

Navigacija robota v zemljevidu kvadratnih celic s pomočjo bilinearne interpolacije diskretnega potencialnega polja

Andrej Zdešar¹, Matevž Bošnjak¹, Rok Vrabič², Gašper Škulj², Gregor Klančar¹

¹Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija

²Univerza v Ljubljani, Fakulteta za strojništvo, Aškerčeva cesta 6, 1000 Ljubljana, Slovenija

E-pošta: andrej.zdesar@fe.uni-lj.si

Robot navigation based on a grid map using bilinear interpolation of a discrete potential field

In this paper a smooth navigation function combining the Dijkstra-based discrete static potential field with bilinear interpolation is proposed. We present the necessary modifications of the bilinear interpolation method to make it applicable to the path-planning application. The proposed navigation strategy solves the problem of the local minima and generates smooth paths with moderate computational complexity. Results from several initial robot configurations are presented to illustrate the advantages of the developed navigation method.

1 Uvod

Glavni cilj planiranja poti je določiti izvedljive in varne poti, ki se izognejo oviram in omogočajo vožnjo robota od začetne do ciljne konfiguracije [1]. To delo združuje iskanje poti v grafu, ki je predstavljen z mrežo celic, in nadgradnjo umetnih potencialnih polj [2] za določitev gladkih poti v prostem prostoru. Predlagani pristop uvaja interpolacijo, ki določi gladke poti tudi pri grobi ločljivosti diskretnega zemljevida.

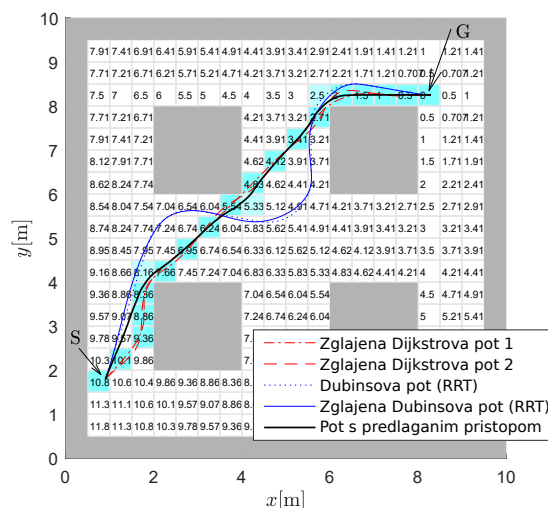
Ravninski zemljevid okolja najpogosteje zapišemo z razcepom prostora na mrežo kvadratnih celic, kjer je prostorska kompleksnost odvisna predvsem od števila celic. Celica mora biti dovolj majhna, da lahko zadovoljivo opišemo pomembne detajle okolja. Za iskanje možnih poti med začetno in ciljno lokacijo se pogosto uporabljata algoritma Dijkstra [3] in A* [4] ter njune številne nadgradnje, kot je D*Lite [5] in podobno. Koncept umetnih ali navidezni potencialnih polj za načrtovanje poti do zelenega cilja je bil prvič predlagan v [2]. Priljubljenost pristopa je posledica njegove intuitivne formulacije, njegova glavna pomanjkljivost pa je prisotnost lokalnih minimumov, za premagovanje katerih je bilo predlaganih več rešitev.

V predlaganem pristopu zemljevid okolja predstavimo z diskretno mrežo (s prostimi in zasedenimi celicami), ki jo uporabimo za določitev diskretnega potencialnega polja. Z metodo bilinearne interpolacije določimo zvezni gradient potencialnega polja, ki nam služi za iskanje optimalne poti s potovanjem v smeri padajočega gradienta. Čeprav je bilinearne interpolacije znana metoda na področju digitalne obdelave slik, je ni moč direktno uporabiti za interpolacijo potencialnega polja in njegovega

gradienta pri načrtovanju poti. V članku predstavljamo izboljšave, ki omogočajo obravnavo zasedenih celic (brez definiranih vrednosti potencialnega polja) in interpolacijo na mejah okolja ter zagotavljajo monotono padanje gradienta potencialnega polja. Tako se pri iskanju poti ne pojavljajo lokalni minimumi, ki so pogosti pri splošnih metodah, ki temeljijo na potencialnem polju. Pristop je računsko učinkovit, saj temelji na zemljevidu z diskretnim potencialnim poljem, ki ga je za statično okolje in znane cilje mogoče izračunati vnaprej ter ga je mogoče znova uporabiti, če se bodisi cilj bodisi start ne spremeni. Rezultat načrtovanja je gladka zvezna pot v prostem prostoru, ki je blizu optimalne, čeprav je lahko diskretna predstavitev okolja precej groba.

2 Predstavitev problema

Za določen cilj z Dijkstrovim algoritmom izračunamo ceno do cilja za vsako prosto celico v mreži zemljevida. Na sliki 1 so na primeru okolja s štirimi ovirami za vse celice prikazane cene do cilja G (ciljna celica ima ceno 0, začetna celica S pa 10,8).



Slika 1: Primer zemljevida okolja z označeno lokacijo starta S in cilja G , cenami do cilja v prostih celicah (bele celice so proste, sive so zasedene) in primeri nekaj različnih poti med S in G

Na sliki 1 je optimalna diskretna pot med startom S in ciljem G , ki jo dobimo z Dijkstrovim algoritmom,

označena s svetlo modrimi celicami. Gladko pot lahko dobimo tako, da čez centre celic, ki sestavljajo pot, napnemo zlepek parametričnih krivulj (npr. Bézierjeve krivulje ali klotoida). Na sliki 1 sta prikazani dve gladkih poti (zglajena Dijkstra pot 1 in 2), ki smo ju na ta način dobili s pomočja paketa *Automated Driving Toolbox* v okolju *Matlab*. Na sliki 1 je prikazana tudi (zglajena) Dubinsova pot, ki smo jo dobili z metodo hitro-rastočega naključnega drevesa (RRT*) [6]. Omenjeni pristopi sami po sebi ne zagotavljajo varnega poteka dobljenih gladkih poti mimo ovir z dovolj velikim odklikom in brez trkov. V ta namen je potrebno pri glajenju poti upoštevati tudi omejitve, ki trke preprečijo. Nekatere zglajene poti so lahko zelo oscilatorne (z velikimi zavoji mimo ovir), lahko pa so dobljene poti tudi neizvedljive, če trčijo ob ovire.

Slika 1 prikazuje tudi primer poti dobljen s predlaganim pristopom. To je gladka pot, določena na podlagi interpoliranega gradienta potencialnega polja, ki implicitno upošteva ovire. Predlagani pristop je determinističen in vrača gladke in skoraj optimalne poti, ki zagotavljajo minimalno varno razdaljo do ovir. Pristop je tudi računsko učinkovit, saj daje zadovoljive rezultate, tudi če je velikost celic relativno velika.

3 Interpolacija in glajenje potencialov in gradientov za načrtovanje poti

Diskretizacija okolja je potreben korak za omejitev računske zahtevnosti kot tudi za poenostavitev upoštevanja zasedenosti prostora z ovirami. Na diskretni mreži se vrednost potencialnega polja (cena do cilja) izračuna za vsako celico. Ločljivost določanja gradienta je tako omejena, kar vpliva na gladkost navigacije. Za rešitev tega problema je uvedena izboljšava z uporabo uveljavljenega pristopa ponovnega vzorčenja slike, znanega kot bilinearna interpolacija (BiLI) [7, 8]. Ta deluje na slikovnih elementih, ki so v našem primeru celice.

3.1 Bilinearna interpolacija

Ideja delovanja metode BiLI je ponazorjena s pomočjo slike 2 [8]. Z interpolacijo ustvarimo nove podatkovne točke z uporabo linearnih funkcij, kar je v bistvu ravninska razširitev 1D linearne interpolacije. Interpolirana funkcija na osnovi zlepkov potrebuje štiri parametre, katerih vrednosti je potrebno oceniti. V ta namen uporabimo vrednosti potenciala na ogliščih interpoliranega območja.

Glede na lokacijo točke $[x, y]^T$ v celici M je za interpolacijo izbrano območje štirih sosednjih celic, katerih središča so povezana s črkanim kvadratom na sliki 2. Normirane koordinate so določene s središči teh celic kot:

$$x_n = \frac{x-x_0}{d_c}, \quad y_n = \frac{y-y_0}{d_c},$$

kjer je $[x_0, y_0]$ izhodišče normaliziranih koordinat, ki je v spodnjem levem oglišču črkanega kvadrata, d_c pa je velikost celice. Interpolirani in diskretni potencial v normiranih koordinatah sta izražena kot $P_n(x_n, y_n) = P(x, y)$ in $U_n(x_n, y_n) = U(x, y)$. Potenciali ustreznih štirih sosednjih centrov celic (oglišča črkanega kvadrata na sliki 2)

so

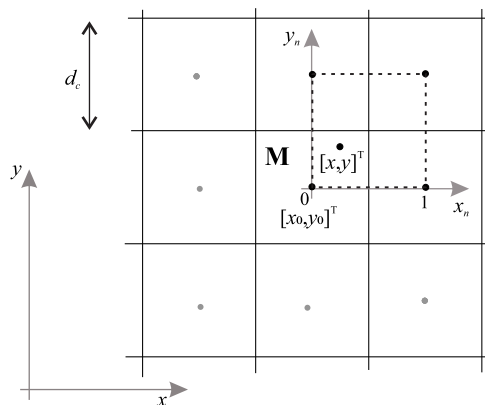
$$p_{cr} = U_n(x_n, y_n) \Big|_{x_n=c, y_n=r}, \quad (1)$$

kjer je $c, r \in \{0, 1\}$ in $U_n(x_n, y_n) = U(x, y)$.

Interpolirani potencial $U_n(x_n, y_n)$ katerekoli normirane lokacije $[x_n, y_n]^T$ znotraj kvadranta celice M , omejene z enotskim kvadratom, je definiran kot [8]:

$$P_n(x_n, y_n) = w_{00}p_{00} + w_{01}p_{01} + w_{10}p_{10} + w_{11}p_{11}, \quad (2)$$

kjer so uteži BiLI določene kot: $w_{00} = (1-x_n)(1-y_n)$, $w_{01} = (1-x_n)y_n$, $w_{10} = x_n(1-y_n)$ in $w_{11} = x_ny_n$.



Slika 2: Interpolirani potencial v dani točki $[x, y]^T$ je definiran z diskretnim potencialom v središčih (črnih pikah) štirih celic, povezanih s črkanim kvadratom

Če sledimo negativnemu gradientu interpoliranega potencialnega polja $P(x, y) = P_n(x_n, y_n)$, lahko dobimo varno pot od kjerkoli v okolju do ciljne lokacije (s potencialom 0). Negativni gradient potenciala $P(x, y)$ v $[x, y]^T$ je mogoče dobiti kot:

$$\mathbf{g}(x, y) = -\nabla P(x, y) = \frac{1}{d_c} \begin{bmatrix} p_{00} - p_{10} - C y_n \\ p_{00} - p_{01} - C x_n \end{bmatrix}, \quad (3)$$

kjer je $C = p_{00} - p_{01} - p_{10} + p_{11}$.

3.2 Prilagoditve bilinearne interpolacije za namen planiranja poti

Pred uporabo interpolacije z enačbo (2) moramo preveriti, če je katera od treh sosednjih celic, vključenih v interpolacijo celice M (glej sliko 2), zasedena. Celica M ni nikoli zasedena, saj interpoliramo potencial v točki $[x, y]^T$ znotraj nje. Za zasedene celice je potencial običajno neskončen ali nedefiniran, saj gibanje čez ovire proti cilju ni možno ali dovoljeno. Potencial $U(x_m, y_m)$ za zasedeno celico s središčem pri (x_m, y_m) lahko določimo iz največjega potenciala nezasedene celice v sosesčini osmih celic:

$$(c, r) = \underset{(i,j)}{\operatorname{argmax}} \{U(x_m + d_c i, y_m + d_c j) \neq \infty\},$$

$$U(x_m, y_m) = U(x_m + d_c c, y_m + d_c r) + d_c \sqrt{c^2 + r^2},$$

kjer $i, j \in \{-1, 0, 1\}$.

Dodatno je potrebno preveriti ali je katera od štirih celic vključenih v interpolacijo (krajše interpolacijska celica,

glej enačbo (2) in sliko 2) zunaj okolja. Preprosta rešitev za to bi lahko bila, da je območje mreže celic vsaj za eno celico večje od območja, ki ga interpoliramo. Splošnejša rešitev uporabljaja za izračun potenciala in gradienta bližnjo lokacijo $[x_t, y_t]^T$, kjer so vse štiri interpolacijske celice znotraj okolja. Za položaj $[x, y]^T$, kjer je ena ali več interpolacijskih celic zunaj okolja, se bližnja lokacija določi s translacijo položaja z meje za $\frac{d_c}{2}$ v smeri x in/ali y proti notranjosti okolja, kot sledi:

$$x_t = \begin{cases} x & ; x_{min} \leq x \leq x_{max} \\ x_{min} + \frac{d_c}{2} & ; x < x_{min} \\ x_{max} - \frac{d_c}{2} & ; x > x_{max} \end{cases},$$

$$y_t = \begin{cases} y & ; y_{min} \leq y \leq y_{max} \\ y_{min} + \frac{d_c}{2} & ; y < y_{min} \\ y_{max} - \frac{d_c}{2} & ; y > y_{max} \end{cases},$$

kjer so meje okolja določene z x_{min} , x_{max} , y_{min} in y_{max} . Za premaknjeno bližnjo lokacijo se interpolirani potencial $P(x_t, y_t)$ izračuna iz (2) in gradient $\mathbf{g}_t(x_t, y_t)$ iz (3). Končno se ustrezen potencial za vsako interpolacijsko celico zunaj okolja (označeno kot P^*) rekonstruira z uporabo Liejevega odvoda (tudi smerni odvod):

$$P^* = P(x_t, y_t) + \mathbf{g}_t(x_t, y_t) \times [x - x_t, y - y_t]^T. \quad (4)$$

Slika 3 prikazuje nastalo potencialno polje, pridobljeno z uporabo pristopa BiLI, ki ustreza okolju na sliki 1. Vidno je, kako se potencialno polje zvezno spušča od začetne do ciljne točke mimo štirih ovir, ki nimajo določenega potenciala. To izhaja iz dejstva, da vrednosti potenciala monotono padajo od katere koli celice v okolju proti cilju. Dodatno je na sliki 4 z modro črto prikazanih nekaj vzorčnih poti, pridobljenih s sledenjem v smeri negativnega gradienta (izračunan iz enačbe (3)) za nekaj različnih začetnih lokacij proti cilju. Opazimo, da so možne nezvezne spremembe smeri gradienta v bližini zasedenih celic. Da bi nezveznosti smeri odpravili in dobili bolj gladke poti (kot je na sliki 4 prikazano z vijolično črto), v poglavju 3.3 dodatno predlagamo interpolacijo gradienta.

3.3 Interpolacija gradienta

Dobljeni potencial na sliki 3 je zvezen, vendar je njegov negativni gradient lahko nezvezen, zlasti v bližini ovir, kot je razvidno s slike 4 na poti negativnega gradienta (modra črta). Negativni gradient označuje zahtevano smer vožnje za doseg cilja na optimalen način. Vsaka nezveznost gradienta je problematična, saj bi se robot s kolesi moral ustaviti in zavrteti na mestu, da bi lahko zanesljivo šel v smeri parajočega gradienta proti cilju. Da bi to izboljšali, predlagamo interpolacijo gradienta, podobno kot je to izvedeno za potencialno polje v poglavju 3.1.

Za izbrano lokacijo $[x, y]^T$ v celici M ocenimo interpolirani potencial $P(x, y)$ in njegov negativni gradient \mathbf{g} z uporabo enačb (2) in (3). Interpolirani potencial se določi iz središčnih potencialov štirih interpolacijskih celic, kot je prikazano na sliki 2. Isti princip interpolacije uporabimo za izračun interpoliranega gradienta $\mathbf{h}(x, y)$, ki

omogoča določitev poti z zveznim potekom, kot je prikazano na sliki 4 z vijolično črto. Gradient za središča štirih interpoliranih celic je mogoče oceniti iz (3), ki za vsako celico upošteva samo levega soseda ($x_n = 0$), da dobimo element x gradienta \mathbf{g} ali samo zgornjega soseda ($y_n = 0$), da dobimo element y od \mathbf{g} . Zato ocenimo gradient centra celice ob upoštevanju najmanjšega diskretnega potenciala, ki velja za center celice na podlagi obeh sosednjih celic na osi x ali y .

Označimo interpoliran gradient za center celice (podobno kot interpoliran potencial v (1)) z $\mathbf{h}_{cr} = [hx_{cr}, hy_{cr}]^T$, kjer velja $c, r \in \{0, 1\}$. Za celico s središčnimi koordinatami $x_m = x_0 + d_c c$, $y_m = y_0 + d_c r$ se gradient za $U(x_m, y_m) < \infty$ glasi

$$\mathbf{h}_{cr} = -\frac{1}{d_c} \begin{bmatrix} e(U(x_m + d_c e, y_m) - p_{cr}) \\ f(U(x_m, y_m + d_c e) - p_{cr}) \end{bmatrix}, \quad (5)$$

v primeru $U(x_m, y_m) = \infty$ pa je

$$\mathbf{h}_{cr} = -\frac{1}{d_c} \begin{bmatrix} S_x(U(x_m + d_c S_x, y_m) - p_{cr}^*) \\ S_y(U(x_m, y_m + d_c S_y) - p_{cr}^*) \end{bmatrix}, \quad (6)$$

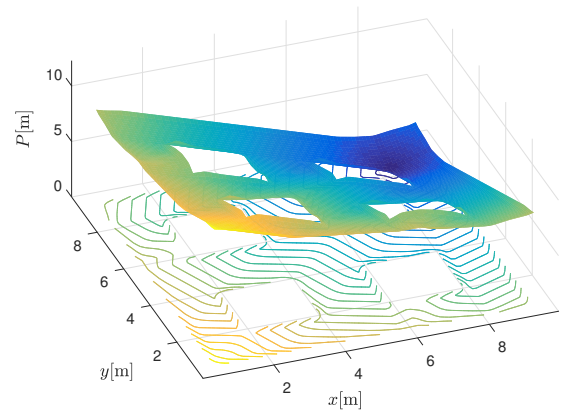
pri čemer je

$$e = \underset{s}{\operatorname{argmin}} U(x_m + d_c s, y_m); s \in \{-1, 0, 1\},$$

$$f = \underset{s}{\operatorname{argmin}} U(x_m, y_m + d_c s); s \in \{-1, 0, 1\}, \quad (7)$$

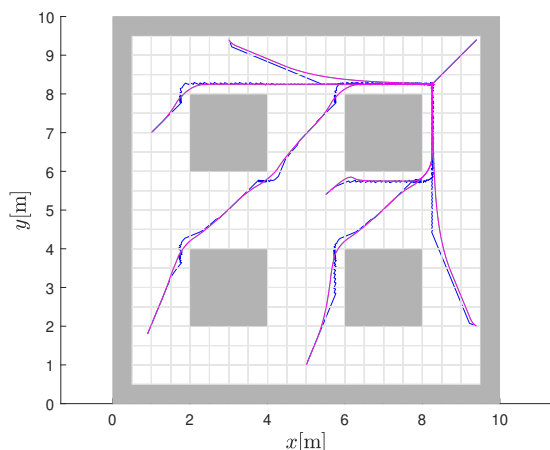
$S_x = \operatorname{sgn}(x - x_m)$ in $S_y = \operatorname{sgn}(y - y_m)$ ter $\operatorname{sgn}(\cdot)$ označuje funkcijo predznaka. Enačba (5) se nanaša na primer, ko je celica prosta, enačba (6) pa na primer, ko je celica zasedena. Če je celica zasedena ($U(x_m, y_m) = \infty$), se njen potencial posodobi, označeno kot p_{cr}^* v (6), iz že znanega potenciala $P(x, y)$ in gradienta $\mathbf{g}(x, y)$ v trenutnem položaju $[x, y]^T$:

$$p_{cr}^* = P(x, y) + \mathbf{g}(x, y) \times [x_m - x, y_m - y]^T.$$



Slika 3: Interpolirano potencialno polje $P_n(x_n, y_n)$ s konturami enakega potenciala (temnejše barve označujejo nižje vrednosti), dobljeno z bilinearno interpolacijo za okolje na sliki 1

Pred izračunom gradientov celic v enačbah (5) do (7) moramo preveriti, ali je katera od sosednjih celic zunaj okolja. Če je temu tako, se potencial te celice rekonstruira



Slika 4: Dobljene poti s potovanjem v smeri negativnega gradienta (modra črta) in interpoliranega negativnega gradienta (vijolična črta) potencialnega polja s slike 3

podobno kot v (4) ob upoštevanju znanega interpoliranega potenciala $P(x, y)$ in gradienta $\mathbf{g}(x, y)$ za točko $[x, y]^T$.

Iz ocenjenih gradientov celic \mathbf{h}_{00} , \mathbf{h}_{01} , \mathbf{h}_{10} in \mathbf{h}_{11} (enačbi (5) in (6)), se končni interpolirani gradient v trenutnem položaju določi z

$$\mathbf{h}(x, y) = w_{00}\mathbf{h}_{00} + w_{01}\mathbf{h}_{01} + w_{10}\mathbf{h}_{10} + w_{11}\mathbf{h}_{11},$$

kjer so uporabljene enake uteži w_{00} , w_{01} , w_{10} , w_{11} kot v (2). Primerjava dobljenih poti z gradientom \mathbf{g} in izboljšanim interpoliranim gradientom \mathbf{h} je prikazana na sliki 4. Dobljene poti s sledenjem interpoliranega gradienta potencialnega polja so gladke in ne trčijo ob ovire.

Bilinearne interpolacije je torej mogoče elegantno uporabiti za določitev zveznega potencialnega polja in njegovega gradienta na podlagi zemljevida z mrežo diskretnih cen (diskretno potencialno polje). To omogoča določitev ustreznih smeri vožnje, brez lokalnih minimumov, in poti, ki ne trčijo ob ovire in imajo zvezni potek, kar je pomembno za robotska vozila. Omeniti velja, da bi lahko bikubična interpolacija [9, 10] ustvarila tudi bolj gladko interpolacijo, vendar zahteva soseščino šestnajstih celic, kar je problematično v bližini ovir ali v ozkih koridorjih, saj zahtevajo zasedene celice posebno obdelavo, preden se uporabijo v interpolaciji. In tudi sicer lahko v tem primeru pride do lokalnih minimumov. Zato za načrtovanje gladkih poti predlagamo uporabo bilinearne interpolacije z ustrežno predobdelavo zasedenih celic in z dodatno interpolacijo gradienta. Dobljena pot je gladka, hkrati je tudi intuitivna in skoraj optimalna. Kot smo že omenili, v potencialnem polju, zaradi izbranega načina njegove konstrukcije, ne bo lokalnih minimumov, ki so lahko prisotni v klasični formulaciji umetnega potencialnega polja.

4 Zaključek

Delo predlaga nov pristop za določitev navigacijske funkcije, ki je različica umetnega potencialnega polja. Mogoče jo je uporabiti za navigacijo, načrtovanje poti in vodenje mobilnega robota. Navigacijska funkcija zagotavlja varno vodenje do cilja brez lokalnih minimumov potencialnega

polja, kar je pogosta težava pri umetnih potencialnih poljih. Optimalnost in konvergenca sta podedovani iz optimalnega iskanja na mreži celic, ki vrne diskretno potencialno polje.

Za doseg gladke navigacijske funkcije in ustreznih smeri vožnje v smeri padajočega gradienta, predlagamo bilinearne interpolacije z več novimi razširitvami, kar omogoča učinkovito uporabo pri načrtovanju poti. Najprej predlagamo določitev diskretnega potenciala za celice, ki jih potrebujemo za interpolacijo in pripadajo oviram ali so zunaj okolja, na podlagi njihove soseščine. Nadalje uvedemo dodatno interpolacijo gradienta iz ocenjenega diskretnega gradienta interpoliranih celic. To vodi do gladke in skoraj optimalne poti brez trkov oziroma navigacije robota proti cilju. Predlagani interpolacijski pristop je mogoče izvajati v sprotne načinu, saj je računsko učinkovit, ker lahko interpolira tudi le diskretne potencialne celice vzdolž načrtovane poti.

Zahvala

Avtorji se zahvaljujejo za finančno podporo Agencije Republike Slovenije za raziskovalno dejavnost in podjetju Epilog d.o.o. (št. L2-3168 in P2-0219).

Literatura

- [1] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.
- [2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [3] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, p. 269–271, dec 1959.
- [4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [5] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.
- [6] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE conference on decision and control (CDC)*, pp. 7681–7687, IEEE, 2010.
- [7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C: the art of scientific computing (2nd ed.)*. New York, USA, 1992.
- [8] "Bilinear interpolation." https://en.wikipedia.org/wiki/Bilinear_interpolation. dostopano: 15. 2. 2022.
- [9] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.
- [10] G. Klančar and M. Seder, "Coordinated multi-robotic vehicles navigation and control in shop floor automation," *Sensors*, vol. 22, no. 4, 2022.