

Določanje ravnine za zaznavanje ovir in proste poti avtonomnega mobilnega robota

Anamarija Hauptman, Maruša Petrovčič, Žiga Špiclin, Žiga Bizjak

Fakulteta za elektrotehniko, Univerza v Ljubljani, Tržaška c. 25, 1000 Ljubljana
ziga.bizjak@fe.uni-lj.si

Plane Detection for Obstacle Recognition and Free Path Determination of an Autonomous Mobile Robot

This study presents a method for free path detection and obstacle avoidance in autonomous mobile robots using planar depth imaging with an Intel RealSense Depth Camera. Our approach bypasses the complexities of deep learning, integrating depth and planar data to dynamically distinguish free paths from obstacles. Experimental results demonstrate robust obstacle detection accuracy and reduced computational requirements, making it a robust and energy-efficient solution for industrial AMR applications. This method can be utilised for obstacle detection in AMR navigation in uncontrolled environments, contributing to enhanced productivity and safety in industrial settings.

keywords: AMR, point cloud, plane segmentation, object detection

1 Uvod

Dandanes se industrija vse bolj usmerja v avtomatizacijo in robotizacijo proizvodnih procesov. Na ta način poskušajo podjetja izboljšati učinkovitost delovanja, pospešiti proizvodni proces in zmanjšati zastoje ter tako izboljšati produktivnost, natančnost in varnost. Za izvajanje nalog, kot so prevoz materialov, upravljanje zalog in operacije na proizvodnih linijah, se pogosto uporabljajo avtonomni mobilni roboti (angl. *autonomous mobile robots*, AMR).

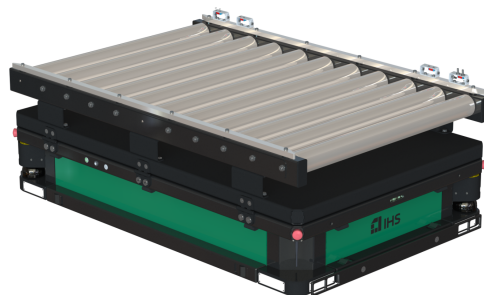
Za razliko od t.i. avtomatsko vodenih vozil (angl. *automated guided vehicles*, AGV), ki se po proizvodnih prostorih gibljejo s sledenjem črtam ali žičnim povezavam, avtonomni mobilni roboti omogočajo navigacijo v nekontroliranih okoljih. AMR-ji uporabljajo različne senzorje in algoritme umetne inteligence za načrtovanje poti. Kamere in senzorji omogočajo zaznavanje nepričakovanih objektov na robotovi poti, kot so na primer škatle ali ljudje. Ob zaznavi ovire se lahko AMR trku izogne (angl. *collision avoidance*) na več načinov. Najpogostejši so ustavitev, upočasnitev ali sprememba poti, tako da vozilo objekt obide in nadaljuje z izvajanjem naloge.

Za zaznavo objektov se v zadnjem času pogosto predlagajo metode globokega učenja [1], vendar, kljub njihovim prednostim, te metode prinašajo tudi omejitve. Metode globokega učenja so običajno prilagojene specifičnim

okoljem (na primer skladiščem), zahtevajo visoko računsko zahtevnost in s tem povečajo porabo energije, kar je ključni faktor pri avtonomnih vozičkih. Zato v tem delu predlagamo klasično metodo, ki temelji na zaznavi prostega voznega polja s pomočjo ravnine.

2 Metodologija

Detekcijo proste poti in zaznavo ovir smo izvajali za potrebe avtonomnega mobilnega robota podjetja IHS, imenovanega SmArtGV 1500, ki se uporablja za transport skladiščnih palet. Primer vozička je prikazan na sliki 1.

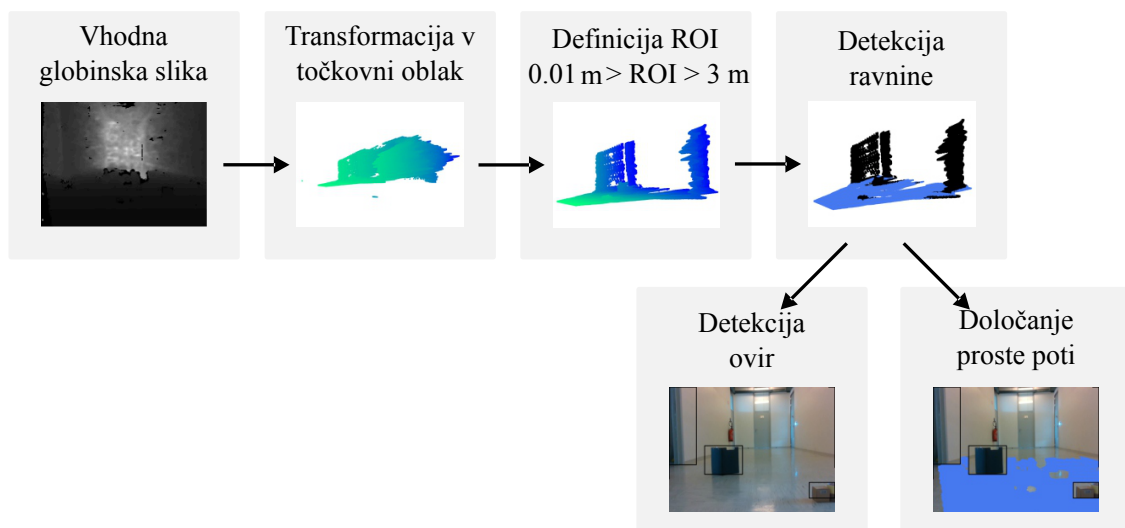


Slika 1: Prikaz avtonomnega mobilnega robota podjetja IHS, imenovanega SmArtGV.

2.1 Uporabljena strojna oprema

Robot je opremljen z dvema 2D laserskima skenerjema, ki sta nameščena na dveh diagonalnih vogalih mobilnega robota na višini 15 cm. Na ta način lahko spremljamo celotno okolico robota (360°) v ravnini namestitve skenerjev. Le-ta delujeta na principu merjenja časa preleta (angl. *time-of-flight*) laserskega žarka od vira do ovire in nazaj v detektor. Na ta način lahko določimo razdaljo med skenerjem ter posamezno točko v 2D ravnini. AMR se giblje samo v ravnini, polno naložen pa lahko doseže tudi višino dveh metrov. Pomanjkljivost obstoječe konfiguracije mobilnega robota je, da s parom laserskih senzorjev lahko zaznamo samo predmete na višini 15 cm od tal, ničesar pa ne vemo o ovirah, nižjih od 15 cm, ter o morebitnih ovirah v zraku.

Za rešitev problema nezadostnega zaznavanja okolice je bila na AMR na sprednji strani v smeri vožnje, približno 36 cm od tal, nameščena kamera Intel RealSense Depth Camera D435i. Kamera omogoča zajemanje slik



Slika 2: Prikaz predlaganega postopka za detekcijo ovir in določanje proste poti.

RGB in zajemanje globinskih slik (angl. *depth images*). Podatki se iz skenerjev in kamere sproti pošiljajo v osrednji računalnik, kjer poteka njihova obdelava.

2.2 Določanje proste poti

Čeprav kamera Intel RealSense D435i omogoča snemanje tako RGB kot globinskih slik, smo v naši metodi uporabili le globinske slike. RGB slike iz kamere smo uporabili zgolj za vizualni prikaz v članku. Shema postopka določanja proste poti in iskanja ovir je prikazana na sliki 2.

2.2.1 Pretvorba globinske slike v točkovni oblak

Na začetku postopka preberemo globinsko sliko, v kateri je v vsakem slikovnem elementu (angl. *pixel*) zapisan 16-bitni podatek o oddaljenosti te točke od kamere. Nato s pomočjo enačb 1 in 2 za vsak slikovni element (u, v) določimo vrednosti koordinat x in y v 3D prostoru.

$$x = (u - x_{oc}) \cdot \frac{z}{x_{fl}} \quad (1)$$

$$y = (v - y_{oc}) \cdot \frac{z}{y_{fl}} \quad (2)$$

V enačbah 1 in 2 spremenljivka z predstavlja vrednost tretje koordinate v 3D prostoru, ki jo dobimo tako, da preberemo vrednost slikovnega elementa (u, v) na globinski sliki. x_{oc} in y_{oc} predstavljata koordinati optičnega centra kamere, x_{fl} in y_{fl} pa sta horizontalna in vertikalna goriščna razdalja leče kamere.

Rezultat, ki ga vrne naša funkcija za pretvorbo iz globinske slike v oblak točk, je seznam točk v 3D prostoru, ki jih nato zapišemo v obliki točkovnega oblaka (angl. *PointCloud* [2]).

2.2.2 Definicija ROI

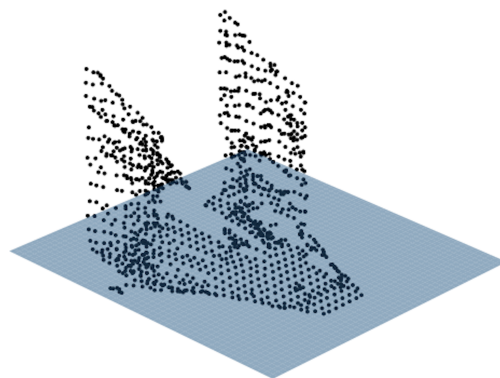
Bolj oddaljene strukture, zaznane z globinsko kamero, so zelo podvržene šumu. Zato smo iz oblaka točk odstranili točke, ki so od kamere oddaljene več kot 3000 mm. Prav tako smo odstranili točke, ki so bile pretvorjene v točkovni oblak in določene na koordinato $[0, 0, 0]$, saj so le-te prispevale k napačni določitvi ravnine.

2.2.3 Izračun ravnine

Tako dobljeni oblak točk nato uporabimo kot vhod v Open-3D funkcijo *segment_plane*. Le-ta v oblaku poišče optimalno ravnino in vrne parametre a , b , c in d za njen funkcijski predpis (enačba 3).

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \quad (3)$$

Funkcija *segment_plane* temelji na metodi RANSAC (angl. *RANdom SAMple CONsensus*) [3]. Poleg vhodnega oblaka točk lahko nastavimo tudi vrednosti parametrov: največjo razdaljo med točko in ravnino, da jo še štejemo kot del ravnine (*Distance_threshold*), število naključno vzorčenih točk za oceno ravnine (*ransac_n*) in število vzorčenj ter preverjanj ravnine (*num_iterations*).



Slika 3: Definirana ravnina (modro področje) glede na vhodni točkovni oblak (črne točke). Pokončna oblaka točk predstavljata levo in desno steno hodnika.

Funkcija najprej naključno izbere *ransac_n* točk iz oblaka točk. Na podlagi teh točk nato določi ravnino ter za ostale točke, glede na mejno vrednost *distance_threshold*, preveri, ali pripadajo izbrani ravnini.

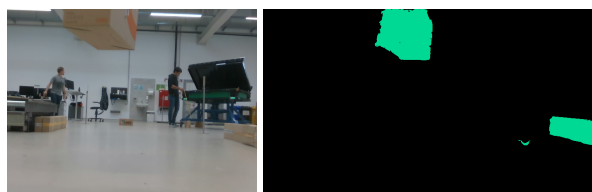
Uporabljena funkcija poleg vrednosti parametrov za funkcijski zapis ravnine vrne še vrednosti indeksov točk v oblaku, ki pripadajo ravnini. Na podlagi indeksov smo

ločili točke, ki so del ravnine, od točk, ki ravnini ne pripadajo. Tako smo tudi določili, ali obstaja prosta pot za premik vozička.

Med uporabo vozička v skladišču lahko predvidevamo, da bo ravnina tal ostala nespremenjena. Zato lahko enačbo ravnine posodabljam v daljših časovnih intervalih, na primer vsako sekundo, kar zmanjšuje računsko zahtevnost in porabo energije.

2.3 Detekcija ovir

Z določeno ravnino smo določili varno pot premika, vendar posameznih ovir še nismo detektirali. V tem delu smo detekcijo ovir izvedli na 2D slikah, ki so bile ustvarjene iz točkovnega oblaka. Informacijo o točkah, ki ne pripadajo ravnini, smo pridobili iz podpoglavja 2.2.3 in na podlagi te informacije pripravili masko za globinsko sliko, s katero smo označili vsa ta področja. Primer povratne preslikave na globinsko sliko je prikazan na sliki 4.

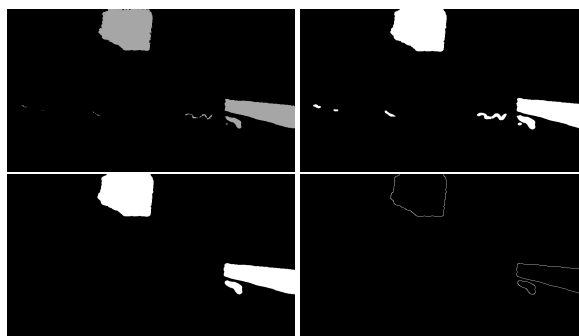


Slika 4: Originalna RGB slika (levo) in preslikana informacija o področjih, ki ne pripadajo ravnini, na izvorno globinsko sliko (desno). Zeleno področje na vrhu slike predstavlja visečo škatlo v skladišču, medtem ko zeleno področje na desni predstavlja manjše kartonaste škatle na tleh.

Sledila je obdelava barvne 2D slike s pomočjo funkcij iz knjižnice OpenCV [4]. Najprej smo sivinsko sliko filtrirali z Eliptičnim jedrom (funkcija *filter2D - morphologyEx*, velikosti 7×7). Zamegljeno sliko smo morfološko odprli (funkcija *morphologyEx*, kjer smo velikost jedra nastavili na 13×13), s čimer smo se znebili šuma, ki se je pojavil že pri generiranju oblaka točk na podlagi globinske slike. Na morfološko odprti sliki smo na podlagi Cannyjevega detektorja robov [5, 6] (funkcija *Canny*) poiskali robove na sliki. Ti robovi so hkrati že predstavljali tudi obrise objektov. Sliko robov smo uporabili kot vhod v funkcijo *findContours*, ki na sliki poišče skupke točk in tako med seboj loči posamezne objekte. Za grafični prikaz rezultatov smo okrog vsakega objekta narisali pravokotnik (funkcija *boundingRect*).

3 Rezultati

Metodologijo smo testirali na štirih 16-bitnih posnetkih, zajetih v skladišču z IHS robotom, in petih posnetkih, opravljenih z isto kamero na hodniku Fakultete za elektrotehniko. Za prikaz delovanja in detekcije ravnine smo naključno izbrali po 10 sekund iz vsakega posnetka. Pri evalvaciji nismo imeli zlatega standarda za natančno lokacijo ravnine, zato smo rezultate določanja ravnine in področja proste poti ocenjevali vizualno. Na RGB slike smo za prikaz dodali detekcijo ovir in pobarvali področje ravnine. Slika 6 prikazuje šest različnih situacij, kjer eval-



Slika 5: Obdelava 2D slike (zgoraj levo: sivinska slika, zgoraj desno: zamegljena slika, spodaj levo: morfološko odprta slika, spodaj desno: robovi na sliki)

viramo detekcijo naključnih ovir na tleh (zgornji dve vrstici slik) ter ljudi in ovire v zraku (spodnji dve sliki).

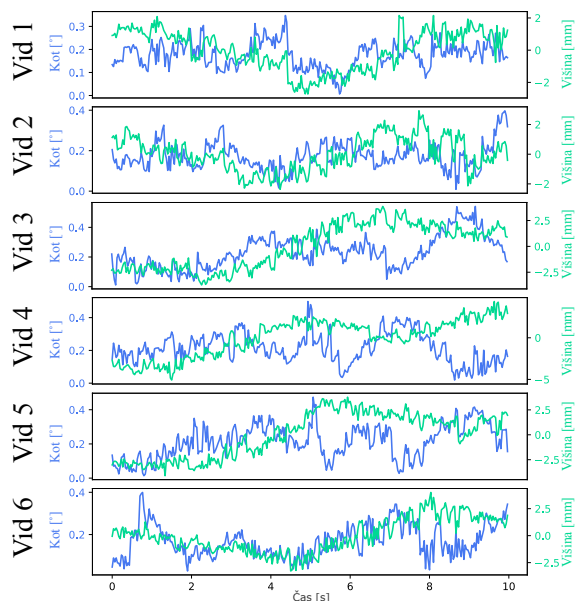
Ker algoritem kot ovire zazna tudi stene, se jim lahko voziček izogne, še preden se jim približa na takšno razdaljo, da bi obstajala nevarnost, da je ravnina stene obsežnejša od ravnine tal, zaradi česar bi lahko algoritem odpovedal.



Slika 6: Grafični prikaz delovanja iskanja proste poti, kjer modro obarvano področje predstavlja prosto pot, omejeno na razdaljo 3 metrov, in detekcije ovir, kjer so vse najdene ovire označene s črnimi kvadrati.

Če so tla ravna in če se kamera ne premika, bi morala ravnina ostati nespremenjena skozi čas, zato smo se odločili analizirati parametre ravnine. Kot zlati standard smo privzeli povprečno normalo v opazovanem obdobju n_0 (parametri a , b , c) ter povprečno višino ravnine z_0 (parameter d). Spremembo parametrov a , b in c smo ocenili z izračunom kota med trenutno normalo n_i in normalo n_0 , kjer i predstavlja trenutno sličico posnetka. Prav tako smo primerjali višino z_i glede na višino z_0 .

Na grafih na sliki 7 prikazujemo spremembe skozi čas za z_i in n_i pri šestih posnetkih. Opazimo, da se kot med



Slika 7: Prikaz sprememb kota med n_i in povprečno normalo n_0 (modra črta) v stopinjah ter prikaz sprememb z_i glede na povprečno višino z_0 (zelena črta) v opazovanem času. Prikazi so za šest krajših odsekov iz različnih posnetkov, in sicer iz skladišča (video 1-3) ter Fakultete za elektrotehniko (video 4-6).

normalama tekom opazovanega obdobja spreminja od 0,0 do 0,5 stopinje glede na povprečni kot. Višina glede na povprečno višino opazovanega področja pa se spreminja v območju ± 5 mm.

4 Diskusija

V zadnjih letih se zaznava ovir v robotskem vidu pogosto izvaja z metodami globokega učenja [7], ki kljub številnim prednostim, kot so natančnost in prilagodljivost, prinašajo tudi slabosti, kot so pristranskost na učno množico, visoka poraba energije, potreba po zmogljivih procesorjih itd. V tem članku predlagamo metodo za določanje proste poti in detekcijo ovir na podlagi določanja ravnine tal. Predlagana metoda je univerzalna in deluje v različnih okoljih, kjer se tla lahko opisujejo z ravnino, ter ne potrebuje zmogljivih grafičnih kartic za delovanje. Razvita je bila za avtonomni mobilni robot SmArtGV, namenjen prevozu palet po skladiščih, kjer predvidevamo ravna tla. Ta robot ima zagotovljene varnostne protokole, skladne z industrijskimi standardi (laser na višini 15 cm, glej poglavje 2.1), vendar ti protokoli ne zaznajo visečih ovir ali ovir nižjih od 15 cm.

Predlagana metoda ima več prednosti pred metodami, ki temeljijo na globokem učenju. Prvič, določanja ravnine ni treba izvajati za vsako sličico v videoposnetku, kar bistveno pohitri detekcijo ovir. Drugič, metoda zahteva le globinsko kamero, zato ni potrebe po RGB slikah, kar zmanjšuje količino obdelanih podatkov. Tretjič, metoda ni omejena na predmete oziroma ovire, zajete v učni množici, in zaznava tudi majhne ovire na tleh sredinske podslike slike 6).

Zasnova metode omogoča številne možnosti za nad-

gradnje. Prva možnost je sledenje oviram in predvidevanje njihove lokacije na podlagi prejšnjih podatkov. Druga možnost nadgradnje je uporaba informacij o prejšnjih parametrih ravnine za določanje nove ravnine, kjer bi lahko uporabili le točke v izbrani širini okoli prejšnje ravnine, s čimer bi hitreje določili novo ravnino.

Predlagana metoda je načeloma neobčutljiva na pozicijo kamere, vendar za optimalno delovanje priporočamo namestitev kamere čim višje nad tlemi. Z višjo kamero se poveča kot med kamero in tlemi, kar zmanjšuje šum točk na tleh, predvsem tistih, ki so bolj oddaljene od kamere.

Naša metoda za določanje ravnine tal in detekcijo ovir predstavlja učinkovit pristop za avtonomne mobilne robote v industrijskih okoljih. Njen uspeh pri detekciji ovir, brez potrebe po obsežnih učnih množicah in zmogljivi strojni opreml, dokazuje njeno praktičnost in prilagodljivost.

Literatura

- [1] A. Balachandran, A. Lal, and P. Sreedharan, "Autonomous navigation of an amr using deep reinforcement learning in a warehouse environment," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, pp. 1–5, IEEE, 2022.
- [2] "Open3d point cloud." Dosegljivo: <https://www.open3d.org/docs/latest/tutorial/Basic/pointcloud.html>. [Dostopano: 26. 5. 2024].
- [3] "Open3d: A modern library for 3d data processing." Dosegljivo: <https://www.open3d.org/docs/release/>. [Dostopano: 26. 5. 2024].
- [4] "Opencv modules." Dosegljivo: <https://docs.opencv.org/4.x/>. [Dostopano: 26. 5. 2024].
- [5] L. Yuan and X. Xu, "Adaptive image edge detection algorithm based on canny operator," in *2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS)*, pp. 28–31, 2015.
- [6] D. Ji, Y. Liu, and C. Wang, "Research on image edge detection based on improved canny operator," in *2022 3rd International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, pp. 229–232, 2022.
- [7] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, 2021.

Zahvala

Avtorji se iskreno zahvaljujemo ekipi razvoja Avtonomnih mobilnih robotov iz podjetja I.H.S. d.o.o. za njihovo sodelovanje in podporo pri izvedbi tega projekta. Njihova strokovna pomoč in vpogledi so bili ključnega pomena pri razumevanju in opisu problematike, ki smo jo obravnavali. Podjetje je poskrbelo za podporne materiale, na katerih smo izvajali vsebine robotskega vida, kar je bistveno pripomoglo k verodostojnosti naših rezultatov. Cenimo njihovo prizadevnost in strokovnost, ki sta pomembno prispevala k uspehu tega projekta. (<https://www.ihs.systems/product.category/amr/>)