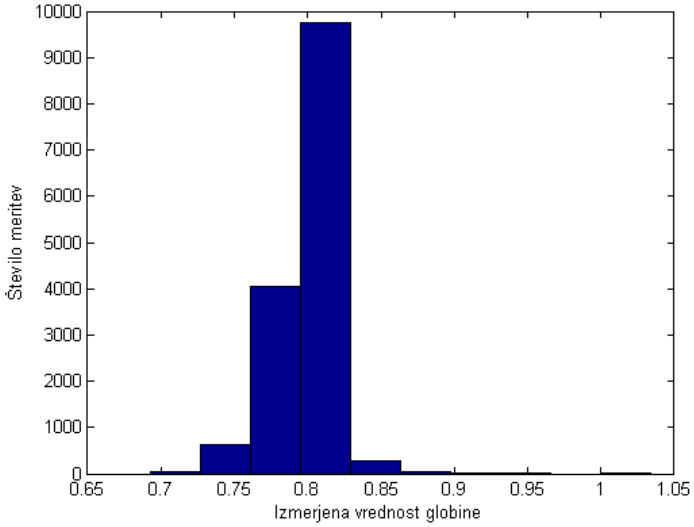
- Vse korake smo ponovili s postavitvijo predmeta na razdalje 5 m, 3 m, 1 m, 0.5 m in 0.25 m in rezultate prikazali v Tabeli 4.1.

Sliki 4.4 in 4.5 prikazujeta porazdelitev globin izbranih slikovnih elementov, na katerih je viden opazovan predmet. Opazimo lahko, da so globine porazdeljene približno po Gaussovi normalni porazdelitvi. Histogram na Sliki 4.4 je bil zgrajen na podlagi meritev predmeta na razdalji 10 m, histogram na Sliki 4.5 pa na podlagi meritev na razdalji 0.25 m. Razvidno je, da ima histogram na Sliki 4.5 veliko manjšo varianco, kar že kaže na nižjo napako pri meritvi.

Eksperiment je pokazal, da natančnost meritve z razdaljo pada. Graf na Sliki 4.6 prikazuje odvisnost standardne deviacije od razdalje na podlagi podatkov prikazanih v Tabeli 4.1. Tabela vsebuje podatke posameznih meritev za vse razdalje, njihove povprečne vrednosti in napako. Na grafu, prikazanem na Sliki 4.6, lahko opazimo, da je zaradi majhnega števila testov za posamezno razdaljo prišlo do anomalije, kjer so meritve za razdaljo 5 m bolj natančne kot meritve za razdaljo 3 m. Do tega je prišlo, ker je sistem zaznal več referenčnih točk v okolici na različnih globinah, kar ga je v primeru petih metrov bolje kalibriralo.



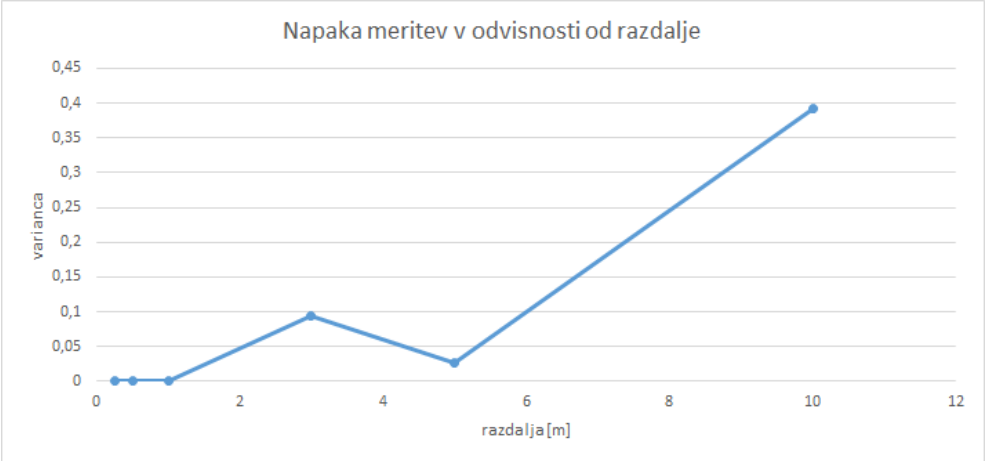
Slika 4.4: Porazdelitev izmerjenih globin izbranih slikovnih elementov na razdalji 10 m.



Slika 4.5: Porazdelitev izmerjenih globin izbranih slikovnih elementov na razdalji 0.25 m.

d_{real} [m]	d_1	d_2	d_3	d_4	d_5	$\hat{d}$	S
10	1,30	1,15	1,11	1,15	1,20	1,17	8,54
5	1,04	0,96	1,04	0,98	1,03	1,00	4,98
3	1,51	1,50	1,16	1,36	1,41	1,38	2,17
1	0,94	0,89	0,95	0,91	0,95	0,93	1,07
0,5	0,79	0,83	0,84	0,80	0,84	0,82	0,61
0,25	0,80	0,79	0,77	0,77	0,80	0,78	0,32
d_{real} [m]	d_{R1} [m]	d_{R2} [m]	d_{R3} [m]	d_{R4} [m]	d_{R5} [m]	$\bar{d}_R$ [m]	σ_R
10	11,08	9,83	9,44	9,79	10,21	10,07	0,626
5	5,15	4,80	5,17	4,90	5,10	5,02	0,165
3	3,27	3,25	2,52	2,94	3,06	3,01	0,306
1	1,00	0,96	1,02	0,98	1,02	1,00	0,028
0,5	0,48	0,51	0,51	0,49	0,51	0,50	0,015
0,25	0,26	0,25	0,25	0,25	0,25	0,25	0,005

Tabela 4.1: V zgornjem delu tabele so za vsako resnično razdaljo d_{real} prikazane razdalje posameznih relativnih meritev d_i, njihovo povprečje $\hat{d}$ in skala S. V spodnji polovici tabele pa so za zgornje meritve izračunane razdalje d_{Ri} v metrih, $\bar{d}_R$ in σ_R pa predstavljata njihovo povprečje in standardni odklon.



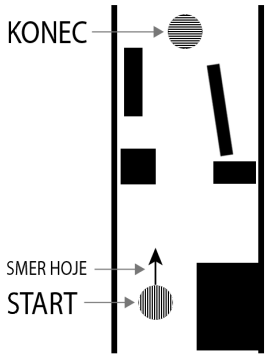
Slika 4.6: Na abscisni osi grafa so prikazane različne razdalje predmeta od kamere v metrih, na ordinatni osi pa varianca. Razvidno je, da od čim bliže kot sistem opazuje predmet, tem manjšo napako naredi pri ocenjevanju oddaljenosti predmeta.

4.4 Testiranje sistema za izogibanje oviram

Da bi ugotovili, kako uporaben je naš sistem za izogibanje oviram, smo pri testiranju primerjali uspešnost hoje po prostoru z uporabo sistema in brez njega. Subjekte z zavezanimi očmi smo postavili vedno na isto začetno točko, od koder so morali doseči določen cilj brez zadevanja v ovire. Za vsak poskus smo šteli, kolikokrat se je subjekt zaletel v ovire. Na koncu smo primerjali pogostost trkov subjektov, ki so uporabljali sistem, s pogostostjo trkov subjektov, ki sistema niso uporabljali. Postavili smo dva različna poligona. V poskusih so sodelovali štirje subjekti. Vsak poskus je posamezen subjekt ponovil osem krat s sistemom in osem-krat brez sistema za vsak poligon.

4.4.1 Priprava na testiranje

Ker metoda LSD-SLAM ne računa oddaljenosti homogenim površinam, je bilo potrebno prilagoditi prostore, kjer smo postavili poligona. Poskuse smo izvajali v notranjih prostorih, kar pomeni, da je bilo veliko homoge-

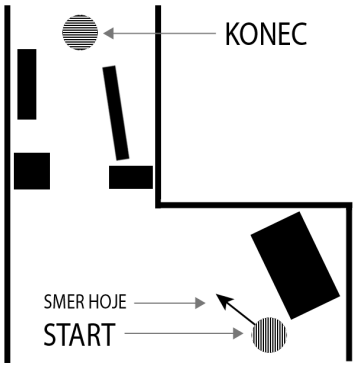


Slika 4.7: Skica poligona 1. Črne regije so ovire in zidovi. Krog z navpičnimi črtami predstavlja začetek poligona, krog z vodoravnimi črtami pa predstavlja konec poligona. Subjekt na začetku poskusa stoji na začetku poligona in je usmerjen proti cilju.

nih površin, kot so na primer stene. Homogene površine so problematične pri delovanju algoritma za izogibanje oviram, saj ta ne upošteva slikovnih elementov, za katere ne pozna globine. Te površine smo popestrili z raznimi predmeti, barvnimi papirji in pohištvom. Tako je algoritem za izogibanje oviram upošteval vse površine v prostoru in posledično je bilo delovanje sistema boljše.

Poligon 1 je bil enostavnejši. Ob začetku poskusa je subjekt stal na začetku ravnega hodnika, njegov cilj pa se je nahajal na drugi strani hodnika. Po hodniku so bile postavljene ovire, ki se v času poskusov niso premikale. Subjekt se je moral izogibati stenam in oviram. Slika 4.7 prikazuje postavitev ovir v poligonu 1.

Da bi bil poligon 2 težji, smo ga zasnovali rahlo drugače. Ob začetku poskusa je subjekt stal sredi prostora pred hodnikom. Da je subjekt sploh prišel do hodnika, je moral na prehodu med prostoroma zamenjati smer. Cilj pa se je, prav tako kot pri prvem poligonu, nahajal na koncu hodnika. Subjekt se je med gibanjem izogibal sten in nepremikajočih ovir, postavljenih po prostoru in hodniku. Skica 4.8 prikazuje postavitev ovir v poligonu 2.



Slika 4.8: Skica poligona 2. Črne regije so ovire in zidovi. Krog z navpičnimi črtami predstavlja začetek poligona, krog z vodoravnimi črtami pa predstavlja konec poligona. Subjekt na začetku poskusa stoji na začetku poligona in je usmerjen proti cilju.

4.4.2 Izvedba testov

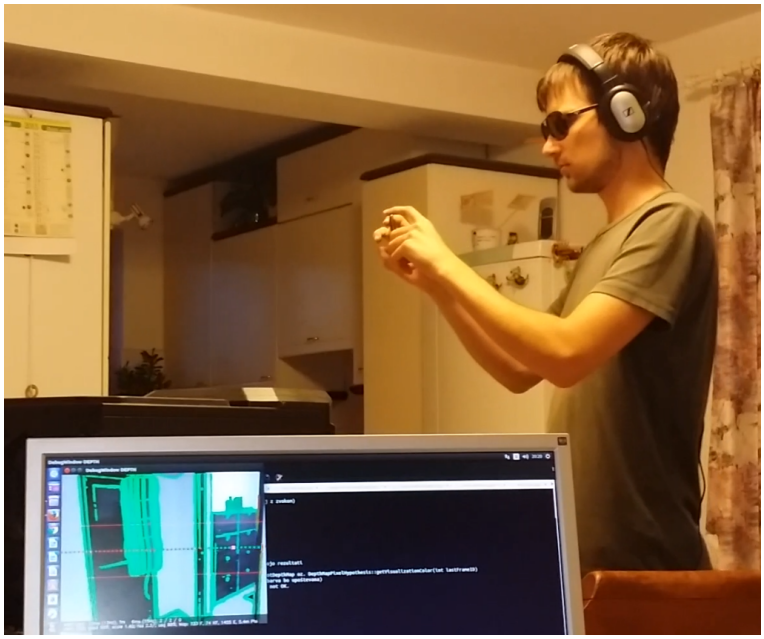
V poskusih smo preverjali, kolikokrat se bo subjekt dotaknil stene ali katere od ovir na poti skozi poligon. Cilj subjekta je bil dosega cilja brez zadevanja v ovire.

Da bi testirali uspešnost subjektov brez pomoči sistema, smo pri poskusih sledili naslednjim korakom:

1. Subjektu se je pred začetkom poskusa zavezalo oči, da si ta ni mogel neposredno pomagati z vizualno informacijo.

2. Subjekt se je nekajkrat zavrtel na mestu, da ni imel svežih podatkov o svoji lokaciji in usmerjenosti. Ko se je ta nehal vrteti, se je lahko orientiral le na podlagi zvoka, ki je izviral iz sosednjega prostora - tako smo zadostili predpostavko algoritma za izogibanje oviram, da mora biti subjekt približno usmerjen, če želi doseči svoj cilj (več o tem v Poglavju 3).

3. Samo s pomočjo taktilne informacije se je subjekt podal proti cilju.



Slika 4.9: Subjekt izvaja poskus na poligonu 2. Pred seboj drži mobilni telefon s kamero, katere nadgrajena slika je vidna na spodnjem levem delu slike. Iz vizualizacije je razvidno, da sistem usmerja subjekt proti sredini prehoda na hodnik. Smer mu sporoča prek stereo slušalk.

4. Ko je subjekt prispel do cilja, smo ga opozorili, da je bil poskus zaključen.

Zgoraj opisan eksperiment smo ponovili še z uporabo našega sistema. Koraki izvajanja poskusa so se predvsem razlikovali v začetni fazi. Po koraku (2) smo v roke subjekta dali mobilni telefon, ki je služil kot kamera za ostali del sistema. Na glavo smo mu nadeli slušalke, prek katerih je slišal zvok iz smeri, ki jo je sistem izbral. Tokrat je imel subjekt med poskusom na voljo tudi informacije od sistema (Slika 4.9).

4.4.3 Rezultati

Na poligonu 1 so se subjekti ob pomoči sistema v povprečju zaleteli 0.28 krat na poskus s standardnim odklonom 0.46. Brez sistema pa so se v povprečju zaleteli 1.44 krat na poskus s standardnim odklonom 0.80.

Da bi potrdili hipotezo, da se človek ob uporabi sistema bolj poredko zaleti v ovire kot če ga ne uporablja, smo izvedli Mann-Whitneyev U-test [23] in T-test [14] na številu trkov. Oba testa zavrneta ničelno hipotezo, če se izkaže, da je statistična pomembnost manjša od 5%. Mann-Whitneyev U-test ugotavlja, katera od naslednjih dveh hipotez drži. Ničelna hipoteza pravi, da ima par vhodnih vektorjev enako mediano, alternativna hipoteza pa, da par vhodnih podatkov nima enakih median. Test predpostavlja, da sta vhodna vektorja medsebojno neodvisna.

V primeru poligona 1 je Mann-Whitneyev U-test ovrgel ničelno hipotezo s statistično pomembnostjo $P = 4.0261 \cdot 10^{-8}$. Torej velja alternativna hipoteza. To pomeni, da je statistično pomembna razlika med rezultati testov.

Podobne rezultate daje tudi T-test. Ta primerja naslednji hipotezi. Ničelna hipoteza pravi, da je razlika vhodnih vektorjev porazdeljena z normalno distribucijo in ima povprečno vrednost enako nič. Alternativna hipoteza pa pravi, da povprečje razlike vektorjev ni enako nič.

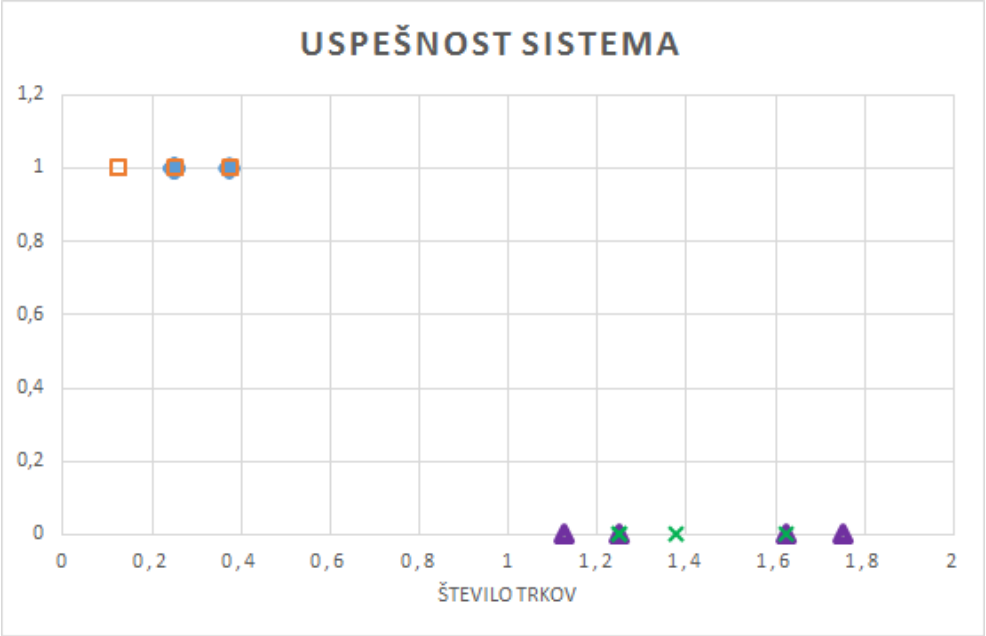
Za poskuse poligona 1 je T-test ovrgel ničelno hipotezo s statistično pomembnostjo $P = 1.1048 \cdot 10^{-7}$. Tudi v tem primeru kaže, da se rezultati statistično pomembno razlikujejo.

Kot kaže, uporaba sistema zelo pripomore k uspešnemu izogibanju oviram. To potrjujejo tudi rezultati testov na poligonu 2, kjer so se subjekti ob pomoči sistema v povprečju zaleteli 0.22 krat na poskus s standardnim odklonom 0.42. Brez sistema pa so se v povprečju zaleteli 1.38 krat na poskus s standardnim odklonom 0.87.

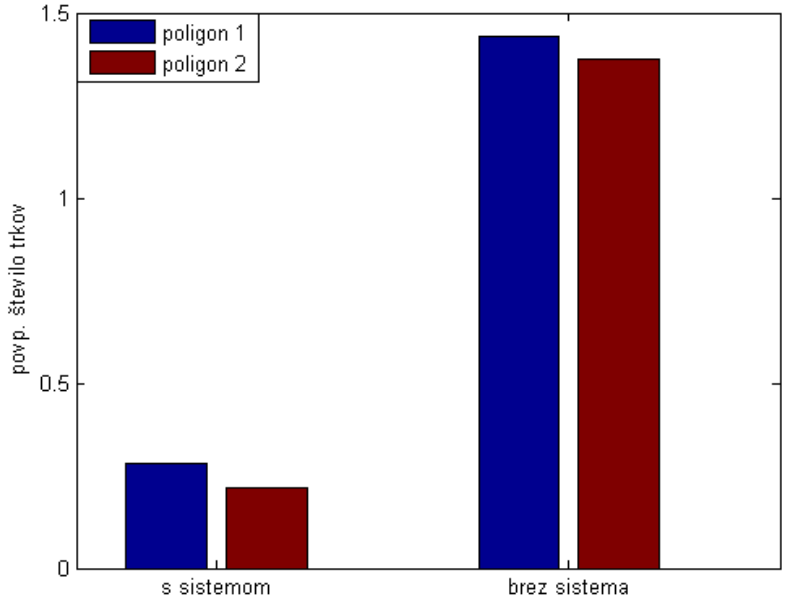
Mann-Whitneyev U-test je pri poligonu 2 prav tako kot pri poligonu 1 ovrgel ničelno hipotezo, tokrat s statistično pomembnostjo $P = 3.5530 \cdot 10^{-8}$.

Test 1s	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mathbf{s_3}$	$\mathbf{s_4}$	**Test 2s**	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mathbf{s_3}$	$\mathbf{s_4}$
#1	0	1	0	1	#1	0	1	0	0
#2	0	0	0	0	#2	0	0	0	0
#3	1	0	1	0	#3	0	1	0	1
#4	0	0	0	0	#4	1	0	1	0
#5	0	1	1	0	#5	0	0	0	0
#6	0	0	0	1	#6	1	0	0	0
#7	1	0	1	0	#7	1	0	0	0
#8	0	0	0	0	#8	0	0	0	0
μ	0,250	0,250	0,375	0,250	μ	0,375	0,250	0,125	0,125
σ	0,463	0,463	0,518	0,463	σ	0,518	0,463	0,354	0,354
Test 1b	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mathbf{s_3}$	$\mathbf{s_4}$	**Test 2b**	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mathbf{s_3}$	$\mathbf{s_4}$
#1	1	2	1	1	#1	1	1	2	2
#2	1	1	1	2	#2	2	1	1	2
#3	0	1	2	2	#3	1	1	2	1
#4	2	2	1	1	#4	0	1	0	1
#5	3	2	2	3	#5	1	2	2	1
#6	1	1	3	2	#6	1	1	2	1
#7	0	1	1	2	#7	2	0	1	2
#8	1	0	2	1	#8	2	4	0	3
μ	1,125	1,250	1,625	1,750	μ	1,250	1,375	1,250	1,625
σ	0,991	0,707	0,744	0,707	σ	0,707	1,188	0,886	0,744

Tabela 4.2: Na tabeli so naštete meritve posameznih testov. Tabela je sestavljena iz štirih delov. Vsak del prikazuje rezultate ustreznega testa. **Test 1s** je test uspešnosti z uporabo sistema na poligonu 1, **Test 1b** je test uspešnosti brez uporabe sistema na poligonu 1, **Test 2s** in **Test 2b** pa ustrezata različicam ustreznih testov na poligonu 2. V stolpcih je predstavljeno število trkov posameznega subjekta s_i za 8 poskusov. Za vsak test smo vsakemu subjektu s_i izračunali njegovo povprečno število trkov μ in standardni odklon σ.



Slika 4.10: Na grafu je prikazana uspešnost subjektov na posameznih testih. Vsakemu subjektu smo za posamezen test izračunali povprečno vrednost μ števila trkov pri izvajanju testa. Njegovo uspešnost smo vizualizirali tako, da smo na grafu z osjo x predstavili število trkov subjekta v posameznem testu, vrednost y pa smo postavili na 1, če je bil pri testu uporabljen sistem za izogibanje oviram, in 0, če sistem ni bil uporabljen. V zgornjem levem kotu so zbrane povprečne vrednosti trkov μ testov, pri katerih so subjekti uporabljali sistem. V spodnjem desnem predelu grafa pa so povprečne vrednosti trkov μ testov, kjer se sistema ni uporabljalo. Vsak test ima svojo oznako. Ker se določene vrednosti podvajajo, je nekaterih oznak manj kot drugih.



Slika 4.11: Povprečno število dotikov pri poskusih z in brez uporabe sistema na prvem in drugem poligonu.

T-test je tudi ovrgel ničelno hipotezo s statistično pomembnostjo $P = 6.4696 \cdot 10^{-7}$.

Kot kaže, so rezultati subjektov, ki so uporabljali sistem, bistveno boljši od rezultatov subjektov, ki sistema niso uporabljali. Rezultati testiranja dobro nakazujejo prednost uporabe sistema. V Tabeli 4.2 so našteta števila trkov vseh subjektov za vsak poskus izveden v vseh štirih testih. Posameznemu subjektu smo pri vsakem testu izračunali povprečno število trkov μ in standardni odklon σ. Povprečne vrednosti μ_i so na Sliki 4.10 vizualizirane kot točke v 2D prostoru, kjer lahko vidimo, da uporaba našega sistema neposredno vpliva na uspeh subjektov. Na Sliki 4.11 je razviden uspeh subjektov na posameznih testih. Subjekti so se ob uporabi sistema zaletavali približno šest-krat redkeje.

4.4.4 Diskusija

Zavedamo se, da na rezultate testov vplivajo različni faktorji, katere težko stabiliziramo. Ljudje si pomagamo na različne načine pri reševanju problemov, tudi če se tega ne zavedamo. Kot primer tega je bil poskus, ko so subjekti brez pomoči sistema poskusili le zamižati in hoditi od začetka do konca poligona. Na ta način so dosegali boljše rezultate, kot če so se pred začetkom poskusa nekajkrat zavrteli in s tem izgubili sveže podatke o svoji lokaciji in usmerjenosti. Zato so se subjekti pri izvajanju poskusov vedno zavrteli pred začetkom.

Med testiranjem smo ugotovili, da so se subjekti sproti privajali na sistem in se učili, kako ga je potrebno uporabljati za dosego boljših rezultatov. Eden od subjektov je po testiranju večkrat prehodil prvi poligon. V njegovih zadnjih 15 poskusih je dosegel povprečje 0,13 trkov na poskus. Ta ugotovitev kaže na velik potencial sistema, ki bi ga bilo mogoče doseči z izpopolnjevanjem in nadgradnjami. Možnim nadgradnjam je namenjeno Poglavje 5.1.

Poglavje 5

Sklep

V okviru diplomske naloge smo razvili sistem, ki lahko pomaga slepim ljudem pri navigaciji. Pregledali smo podporne sisteme, principe in algoritme računalniškega vida, ki jih uporabljamo pri našem pristopu. Predstavili smo delovanje posameznih komponent našega sistema. Opisali smo način vzpostavitve IP kamere in prenos slik prek ROS-ovega [28] sistema sporočanja v metodo LSD-SLAM [6], od katere smo razložili postopek računanja globinskih slik. Obdelali smo algoritem za izogibanje oviram, ki na podlagi globinskih slik izračuna najbolj obetavno pot za uporabnika. Razloženo je bilo, zakaj izbrana pot vodi mimo zaznanih ovir. Opisali smo, kako uporabnika sistem o izbrani poti obvesti s stereofonskim zvokom, katerega uporabnik lokalizira in mu sledi.

Izvedli smo več analiz sistema. V prvi smo analizirali natančnost sistema. Ugotovili smo, da natančnost meritve razdalij z oddaljenostjo pada. Standardni odklon meritev pri oddaljenosti 10 m znaša 0.626, pri oddaljenosti 0.25 m pa 0.005. V drugi alanizi smo merili uspešnost sistema na poligonu. Rezultati kažejo, da je uporabnik s sistemom približno šest-krat uspešnejši pri izogibanju oviram kot brez sistema. Izvedli smo tudi statistična testa Mann-Whitneyev U-test in T-test. Oba vidno nakazujeta, da obstaja velika statistična razlika med meritvami poskusov s sistemom in brez njega.

Kljub dobrim rezultatom pa je potrebno omeniti, da je sistem le prototip, ki predstavlja konceptualno rešitev. V trenutnem stanju ni primeren za praktično rabo v nestrukturiranih realnih okoljih. Uporabnik bi izgubil več časa za kalibracijo sistema in vzdrževanje delujočega stanja sistema, kot bi ga porabil za dosego cilja brez sistema. Da bi sistem postal primeren za vsakdanjo uporabo, bi ga bilo potrebno nadgraditi.

5.1 Možne nadgradnje

Da bi bil naš sistem uporaben v vseh situacijah, bi lahko marsikaj izboljšali. Metoda LSD-SLAM [6] ob uporabi kamere z večjim zornim kotom bi zagotovo delovala bolje, saj bi imela na razpolago podatke o širši okolici, kar bi ji olajšalo sledenje premikom kamere. Primerna kamera bi imela sposobnost istočasnega osveževanja vseh slikovnih točk (angl. global shutter), širokokotni objektiv in visoko frekvenco osveževanja slike.

Poleg tega bi fuzija s senzorji, kot so žiroskopi in pospeškomeri, izboljšala uporabnost sistema predvsem zaradi natančnejšega sledenja usmerjenosti uporabnika, ki poskuša ugotoviti od kod prihaja generiran zvok.

Celoten sistem bi bilo mogoče prenesti na mobilni telefon, saj je LSD-SLAM dovolj učinkovit, da bi se lahko izvajal v realnem času na šibkejšem procesorju mobilnega telefona.

Naš sistem v vsakem ciklu delovanja ponovno izračuna, kam postaviti zvok za usmerjanje uporabnika. Pri tem je vedno prisoten šum, ki občasno povzroča premik izvora zvoka in posledično zmede uporabnika. Če bi pri tem uporabili globalne koordinate sveta, bi lahko zvok fiksirali na določeno točko. To bi uporabnika manj begalo, saj bi bil izvor zvoka stabilnejši in bi ga bilo lažje lokalizirati.

Ena bolj entuziastičnih idej je bila, da bi sistem nadgradili z umetno inteligenco, ki bi se lahko učila poti in vodila človeka proti njegovemu cilju, pri čemer bi upoštevala vse potencialne ovire, kot so stopnice, drogovi javne razsvetljave oz. karkoli, kar predstavlja človeku nevarnost trka.

Literatura

[1] Mounir Bousbia-Salah, Abdelghani Redjati, Mohamed Fezari, and Maamar Bettayeb. An ultrasonic navigation system for blind people. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, pages 1003–1006. IEEE, 2007.

[2] The Audiology Awareness Campaign. How We Hear: External, Middle & Inner Ear. `http://www.audiologyawareness.com/hearinfo_howhear.asp`, 2010. [Online; accessed 03-September-2015].

[3] WHO Media centre. Visual impairment and blindness. `http://www.who.int/mediacentre/factsheets/fs282/en/`, 2014. [Online; accessed 30-August-2015].

[4] Sakmongkon Chumkamon, Peranitti Tuvaphanthaphiphat, and Phongsak Keeratiwintakorn. A blind navigation system using rfid for indoor environments. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, volume 2, pages 765–768. IEEE, 2008.

[5] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[6] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[7] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1449–1456. IEEE, 2013.

[8] Kevin R Fall and W Richard Stevens. *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley, 2011.

[9] José Faria, Sérgio Lopes, Hugo Fernandes, Paulo Martins, and João Barroso. Electronic white cane for blind people navigation assistance. In *World Automation Congress (WAC), 2010*, pages 1–7. IEEE, 2010.

[10] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[11] Laurent Gomila. Simple and Fast Multimedia Library. `http://www.sfml-dev.org/`, 2007. [Online; accessed 30-August-2015].

[12] Google. Google Self-Driving Car Project. `http://www.google.com/selfdrivingcar/`, 2015. [Online; accessed 2-September-2015].

[13] Google. Google Self-Driving Car Project. `http://mars.nasa.gov/msl/`, 2015. [Online; accessed 2-September-2015].

[14] Winston Haynes. Student's t-test. In *Encyclopedia of Systems Biology*, pages 2023–2025. Springer, 2013.

[15] Patrick Mihelich James Bowman. camera calibration. `http://wiki.ros.org/camera_calibration`, 2014. [Online; accessed 30-August-2015].

[16] Esteban Bayro Kaiser and Michael Lawo. Wearable navigation system for the visually impaired and blind people. In *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on*, pages 230–233. IEEE, 2012.

[17] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.

[18] Pavel Khlebovich. IP Webcam. `https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en`, 2010. [Online; accessed 30-August-2015].

[19] Matej Kristan. *Machine perception Camera geometry.* prosojnice iz predavanj, 2015.

[20] Tony Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7(5):10491, 2012.

[21] Prof. Dr. Logothesis. Visual Perception - Psychophysics, Physiology and fMRI Studies. `http://www.kyb.tuebingen.mpg.de/research/dep/lo/visual-perception.html`, 2015. [Online; accessed 30-August-2015].

[22] Diego López-de Ipiña, Tania Lorido, and Unai López. Indoor navigation and product recognition for blind people assisted shopping. In *Ambient Assisted Living*, pages 33–40. Springer, 2011.

[23] Patrick E McKnight and Julius Najab. Mann-whitney u test. *Corsini Encyclopedia of Psychology*, 2010.

[24] Microsoft. Microsoft Independence Day. `http://news.microsoft.com/stories/independence-day/`, 2014. [Online; accessed 30-August-2015].

[25] Paul Newman and Kin Ho. Slam-loop closing with visually salient features. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 635–642. IEEE, 2005.

[26] NFB. Statistical Facts about Blindness in the United States. `https://nfb.org/blindness-statistics`, 2014. [Online; accessed 30-August-2015].

[27] Parrot. AR.Drone 2.0. `http://ardrone2.parrot.com/`, 2015. [Online; accessed 2-September-2015].

[28] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

[29] Lisa Ran, Sumi Helal, and Steve Moore. Drishti: an integrated indoor/outdoor blind navigation system and service. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 23–30. IEEE, 2004.

[30] Thomas Schops, Jakob Engel, and Daniel Cremers. Semi-dense visual odometry for ar on a smartphone. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 145–150. IEEE, 2014.

[31] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. In *Robotics: Science and Systems*, volume 2, page 5, 2010.

[32] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[33] ustwo and RLSB. Wayfindr, an aid to navigation on the London transport network for vision-impaired people. `https://ustwo.com/blog/validating-wayfindr/`, 2014. [Online; accessed 30-August-2015].

[34] Eric W Weisstein. *Fast fourier transform*. Wolfram Research, Inc., 2015.

[35] Zhan Huan Zhou. Sound localization and virtual auditory space. *Project report of University of Toronto*, 19, 2002.

[36] Michael Zöllner, Stephan Huber, Hans-Christian Jetter, and Harald Reiterer. *NAVI–A proof-of-concept of a mobile navigational aid for visually impaired based on the microsoft kinect*. Springer, 2011.