

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 108

**UPRAVLJANJE BESPILOTNOM LETJELICOM ZA
PRONALAZAK OBJEKTA U 3D PROSTORU NA
TEMELJU SEMANTIKE PROSTORA**

Marin Maletić

Zagreb, Veljača, 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 108

**UPRAVLJANJE BESPILOTNOM LETJELICOM ZA
PRONALAZAK OBJEKTA U 3D PROSTORU NA
TEMELJU SEMANTIKE PROSTORA**

Marin Maletić

Zagreb, Veljača, 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 30. rujna 2024.

DIPLOMSKI ZADATAK br. 108

Pristupnik: **Marin Maletić (0036516907)**

Studij: Informacijska i komunikacijska tehnologija

Profil: Automatika i robotika

Mentor: prof. dr. sc. Stjepan Bogdan

Zadatak: **Upravljanje bespilotnom letjelicom za pronađazak objekta u 3D prostoru na temelju semantike prostora**

Opis zadatka:

Potrebno je implementirati metodu upravljanja za bespilotnu letjelicu koja će omogućiti da letjelica u poznatom prostoru, u što kraćem vremenu, pronađe traženi objekt. Objekt koji je potrebno pronaći zadaje se opisno. Primjeri ulaznih specifikacija su "plava knjiga" ili "lopta ispod stola". Letjelica je opremljena kamerom, a pretpostavka je da letjelica zna svoju poziciju (što će biti ostvareno pomoću Optitrack Trio sustava). Potrebno je implementirati metodu za prepoznavanje objekata u prostoru iz slike s kamere (npr. YOLO). Također je potrebno ostvariti autonomno kretanje letjelice u 3D prostoru pomoću postojećih algoritama za planiranje i izvođenje trajektorije. Osmisliti metodu za pretraživanje prostora na temelju prepoznatih objekata tj. semantike prostora kako bi se pronašao traženi objekt. Metodu provjeriti u simulaciji te na stvarnom sustavu za različita okruženja. Usporediti učinkovitost razvijene metode s metodom koja ne uzima u obzir semantiku.

Rok za predaju rada: 14. veljače 2025.

Zahvaljujem mentoru prof. dr. sc. Stjepanu Bogdanu na mentorstvu i ukazanom povjerenju, izv. prof. dr. sc. Tamari Petrović na stručnom vodstvu te asistentici Marijani Peti na svim odgovorenim pitanjima tijekom izrade ovog rada.

Sadržaj

1. Uvod	3
2. Opis problema i ideja rješenja	4
3. Pregled korištenih tehnologija	6
3.1. Bespilotna letjelica Crazyflie	6
3.2. OptiTrack sustav za pozicioniranje	7
3.3. Livox MID-360 LiDAR	8
3.4. Cartographer SLAM i CloudCompare	9
3.5. OctoMap	9
3.6. A* algoritam pretraživanja	10
3.7. Polinomsko generiranje trajektorije	12
3.8. YOLOv8 Neuronska Mreža	13
3.9. Veliki jezični modeli (LLM)	13
4. Metodologija	15
4.1. Prepostavke i predradnje	15
4.1.1. Mapiranje prostora	15
4.2. Arhitektura sustava	16
4.2.1. Obrada slike i procjena pozicije objekata	17
4.2.2. Interpretacija semantike prostora	19
4.2.3. Načini skeniranja prostora	20
4.3. Strategija pretraživanja prostora	21
4.3.1. Pregled strategije	21
5. Simulacija	24

5.1. Generiranje tlocrta i tekstualnog opisa prostora	25
5.2. Scenarij I	26
5.2.1. Korisničke upute	26
5.2.2. Pretraživanje	26
5.3. Scenarij II	28
5.3.1. Korisničke upute	28
5.3.2. Pretraživanje	28
5.4. Scenarij III	30
5.4.1. Korisničke upute	30
5.4.2. Pretraživanje	31
5.5. Rezultati	32
6. Eksperiment u laboratorijskom okruženju	35
7. Zaključak	38
Literatura	39
Sažetak	41
Abstract	42

1. Uvod

U posljednjih nekoliko godina svjedočimo sve intenzivnijem razvoju tehnologije bespilotnih letjelica, što je izravno povezano s dostupnošću kvalitetnih senzora, sve snažnijih procesora te napretkom u područjima računalnog vida i umjetne inteligencije. Bespilotne letjelice više se ne koriste samo u vojne svrhe, već pronalaze širu primjenu u komercijalnim, znanstvenim i civilnim područjima – od nadgledanja usjeva u poljoprivredi, inspekcija energetskih postrojenja, snimanja filmova i fotografija, do kompleksnijih zadataka pretraživanja i spašavanja.

Jedna od ključnih prednosti bespilotnih letjelica je njihova sposobnost da pristupe teško dostupnim ili opasnim područjima, u kojima bi ljudsko djelovanje bilo previše rizično ili neisplativo. Stoga se u brojnim istraživanjima razvijaju napredni autonomni sustavi koji omogućuju bespilotnoj letjelici da pretražuje i mapira prostor bez neposrednog ljudskog upravljanja.

U okviru ovog rada, fokus je na razvoju sustava koji omogućuje bespilotnoj letjelici Crazyflie da autonomno pretražuje prostor u potrazi za definiranim ciljem – primjerice, pronađak određenog objekta. Ova vrsta problema omogućuje usporedbu dvaju pristupa: nasumičnog i ciljanog pretraživanja. Dok bi nasumično pretraživanje moglo učinkovito obići cijeli prostor (posebno ako su dimenzije ograničene), ciljano pretraživanje koristi semantiku objekata u neposrednoj blizini ili kontekstualnu informaciju o lokaciji predmeta i time značajno skraćuje vrijeme pretraživanja. Istražila bi se primjena velikih jezičnih modela (engl. Large Language Model - LLM) kao potpora pretraživanju prostora u smislu razumijevanja okoline letjelice i razumijevanja korisnikovih uputa o traženom objektu.

2. Opis problema i ideja rješenja

Glavni je cilj ovog rada opisati, razviti i eksperimentalno validirati sustav koji omogućuje:

- Samostalano polijetanje i stabilan let uz korištenje OptiTrack sustava za pozicioniranje.
- Detekciju objekata u stvarnom vremenu pomoću YOLOv8 neuronske mreže.
- Planiranje putanje i izbjegavanje prepreka u 3D prostoru korištenjem A* algoritma i oktomape.
- Generiranje glatke trajektorije (polinomske krivulje) temeljene na točkama koje vraća A* algoritam.
- Pametno usmjeravanje letjelice prema potencijalnim lokacijama traženog predmeta s ciljem smanjenja ukupnog vremena pretraživanja i poboljšanja učinkovitosti.

Uz senzore za percepciju udaljenosti prepreka, kamera je ključan dio autonomne navigacije kroz prostor. Neuronske mreže za prepoznavanje objekata mogu obraditi sliku kamere te u stvarnom vremenu dati informaciju koji se predmeti nalaze u blizini letjelice. Veliki jezični modeli (LLM) mogu se iskoristiti kako bi se uspostavila logika pretraživanja prostora. Lista detektiranih objekata i njihove koordinate mogu se poslati modelu kao unaprijed definiran *prompt*¹ te bi kao odgovor model izabrao relevantne objekte, odnosno one prema kojima bi se trebalo kretati ukoliko se želi povećati šansa pronađaska traženog predmeta. Primjerice, ako je traženi objekt tipkovnica, s obzirom na njezinu veličinu, moguće je da u inicijalnoj slici prostora ona neće biti vidljiva, ali možda hoće radni stol s računalom (među ostalim stvarima). Model vraća računalo kao predmet u

¹Tekst u prirodnom jeziku koji opisuje zadatak koji umjetna inteligencija (AI) treba obaviti

čijoj blizini je najveća vjerojatnost pronalaska tipkovnice i prema njemu će se orijentirati daljnja pretraga.

Dodatna mogućnost je iskoristiti LLM za obradu zahtjeva korisnika i danih informacija o traženom predmetu. Primjerice, ako je potrebno pronaći naočale za vid, upute mogu sadržavati informacije poput kada se najčešće nose i slično. Model obrađuje korisnikov opis problema te zaključuje što je traženi objekt i gdje bi mogao biti. Ovakvo ciljano pretraživanje omogućuje letjelici planiranje putanje čime smanjuje vrijeme potrebno za pronalazak zadanog predmeta, a korisniku daje mogućnost komunikacije s letjelicom prirodnim jezikom, a ne samo pomoću jasno definiranih uputa.

Sustav je predviđen za pretragu konveksnih prostora manjih dimenzija čije lokacije prepreka unaprijed poznajemo. To znači da je prostor prije leta mapiran i informacije o preprekama su spremljene. Ovo proizlazi iz ograničenja korištene letjelice koja je malih dimenzija te nošenje kvalitetnog senzora za trodimenzionalno mapiranje nije moguće. Slijedno tome, putanja se planira pod pretpostavkom da su prepreke statične i stalne. Ograničenje područja leta predstavljaju vidna polja kamera koje se, uz reflektirajuće oznake na letjelici, koriste za njeno lociranje, pa je let moguć samo u području gdje se vidna polja barem tri kamere isprepliću. Također, kamera na Crazyflie letjelici je monokromatska kamera rezolucije 324x324 piksela, što u letu s pravom letjelicom može izazvati poteškoće prepoznavanju objekata. Letjelica sliku šalje preko WiFi mreže na prijenosno računalo gdje ju obrađuje neuronska mreža, te ta veza isto može biti nestabilna. Ovih ograničenja u simulaciji nema.

U sklopu rada bit će prikazan cjelokupni postupak razvoja, od teorijske pozadine i pregleda korištenih tehnologija, preko implementacije i integracije raznih softverskih i hardverskih komponenti, do eksperimentalnih rezultata i evaluacije sustava u simulaciji i u laboratorijskom okruženju.

3. Pregled korištenih tehnologija

U ovom poglavlju opisane su tehnologije i alati korišteni u razvoju metode pretraživanja prostora. Prikazana je bespilotna letjelica i njeni senzori te LiDAR korišten za mapiranje prostora. Također, opisane su tehnologije mapiranja i prostorne podjele kao što su oktalna stabla, Yolov8 neuronska mreža za detekciju objekata, A* (*A-star*) algoritam pretraživanja za pronašetak putanje, polinomsko planiranje trajektorije i ChatGPT kao LLM model za obradu uputa i listi prepoznatih objekata.

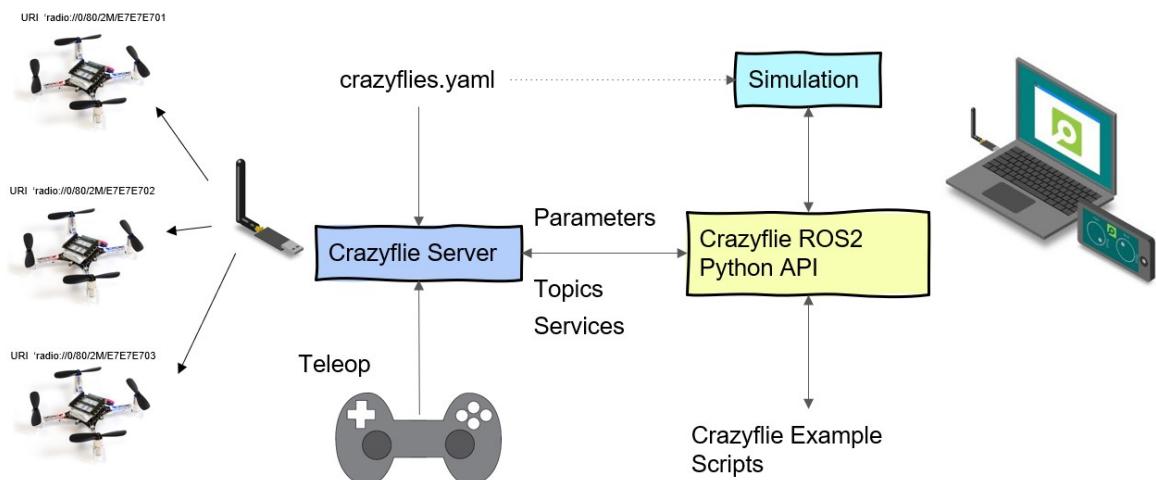
3.1. Bespilotna letjelica Crazyflie

Bespilotna letjelica korištena u ovome radu je kompaktna, lagana i modularna Crazyflie 2.1 letjelica švedske tvrtke Bitcraze^[1], prikazana slikom 3.1. Zahvaljujući svojoj otvorenoj arhitekturi i mogućnosti prilagodbe, Crazyflie se široko koristi u istraživačkim laboratorijima i akademskim ustanovama. Ručno upravljanje letjelicom moguće je putem mobilne aplikacije koristeći Bluetooth vezu ili računalom koristeći Crazyradio 2.0, wifi adapter za komunikaciju letjelice i računala. CFLib je API napisan u Pythonu koji se koristi za komunikaciju s Crazyflie letjelicama. Omogućuje slanje naredbi letjelicama pomoću Python koda, a u ovom radu korišten je i ROS 2 (*Robot Operating System*) kojim je uspostavljeno autonomno upravljanje letjelicom preko ROS servisa. Crazyflie 2.1 teži neznatnih 27 grama što je čini odličnom za eksperimentalne letove u zatvorenim prostorima jer je rizik od oštećenja prilikom sudara mali. To je modularna platforma, te se može nadograditi dodatnim senzorima, računalnim platformama i kontrolnim jedinicama koji proširuju funkcionalnosti letjelice. Tako je u ovome radu dodan *AI Deck* s monokromatskom kamerom i reflektivni marker koji služi određivanju točne pozicije u 3D prostoru. S obzirom na veličinu i korištene senzore, vrijeme leta je svega par minuta pa je neprekinituti let kratak, ali baterije su lako zamjenjive što omogućuje duže testiranje.



Slika 3.1. Crazyflie 2.1 s AI Deck-om i reflektivnim markerima

U ovome radu algoritam je prvotno testiran u Crazyswarm2 simulatoru [2] koji integrira Gazebo Garden simulaciju, model Crazyflie letjelice, python biblioteke te ROS2 poruke i servise. ROS2 servisi poput *Takeoff* / *Land* / *GoTo* i *Upload* / *Start trajectory* omogućuju jednostavno upravljanje letjelicom, a ista računalna pozadina se koristi i pri upravljanju stvarnom letjelicom pa je selidba iz simulacije na eksperiment vrlo jednostavna. Kako bi instalacija, postavljanje i migracija okruženja bila jednostavnija, algoritam je razvijan unutar Docker kontejnera.



Slika 3.2. Pregled komunikacije između čvorova unutar Crazyswarm2 [2]

3.2. OptiTrack sustav za pozicioniranje

OptiTrack je jedan od vodećih optičkih sustava za precizno praćenje pokreta koji se široko koristi u robotici, animaciji, virtualnoj stvarnosti i industrijskoj automatizaciji. Ovaj

sustav koristi infracrvene kamere visoke rezolucije kako bi pratio objekte u stvarnom vremenu, omogućujući preciznu lokalizaciju dronova, robota i drugih pokretnih tijela u trodimenzionalnom prostoru.

U ovom radu, OptiTrack se koristi za pozicioniranje Crazyflie letjelice unutar kontroliranog eksperimentalnog okruženja. Sustav prikuplja podatke s više kamera i koristi algoritme triangulacije za precizno određivanje pozicije svakog markera, a kako bismo mogli sa sigurnošću odrediti položaj triangulacijom, u svakom trenutku letjelica mora biti u vidnom polju barem tri kamere. Budući da Crazyflie nema ugrađeni GPS, OptiTrack i reflektivni marker na letjelici omogućuju preciznost pozicioniranja, a IMU ugrađen na letjelici preciznost orientacije, čime se postiže stabilan i kontroliran let.

3.3. Livox MID-360 LiDAR

LiDAR (Light Detection and Ranging) je napredna senzorska tehnologija koja koristi pulsirajuće laserske zrake za precizno mjerjenje udaljenosti do objekata u prostoru. Ova tehnologija omogućuje vrlo pouzdano trodimenzionalno mapiranje okoline i često se koristi u autonomnim vozilima i robotici.

U ovom radu koristi se Livox MID-360, kompaktan i visoko precizan LiDAR, u kombinaciji sa svojim ROS driver-om koji omogućava snimanje izlaznih PointCloud2 i IMU poruka. U usporedbi s klasičnim LiDAR senzorima koji koriste mehaničko skeniranje (npr. Velodyne serija), Livox koristi nesekvencijalni sken (engl. Non-Repetitive Scanning), što omogućuje gušće i učinkovitije uzorkovanje prostora tijekom vremena. Prije samog leta, cijeli prostor u kojem će se letjeti mapiran je i podaci su spremljeni u *ROS bag*.



Slika 3.3. Livox MID-360 LiDAR [3]

3.4. Cartographer SLAM i CloudCompare

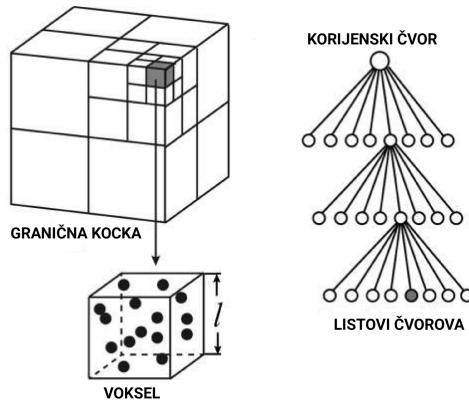
Nakon snimljenog oblaka točaka (engl. point cloud) prostora, vrlo je bitno te podatke pretvoriti u koristan oblik. Simultana lokalizacija i mapiranje (SLAM – Simultaneous Localization and Mapping) jedna je od ključnih tehnika u autonomnim robotskim sustavima. SLAM omogućuje robotima i bespilotnim letjelicama da istovremeno kreiraju mapu okoline i određuju vlastitu poziciju unutar tog prostora. Za potrebe ovog diplomskog rada, korišten je Cartographer SLAM[4][5], open-source sustav razvijen od strane Google Research-a, namijenjen za 2D i 3D mapiranje pomoću očitanja LiDAR-a. Cartographer koristi graf-bazirani SLAM algoritam, gdje se podaci iz LiDAR-a kontinuirano uspoređuju kako bi se poboljšala preciznost mape. To omogućuje ispravljanje pogrešaka u lokalizaciji i spajanje više skenova u konzistentnu 3D mapu.

Kvaliteta dobivene mape prostora može biti kompromitirana utjecajem uvijek prisutnog šuma pa ju je potrebno očistiti koristeći filtre. *Statistical Outlier Removal* (SOR) uklanja izolirane točke na temelju statističke analize njihovih udaljenosti od susjednih točaka. U tu svrhu korišten je CloudCompare, besplatni open-source softver specijaliziran za analizu, usporedbu i obradu 3D oblaka točaka koji pruža sve potrebne alate za uređenje mape [6].

3.5. OctoMap

Oktalna stabla[7] predstavljaju hijerarhijski način pohrane podataka. Trodimenzionalni prostor dijeli se na čvorove, od kojih svaki ima osam manjih podprostora (engl. voxel cells) te se proces ponavlja rekursivno do željene rezolucije [3.4]. Svaka ćelija može biti slobodna, zauzeta ili neodređena. Prostori koji su prazni ili homogeni mogu se sažeti u hijerarhijski više čvorove odnosno roditelje. Time se smanjuje potrebna memorija, a struktura stabla omogućuje brzo pretraživanje čvorova.

OctoMap[9] je probabilistički trodimenzionalni mapirajući sustav temeljen na oktalnim stablima. Omogućuje efikasno pohranjivanje, pretraživanje i obradu trodimenzionalnih podataka. Razvijen je prvenstveno za robotske sustave kako bi omogućio učinkovito mapiranje prostora pomoću senzora poput LiDAR-a ili stereo kamera, a u ovom radu korišten je za stvaranje okupiranih i slobodnih područja, pružajući bespilotnoj le-



Slika 3.4. 3D prostorna dekompozicija za oblake točaka temeljena na oktalnim stablima [8]

tjelici mogućnost izbjegavanja prepreka i planiranja putanje.

3.6. A* algoritam pretraživanja

A* (*A star*) algoritam pretraživanja jedan je od najčešće korištenih algoritama za pronađazak optimalne putanje u grafovima i prostornim mrežama. Razvijen je kao poboljšana verzija Dijkstrinog algoritma [10], kombinirajući najkraći poznati put do trenutne točke s heurističkom procjenom udaljenosti od te točke do cilja. Ovaj pristup omogućuje brže i učinkovitije pretraživanje u usporedbi s klasičnim algoritmima. A* često se koristi u robotici, računalnim igrama i navigacijskim sustavima.

U ovom radu algoritam se koristi za planiranje putanje u trodimenzionalnom prostoru na temelju *OctoMap* reprezentacije okoline kako bi letjelica Crazyflie mogla sigurno navigirati i izbjegavati prepreke.

Prostor je prvotno potrebno diskretizirati, odnosno umjesto koordinata u metrima prelazimo u reprezentaciju prostora rešetkom. Veličina ćelija ovisi o željenoj rezoluciji - veća rezolucija znači više ćelija i ujednačeniji put, ali iziskuje više memorije i računalne snage. A* algoritam za svaku točku u prostoru (čvor u grafu) izračunava trošak prelaska koristeći sljedeću funkciju:

$$f(n) = g(n) + h(n) \quad (3.1)$$

gdje je:

- $g(n)$ – trošak dolaska do čvora n iz početne točke (najkraći put do trenutne lokacije),

- $h(n)$ – heuristička procjena udaljenosti čvora n do cilja (npr. Euklidska udaljenost),
- $f(n)$ – ukupni trošak čvora n , koji se koristi za odabir sljedeće najbolje točke u pretraživanju.

Kreće od početne točke te se pronalaze svi njezini susjedi u trodimenzionalnom prostoru, njih 26. Zatim se izračunava ukupni trošak te točke prema jednadžbi (3.1) i odabire se točka s najmanjim troškom za daljnje širenje. Proces se ponavlja dok se ne dosegne ciljna točka ili maksimalan broj iteracija. Na kraju, ako je put pronađen, rekonstruiraju se njegove točke.

Algorithm 1 A* Algoritam za pretragu najkraćeg puta

Require: Startna pozicija $start$, Ciljna pozicija $goal$

Ensure: Najkraći put od $start$ do $goal$

```

1:  $openSet \leftarrow \{start\}$                                 ▷ Čvorovi koji se trebaju istražiti
2:  $cameFrom \leftarrow \emptyset$                                ▷ Pamćenje najkraćih puteva
3:  $gScore[start] \leftarrow 0$                                 ▷ Trošak dolaska do čvora
4:  $fScore[start] \leftarrow h(start, goal)$                   ▷ Procijenjeni trošak
5: while  $openSet \neq \emptyset$  do
6:    $current \leftarrow$  čvor u  $openSet$  s najmanjom vrijednosti  $fScore$ 
7:   if  $current = goal$  then
8:     return RekonstruirajPut( $cameFrom, current$ )
9:   end if
10:  for all  $neighbor$  in Susjedi( $current$ ) do
11:     $temp\_gScore \leftarrow gScore[current] + dist(current, neighbor)$ 
12:    if  $neighbor \notin gScore$  or  $temp\_gScore < gScore[neighbor]$  then
13:       $cameFrom[neighbor] \leftarrow current$ 
14:       $gScore[neighbor] \leftarrow temp\_gScore$ 
15:       $fScore[neighbor] \leftarrow gScore[neighbor] + h(neighbor, goal)$ 
16:      if  $neighbor \notin openSet$  then
17:        Dodaj  $neighbor$  u  $openSet$ 
18:      end if
19:    end if
20:  end for
21: end while
22: return Ne postoji put do cilja

```

Kod 3.1. Pseudokod algoritma za pretragu najkraćeg puta

U ovom radu, A* koristi oktomapu za prepoznavanje slobodnih i zauzetih prostora, kako bi letjelica mogla planirati sigurnu putanju kroz okruženje bez sudara s preprekama. Funkcija provjerava svaku susjednu točku nalazi li se oko nje u granicama graničnog okvira (engl. bounding box) neka zauzeta ćelija (engl. voxel). Varijablom "cle-

arance" možemo upravljati kolika treba biti veličina tog okvira što je ujedno i sigurnosna udaljenost od prepreka.

3.7. Polinomsko generiranje trajektorije

Planiranje putanje pomoću A* algoritma daje niz diskretnih točaka u prostoru koje predstavljaju najkraći put od početne do ciljane točke. Međutim, ova putanja nije pogodna za direktno praćenje bespilotnom letjelicom, jer su točke međusobno naglo povezane, što može dovesti do nestabilnog leta i naglih promjena smjera. Polinomijalno generiranje trajektorije osigurava da letjelica može kontinuirano i polako mijenjati položaj, brzinu i akceleraciju, što omogućuje precizniju kontrolu leta.

Tradicionalne metode generiranja putanje često zanemaruju derivacije položaja višeg reda. Metoda polinomijalnih spline-ova omogućava da trajektorija prođe kroz sve točke putanje tako što izračunavamo polinome za svaki dio, odnosno između uzastopnih točaka. Za N točaka putanja je podijeljena na N-1 polinomskih segmenata gdje svaki segment ima polinom sedmog reda (3.2) za svaku prostornu os (x, y, z) te yaw.

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7 \quad (3.2)$$

$$x_0..x_7, y_0..y_7, z_0..z_7, yaw_0..yaw_7 \quad (3.3)$$

Glavni zahtjev je osigurati dovoljno stupnjeva slobode da se zadovolje rubni uvjeti (položaj, brzina, ubrzanje, trzaj)[11]. Imajući na umu da prijelaz iz svakog polinoma između točaka p_i i p_{i+1} u sljedeći polinom (odnosno točno na poziciji točaka) bude stalne brzine i bez zaustavljanja, treba se definirati željena brzina (određuje trajanje T pojedinog segmenta) te postaviti sljedeće rubne uvjete u $t = 0$ i $t = T$:

$$\begin{aligned} p(0) &= p_i, & p(T) &= p_{i+1} \\ p'(0) &= v_i, & p'(T) &= v_{i+1} \\ p''(0) &= 0, & p''(T) &= 0 \\ p'''(0) &= 0, & p'''(T) &= 0 \end{aligned}$$

Nadalje, jednadžbe se mogu složiti u matrični račun $Aa = b$, gdje je A matrica 8×8 sa redovima koji odgovaraju koeficijentima polinoma u točkama $t = 0$ i $t = T$, b je matrica 1×8 s rubnim uvjetima te a predstavlja tražene koeficijente polinoma. Konačno rješenje računa se rješavanjem linearne algebarske jednadžbe. Na kraju, za svaki segment dobiju se 32 koeficijenta (8 koeficijenata za x, y i z) i vrijeme T u kojem je potrebno izvršiti taj dio trajektorije. Radi lakšeg snalaženja, koeficijenti se zapisuju u .csv datoteku iz koje se čitaju kada se trajektorija vizualizira ili šalje bespilotnoj letjelici.

3.8. YOLOv8 Neuronska Mreža

YOLO (*You Only Look Once*) jedna je od najpopularnijih neuronskih mreža za detekciju objekata u stvarnom vremenu. YOLO mreže poznate su po velikoj brzini i točnosti, omogućujući prepoznavanje više objekata na slici u jednom prolazu kroz mrežu.

U ovome radu korištena je inačica algoritma, YOLOv8 [12] tvrtke Ultralytics. Kao i svi prethodnici, trenirana je na skupu podataka MS COCO [13] što mreži omogućuje prepoznavanje 80 različitih objekata uključujući osobe, vozila, namještaj, životinje i svakodnevne predmete. To je čini idealnim odabirom za potrebe detekcije predmeta u ovome radu.

3.9. Veliki jezični modeli (LLM)

Za obradu prirodnog jezika i donošenje odluka temeljenih na podacima iz okoline korišteni su gotovi jezični modeli. ChatGPT API [14] omogućava generiranje tekstualnih odgovora, analizu odnosa objekata u prostoru te optimizaciju kretanja, a obostrana komunikacija ostvarena je putem Python koda. U sustavu razvijenom za ovaj rad, ChatGPT je korišten u nekoliko ključnih segmenata:

- Opisivanje prostora – Model analizira detektirane objekte i njihovu prostornu raspodjelu kako bi generirao deskriptivne sažetke okruženja.
- Donošenje odluka o kretanju – Na temelju podataka prikupljenih pomoću senzora i algoritama za prepoznavanje objekata, LLM pomaže pri odlučivanju o sljedećim koracima u navigaciji letjelice.

- Određivanje ciljnog objekta – LLM može interpretirati tekstualne upute korisnika i odrediti koji objekt u prostoru odgovara traženom kriteriju ili usmjeravati letjelicu.

Kombinacija metoda strojnog učenja i heurističkih algoritama omogućuje učinkovitu navigaciju i interakciju letjelice s okolinom. API je integriran u postojeći sustav pomoću OpenAI SDK-a, a koristeći jasno definiranu zadaću LLM-a i strukturu odgovora u JSON formatu, moguće je potpuno automatizirati komunikaciju između letjelice i modela.

4. Metodologija

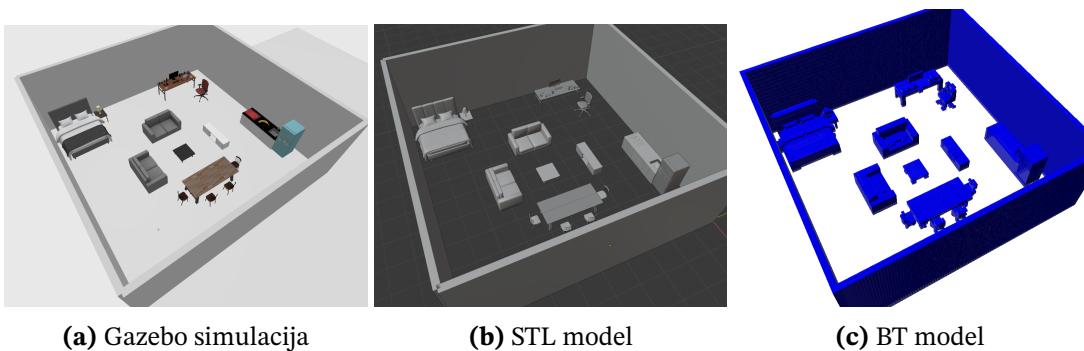
4.1. Prepostavke i predadnje

Uzimajući u obzir ograničenja prostora za letenje i same letjelice, sustav je razvijan uz sljedeće prepostavke:

- Prostor je konveksan, odnosno svi segmenti pravca između bilo koje dvije točke unutar prostora također su u potpunosti unutar tog prostora. Ovo je važno zbog načina određivanja središta prostora pretraživanja.
- Mapa prostora poznata je prije početka leta.

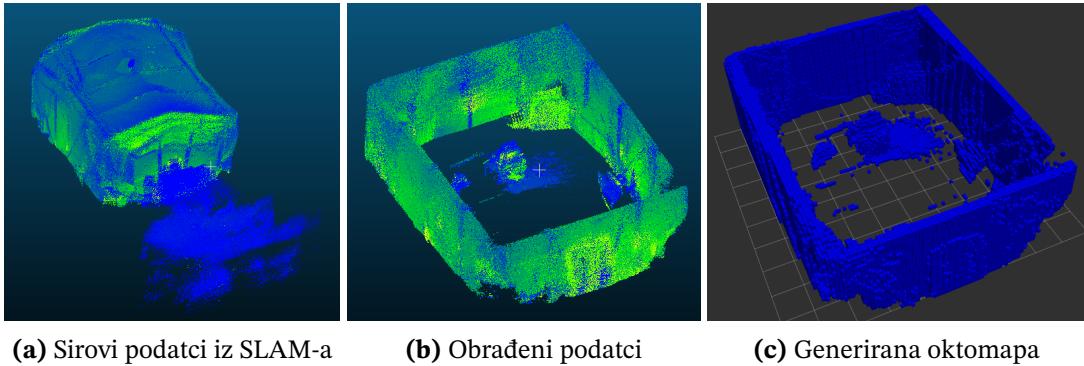
4.1.1. Mapiranje prostora

Kako bi letjelica mogla izbjegavati prepreke i projicirati semantiku iz slike u prostorne položaje, potrebno je izraditi mapu prostora prije leta. Za testiranje u Gazebo simulatoru, stvoren je jednostavan prostor vidljiv na Slici 4.1.a Pomoću alata Blender [15] i Binvox[16], stvorena je binarna reprezentacija prostora gdje je prostor diskretiziran u čelije (tzv. voxeli) koje su ili popunjene (1) ili slobodne (0). Takva mapa koristit će se kao oktomapa prostora.



Slika 4.1. Postupak kreiranja binarne reprezentacije prostora

U svrhu mapiranja prostorije za let na stvarnoj letjelici, korišten je Livox MID-360 LiDAR koji u sebi sadrži IMU senzor pa na svoj izlaz daje i njegova očitanja uz oblak točaka. Snimljeni podaci obrađuju se koristeći Cartographer ROS koji generira 3D mapu prostora. Ona se dalje obrađuje uklanjajući šum, razna pogrešna očitanja, strop i pod (radi lakše vizualizacije leta u Rviz alatu). Postupak obrade oblaka točaka i generiranja oktomape prostora prikazan je na slici 4.2.

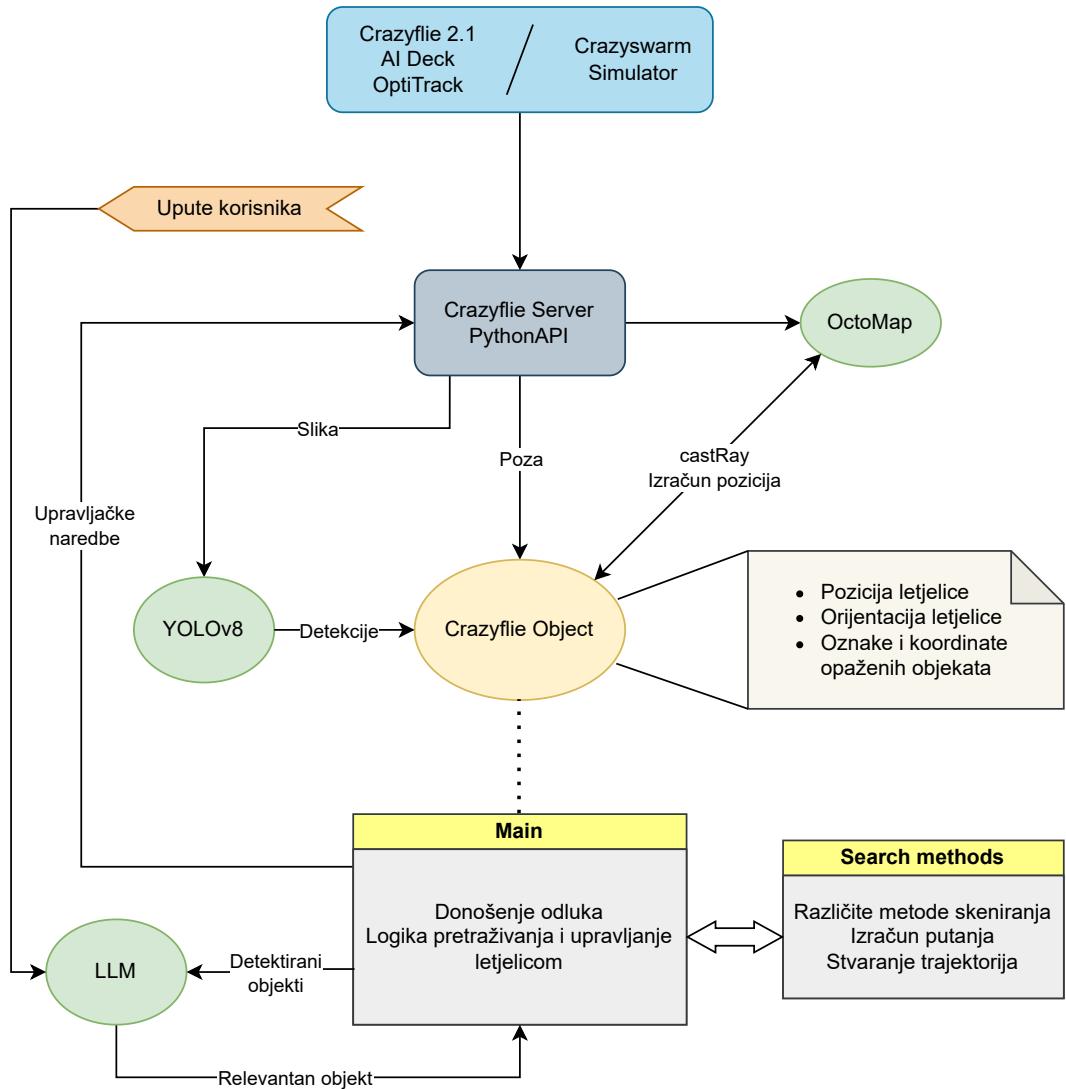


Slika 4.2. Postupak kreiranja oktomape prostora

4.2. Arhitektura sustava

Graf prikazan slikom 4.3. ilustrira arhitekturu sustava za autonomnu navigaciju i pretragu prostora pomoću letjelice Crazyflie 2.1. Koristi se kombinacija različitih modula za obradu podataka, planiranje putanje, kreiranje trajektorije i donošenje odluka. Sustav započinje s Crazyflie Server PythonAPI, koji služi kao središnji upravljački čvor i omogućuje upravljanje letjelicom pomoću Python koda, bilo u stvarnom okruženju (OptiTrack) ili simuliranom (CrazySim). OctoMap omogućuje memorijски učinkovito mapiranje prostora, identifikaciju slobodnih i zauzetih područja te projekciju prepoznatih objekata iz kamere u njihove koordinate u trodimenzionalnom prostoru. Oktomapa se stvara prijenosom unaprijed stvorene binarne datoteke mape prostora (Potpoglavlje 4.1.1.), te joj se može pristupati po potrebi.

ROS čvor Crazyflie Object igra ključnu ulogu u integraciji različitih senzorskih podataka i detekcije objekata. Podaci o poziciji i orientaciji prikupljaju se preko pretplatnika (engl. subscriber) na ROS teme (engl. topic) iz Crazyflie servera te se neprestano ažuriraju čime imamo dobre informacije o položaju i orientaciji letjelice u stvarnom vremenu.

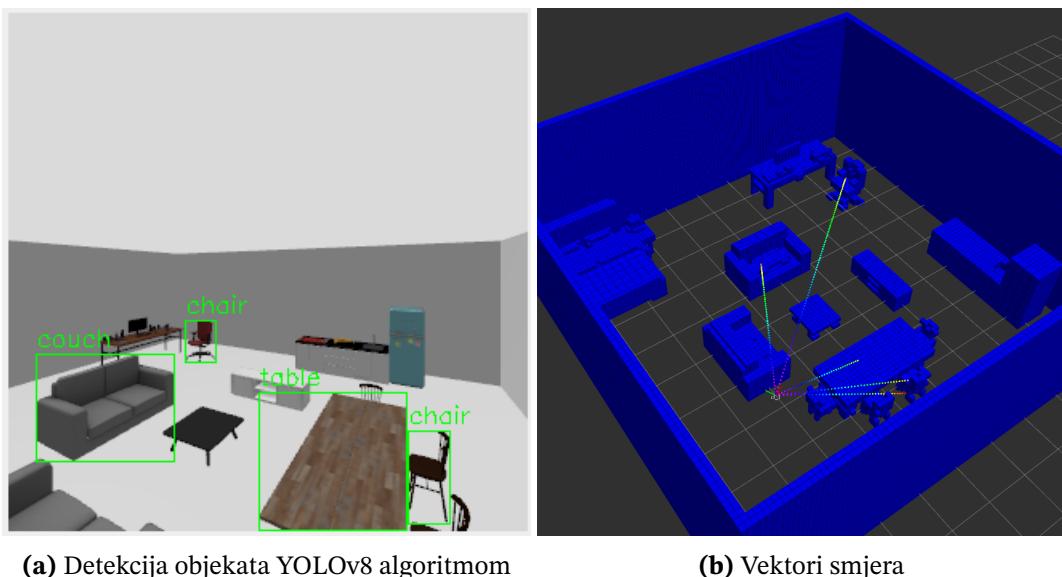


Slika 4.3. Shematski prikaz arhitekture razvijenog sustava

4.2.1. Obrada slike i procjena pozicije objekata

Kamera na letjelici prikuplja slike koje se zatim obrađuju pomoću YOLOv8 neuronske mreže, čime se prepoznaju objekti i njihove koordinate u pikselima. Rezultati modela za svaku sliku pakiraju se u prilagođene ROS poruke naziva *Bounding Boxes* koje sadrže listu svih detektiranih objekata, klase i centre svakog njihovog graničnog okvira. Pomoću intrinzičnih podataka kamere, centre objekata moguće je iz piksela projicirati u jedinični trodimenzionalni vektor smjera u koordinatnom sustavu kamere. Zatim se, koristeći orientaciju letjelice, mogu izračunati rotacijske matrice transformacije iz koordinatnog sustava letjelice u globalni koordinatni sustav. Jedinični vektor smjera tada

je moguće predati kao argument u funkciju 'CastRay' iz OctoMap biblioteke. Funkcija CastRay koristi se za ispitivanje presjeka između virtualne zrake i objekata u prostoru, pri čemu je smjer zrake definiran izračunatim jediničnim vektorom za svaki prepoznati objekt, a izvorište je sama pozicija letjelice u globalnom sustavu. Slika 4.4. prikazuje primjer detekcije objekata na slici iz simulacije te vizualizaciju vektora smjera prema tim objektima. Funkcija vraća koordinate prve zauzete ćelije na svome putu, a te koordinate možemo pridijeliti pojedinom objektu kao njegov položaj u prostoru. Ovi podaci spremaju se u Python rječnik (engl. dictionary) odakle im se može pristupati iz glavnog dijela sustava (Main) koji upravlja logikom pretrage i kontrolom bespilotne letjelice.

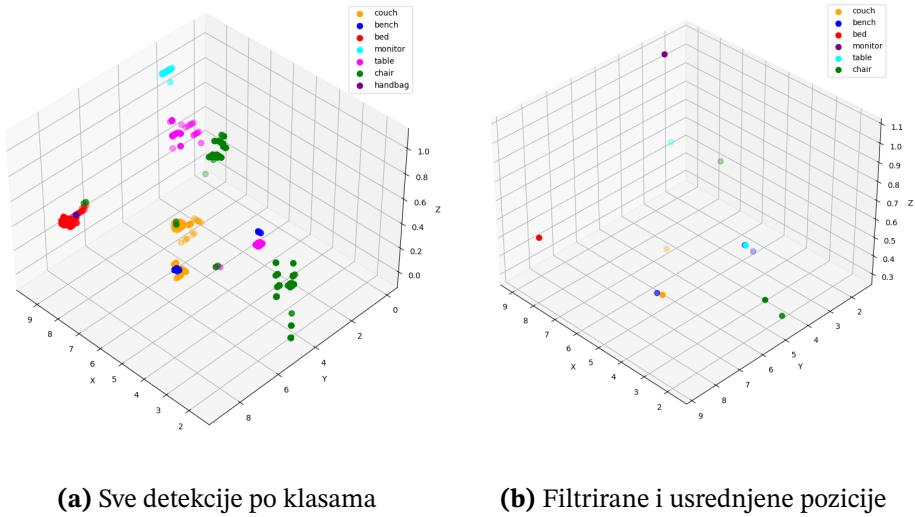


Slika 4.4. Primjer detekcije objekata i računanja vektora smjera prema istima

Kako se letjelica kreće kroz prostor, kontinuirano mijenja kut pod kojim promatra svaki detektirani objekt. Budući da se podaci prikupljaju sa svakom slikom iz kamere, pri brzini od 20 slika u sekundi generira se jednako toliko očitanja, što rezultira velikom količinom redundantnih podataka. Osim toga, prisutnost netočnih očitanja dodatno otežava obradu i analizu, zahtijevajući filtriranje i optimizaciju podataka kako bi se osigurala točnost detekcije.

Kako bi se stvarna pozicija pojedinog predmeta točnije procijenila, polazi se od pretpostavke da će sva očitanja biti raspršena oko stvarne pozicije objekta. U tu svrhu koristi se DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algoritam za

grupiranje detektiranih točaka u prostoru. DBSCAN omogućuje otkrivanje gusto povezanih skupova podataka na temelju prostorne blizine. Za svaku klasu objekata, analizira skup točaka i formira skupine koje predstavljaju grupirana pojavljivanja istog objekta u prostoru. Ključni hiperparametri uključuju maksimalnu udaljenost između točaka unutar iste skupine i minimalan broj točaka potreban za formiranje skupine, te su priлагodeni specifičnostima okruženja i osjetljivosti senzora. Nakon grupiranja, određuje se prosječna vrijednost koordinata svake skupine točaka, čime se dobiva središnja točka objekta, eliminirajući pritom pojedinačne lažne detekcije i šum u podacima.



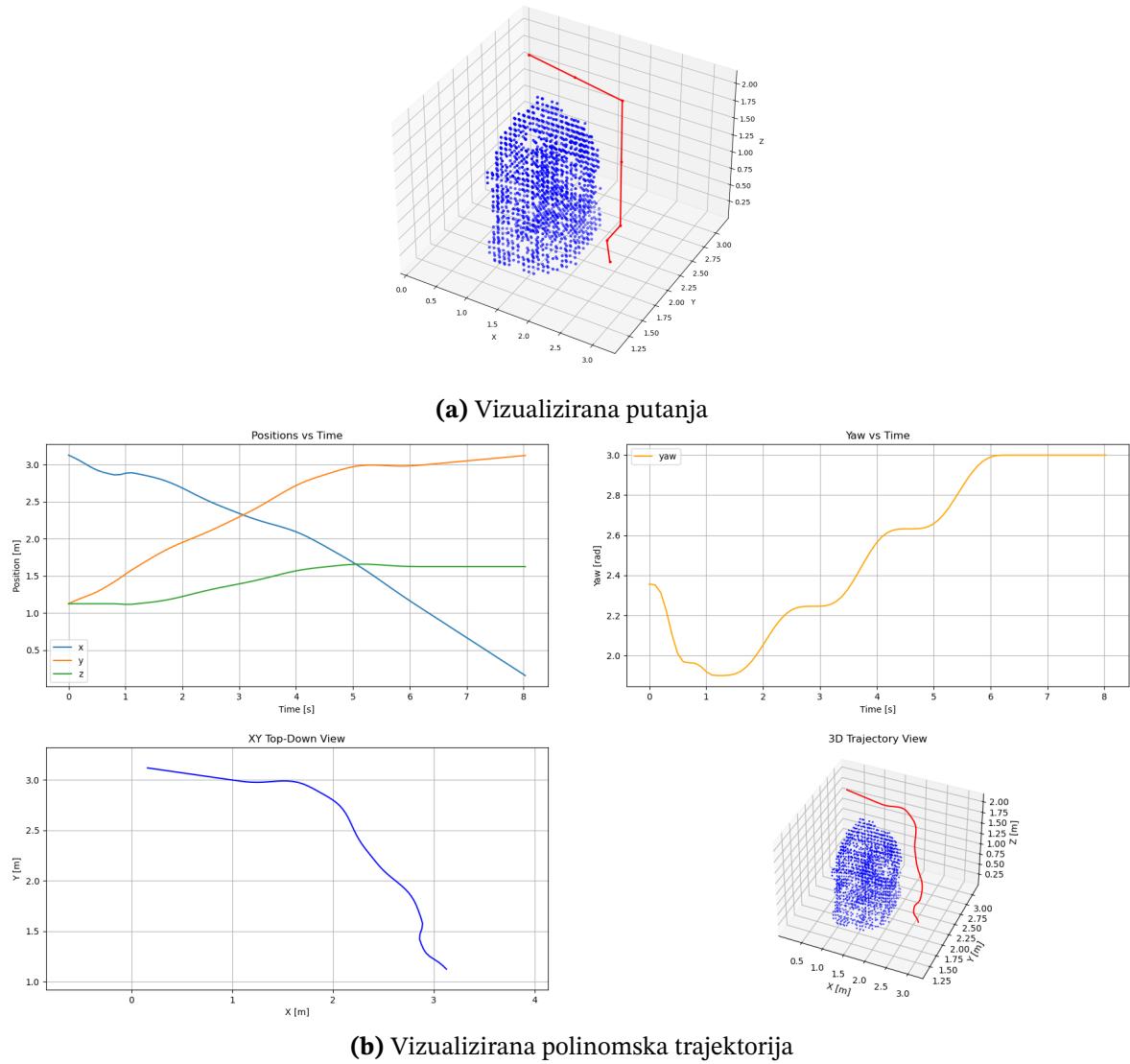
Slika 4.5. Primjer usrednjavanja detektiranih pozicija objekata

4.2.2. Interpretacija semantike prostora

Veliki jezični model (LLM) analizira dobivene podatke i pomaže u određivanju logičkog smjera kretanja letjelice. Jezičnim modelom obraduju se i početne korisnikove informacije o predmetu koji se traži s dodatnim uputama o potrazi (ako ih ima). Iz popisa objekata i njihovih koordinata, model može zaključiti odnose među njima i pridati im temeljitije objašnjenje od samo klase koju YOLO mreža prepoznaće. Tako može razlikovati blagovaonski stol od radnog stola jer se na radnom stolu nalaze računalo, tipkovnica i kraj njega jedna stolica, dok je oko blagovaonskog stola pozicionirano više stolica. Takva značenja može proširiti i na višu razinu pa povezujući predmete može razlikovati dnevnu sobu, radnu sobu, kuhinju i spavaću sobu. Shvaćanje prostora u cijelosti na višim i nižim razinama omogućuje precizniju i točniju navigaciju pri potrazi raznovrsnih predmeta. Također, jezični model koristi se i za generaciju tekstualnog opisa prostora iz prikupljenih podataka uz popratni tlocrt prostora.

4.2.3. Načini skeniranja prostora

Modul Search methods definira različite metode pretrage prostora: inicijalno skeniranje, praćenje zida, kružna putanja oko središta prostora (Slika 4.8.b) ili temeljita pretraga oko nekog predmeta, koje će biti detaljnije opisane u nastavku. Unutar svake metode provodi se računanje putanja koje se trebaju izvesti, izbjegavajući prepreke, te generiranje polinomskih trajektorija koje letjelica zatim slijedi. Prilikom izračuna uzima se u obzir trenutna orientacija letjelice i smjer gledanja kamere tijekom leta čime se izbjegavaju nagli okreti koji mogu dovesti do nestabilnosti. Na slici 4.6.a prikazan je primjer generiranja putanje A* algoritmom iz točke (3.0, 1.0, 1.0) u točku (0.0, 3.0, 1.7) dok je na slici 4.6.b prikazana generirana trajektorija koja prati točke putanje. Također, dani su i grafovi promjene pozicije i orientacije letjelice (engl. yaw) u vremenu.



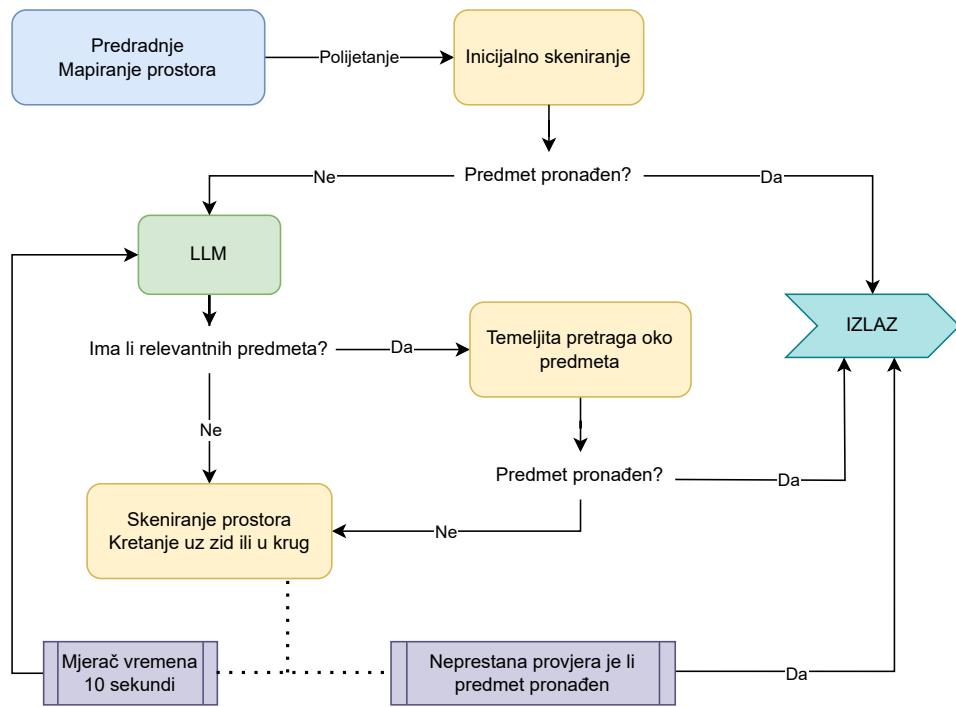
Slika 4.6. Primjer putanje i trajektorije uz izbjegavanje prepreke

Na temelju informacija iz YOLO detekcije, OctoMap mapiranja i LLM odluka, sustav formira preciznu strategiju pretrage i prilagođava navigaciju u stvarnom vremenu. Ova arhitektura omogućuje inteligentno pretraživanje prostora, prepoznavanje objekata i sigurno kretanje bespilotne letjelice kroz prostor.

4.3. Strategija pretraživanja prostora

4.3.1. Pregled strategije

Na slici 4.7. prikazana je osmišljena strategija skeniranja prostora u potrazi za određenim objektom. Prikazana strategija pretraživanja prostora temelji se na sekvencijalnom pristupu koji uključuje mapiranje, inicijalno skeniranje, intelligentnu analizu trenutno dostupnih podataka i detaljnu pretragu oko ciljanih objekata.

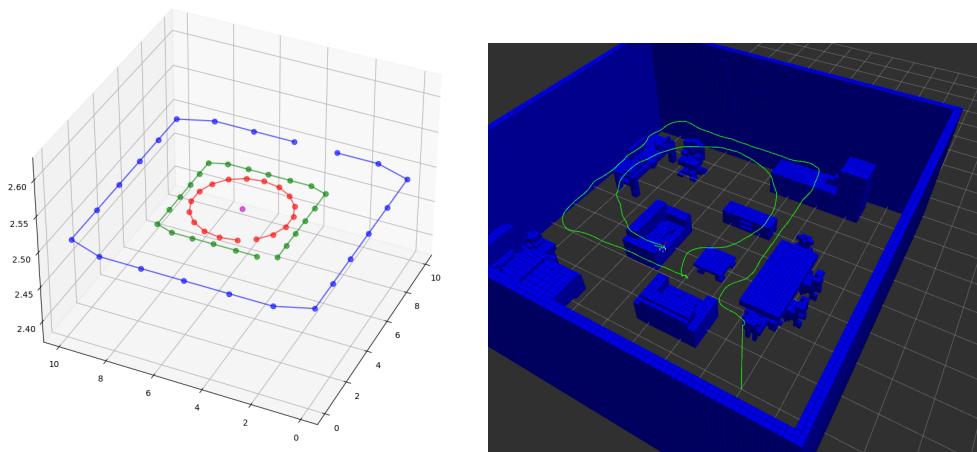


Slika 4.7. Prikaz strategije pretraživanja prostora

Proces započinje predradnjom mapiranja prostora gdje se gradi oktomapa koristeći senzorska očitanja LiDAR-a kako je opisano u potpoglavlju 4.1.1. Trodimenzionalna reprezentacija okruženja bitna je za sigurno kretanje letjelice u svim smjerovima, a iz tih podataka čitaju se i pozicije detektiranih predmeta. Nakon mapiranja, letjelica polijeće i izvodi inicijalno skeniranje oko svoje vertikalne (z) osi, analizirajući prostor kako bi identificirala potencijalne relevantne objekte. Ovdje se uzima u obzir i situacija gdje je predmet odmah vidljiv i dostupan te se potraga završava.

Ukoliko predmet nije opažen inicijalnim pregledom prostora, popis opaženih predmeta i njihovih koordinata šalje se na obradu jezičnom modelu, u ovom slučaju ChatGPT-u. Ako LLM, od opaženih objekata, ne predloži jedan relevantan traženom (po semantici ili uputama korisnika), letjelica prelazi u fazu sistematskog skeniranja prostora, gdje se kreće uz zid ili kružnom putanjom kako bi pokrila što veći dio okruženja. Izračun ovih putanja neovisan je o veličini ili izgledu prostora, te se automatski generira uzimajući podatke iz oktomape prostora. Primjer generiranja obiju putanja i izvršenje istih trajektorija u simuliranom okruženju prikazan je na slici 4.8. Plave točke predstavljaju točke zida, zelene su putanja praćenja zida uz sigurnosni razmak, dok su crvene točke kružna putanja oko središta prostora označenog ljubičastom bojom.

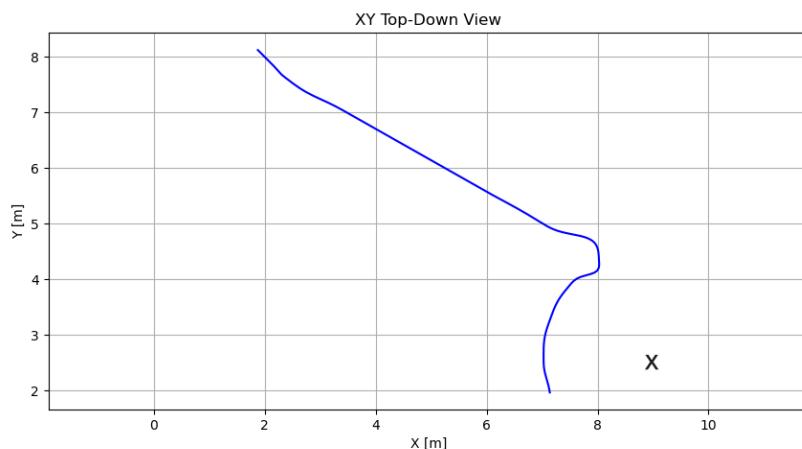
Proces uključuje mjerač vremena koji osigurava da pretraga traje određeni period prije ponovnog vrednovanja situacije i opaženih objekata. Ukoliko se u nekom trenutku detektira traženi objekt, taj događaj ima najveći prioritet i potraga se završava, a kao izlaz algoritma ispisuju se njegove koordinate.



Slika 4.8. Generiranje putanje i izvršenje trajektorije kružnice i praćenja zida

Iz opaženih objekata, ukoliko procijeni dovoljnu relevantnost, jezični model može izabrati predmet koji ima veliku vjerojatnost sadržavanja traženog objekta u svojoj blizini. Primjerice, ako je traženi objekt 'tipkovnica', a opaženi su monitor, stol i jedna stolica svi prostorno blizu jedni drugome, zaključuje se da je to mjesto vrlo vjerojatno prostor za rad na računalu pa vjerojatnost pronalaska tipkovnice raste. U tom slučaju jezični model, kao relevantan objekt, vraća objekt 'monitor'.

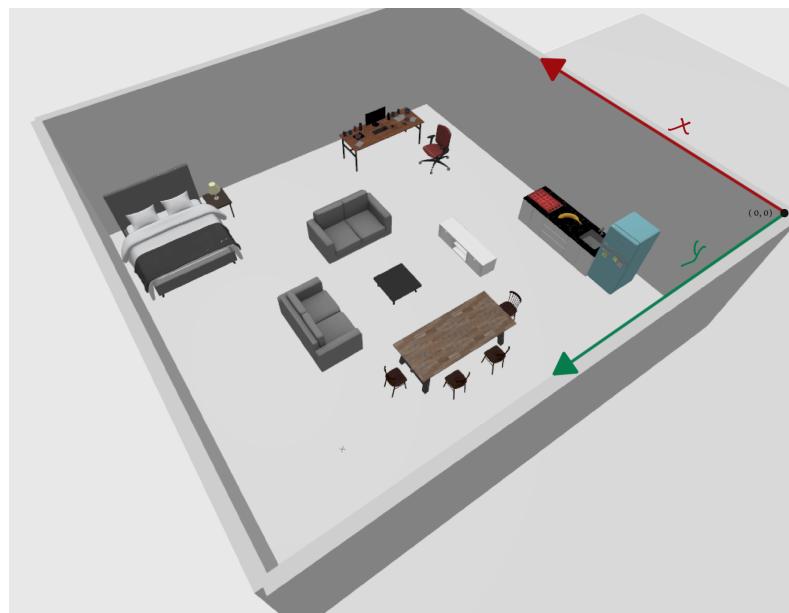
Nakon zaključka o relevantnosti objekta, na redu je temeljita pretraga u toj blizini. Računa se kružnica određenog radiusa u trodimenzionalnom prostoru sa središtem u (x, y) koordinatama relevantnog objekta i na visini definiranoj kao koordinata objekta z uvećana za neku vrijednost. Uzimaju se u obzir i granice prostora te se brišu točke kružnice koje prelaze van prostora ili su nevaljane jer je na tome mjestu neka prepreka. Također se uzima u obzir i sigurnosna udaljenost od prepreka i zida. Na kraju, dobija se putanja i dalje trajektorija koja okružuje relevantan objekt sa smjerom gledanja kamere u smjeru objekta. Time je omogućena temeljita pretraga relevantnog prostora te pregled iz svih mogućih smjerova. Ako se ciljni objekt nalazi daleko od zidova i prepreka, bespilotna letjelica opisat će cijelu kružnicu oko njega. Ako je objekt negdje uza zid, letjelica će letjeti samo dio kružnice koji je valjan, odnosno izbjegava zidove i prepreke. Slika 4.9. prikazuje 2D prikaz odozgo trajektorije kojoj je cilj temeljito pretražiti relevantan objekt detektiran na lokaciji $(9.0, 2.5, 1.0)$. Trajektorija sadrži i kretanje letjelice od njene trenutne pozicije do prve točke opisane kružnice. Vidljivo je u ovome primjeru kako se poštuje ograničenje zidova na pravcima $y = 0$ i $x = 10$ te sigurnosna udaljenost dva metra od njih.



Slika 4.9. 2D trajektorija temeljite pretrage objekta X na koordinatama $(9.0, 2.5, 1.0)$

5. Simulacija

U svrhu testiranja algoritma pretraživanja predstavljenog u ovom radu, u Gazebo Garden simulaciji modeliran je jednostavan prostor 10x10 metara omeđen zidovima, vidljiv na slici 5.1. Prostor sadrži svakodnevne predmete koje YOLO mreža može detektirati kao što su krevet, kauč, stol, stolice, hladnjak, pećnica, monitor itd. Mnogi od tih predmeta veći su i dobro vidljivi, ali pretraga se može usmjeriti na sitnije predmete teško vidljive iz daljine kao što su hrana, tipkovnica i računalni miš. Prostor je raznovrsan i smislen te čini dobru osnovu za testiranje razvijenog algoritma. Cijela prostorija nalazi se u prvom kvadrantu koordinatnog sustava gdje su osi i ishodište položeni kao na slici 5.1.



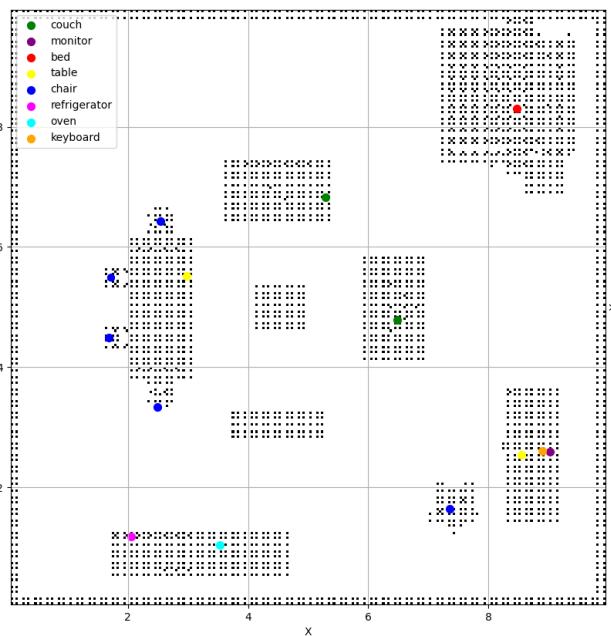
Slika 5.1. Model prostora za testiranje u simulaciji

Kao primjer testiranja razvijenog algoritma pretraživanja, u sljedećim poglavljima opisani su scenariji s različitim traženim predmetima te korisničkim uputama.

5.1. Generiranje tlocrta i tekstualnog opisa prostora

Jedna od mogućnosti ovog sustava nije vezana uz potragu za ciljanim objektom već za skeniranje cijelokupnog prostora. Ako korisniku nije primarna lokacija pojedinačnog objekta, već ga zanima cijelokupna prostorna organizacija uključujući raspored svih objekata i prostorija, može izostaviti eksplisitne upute za pretragu. U tom slučaju, letjelica će autonomno pratiti zidove i kružnim kretanjem kroz prostor dokumentirati dostupne informacije, bilježeći strukturu okruženja i položaj elemenata unutar njega.

Nakon dovršenog skeniranja, iz podataka oktomape generira se tlocrt prostora obogaćen semantičkim oznakama detektiranih objekata, omogućujući vizualizaciju i analizu prepoznatih elemenata u prostoru, ali i validaciju točnosti prepoznavanja pozicija objekata. Rezultirajuća mapa, prikazana na slici 5.2., pruža detaljan uvid u strukturu okruženja i raspored objekata. Uz obrise predmeta, točkama različitih boja predstavljene su grupirane i usrednjene pozicije i klase pojedinih predmeta. Efikasnost pretraživanja proporcionalno raste s količinom dostupnih informacija o prostornim odnosima objekata.



Slika 5.2. Tlocrt prostora sa semantičkim oznakama detektiranih objekata

Jezični model povezuje klase objekata i njihov prostorni razmještaj kako bi generirao tekstualni opis prostora. Uz tlocrt na slici 5.2. generiran je i sljedeći opis:

The space appears to be a multi-functional room with distinct areas. In one section, a

couch is positioned diagonally with a nearby table, suggesting a living or lounging area. A monitor, keyboard and another table are located close together, indicating a workspace or entertainment zone. A bed is situated in a corner, creating a sleeping area. The presence of multiple chairs scattered throughout suggests flexibility in seating arrangements. An oven and refrigerator are positioned near each other, likely forming a compact kitchen area.

5.2. Scenarij I



Slika 5.3. Objekt pretraživanja - tipkovnica označena zelenom i letjelica crvenom bojom

5.2.1. Korisničke upute

U ovom scenariju provodi se pretraga prostora s ciljem lociranja tipkovnice. Korisnik unosi tekstualni opis traženog objekta te, prema potrebi, može dodati dodatne informacije koje smatra relevantnim za poboljšanje učinkovitosti pretrage. U ovome slučaju korisničke upute dane su bez dodatnih uputa za pretraživanje te glase:

"Please find my keyboard."

LLM obrađuje tekst i određuje objekt pretrage kao *tipkovnicu*, odnosno engl. *keyboard*.

5.2.2. Pretraživanje

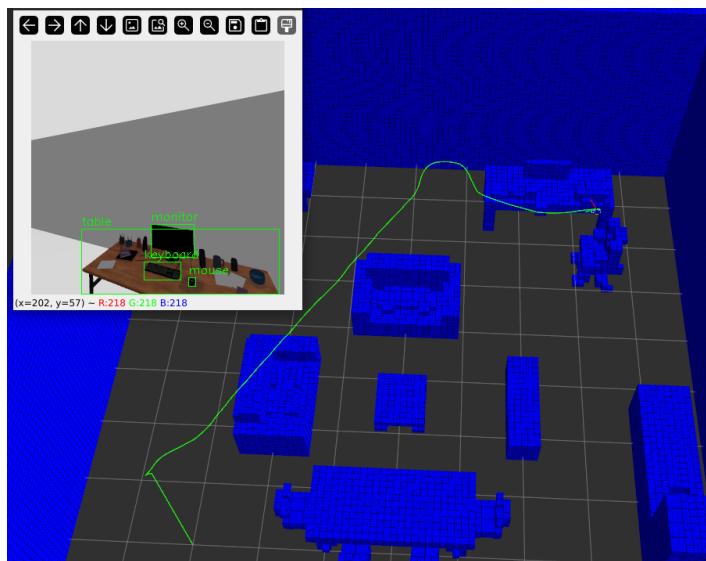
Bespilotna letjelica polijeće s koordinata (2, 8), dijagonalno suprotno od ciljanog objekta (Slika 5.3.) što potragu čini zahtjevnijom. Zatim prati strategiju pretraživanja opisanu u poglavljju 4.3. te započinje početni pregled prostora rotacijom oko vertikalne osi. Po završetku, obrađuje prikupljene podatke o objektima u prostoru, grupira ih, filtrira odstupajuće podatke, usrednjuje njihove koordinate i šalje ih u jezični model.

Među prepoznatim važećim objektima su: kauč na dvije lokacije, klupa, monitor, krevet, stol na dvije lokacije i stolica na tri. Uzimajući u obzir sve predmete i njihove lokacije, jezični model potvrđuje da među njima ima barem jedan relevantan objekt, te kao rezultat vraća monitor na lokaciji (9.03, 2.58, 1.07) uz objašnjenje:

"A keyboard is most likely to be found near a monitor as they are commonly used together in a computer setup"

Stoga letjelica ne kreće skenirati cijeli prostor već leti dijagonalno prema monitoru i kreće temeljitu pretragu njegove okoline, kako je objašnjeno u poglavlju 4.3. Po završetku detaljnog skeniranja područja oko monitora, sustav analizira popis detektiranih objekata. Ukoliko se traženi predmet nalazi među opaženim objektima, pretraga se zaključuje, a koordinate pronađene tipkovnice se evidentiraju i ispisuju:

"Keyboard found at x: 8.8 y: 2.62 z: 0.67"



Slika 5.4. Potraga za tipkovnicom provedena uz primjenu semantičkih metoda

Pretraživanje je provedeno i s računalnim mišem pokraj tipkovnice kao ciljanim objektom, pri čemu su rezultati i logika pretraživanja ostali isti kao u slučaju tipkovnice. Međutim, zbog manjih dimenzija, miš se pokazao kao znatno zahtjevniji objekt za detekciju. Također, isti scenarij testiran je i bez primjene jezičnog modela za ubrzavanje pretrage, a usporedba rezultata oba pristupa prikazana je u poglavlju 5.5.

5.3. Scenarij II



Slika 5.5. Objekt pretraživanja - banana na blagovaonskom stolu

5.3.1. Korisničke upute

U ovom scenariju provodi se pretraga prostora s ciljem lociranja banane. Ona je, sa strane letjelice, djelomično prikrivena vazom s cvijećem koja je dodana sa svrhom otežavanja detekcije iz pojedinih smjerova gledanja. Korisnik unosi tekstualni opis traženog objekta te, prema potrebi, može dodati dodatne informacije koje smatra relevantnima za poboljšanje učinkovitosti pretrage. I ovaj put, korisničke upute dane su bez dodatnih uputa za pretraživanje te glase:

"Where is my banana?"

LLM obrađuje tekst i definira da je objekt pretrage *banana*.

5.3.2. Pretraživanje

Bespilotna letjelica ponovno polijeće s početnih koordinata (2, 8), a ciljni objekt u ovom je slučaju blizu letjelice (Slika 5.5.). Zatim prati strategiju pretraživanja opisanu u poglavljiju 4.3. te započinje početni pregled prostora rotacijom oko vertikalne osi. Po završetku, obrađuje prikupljene podatke, grupira ih, filtrira odstupajuće podatke, usrednjuje njihove koordinate i šalje ih u jezični model.

Među prepoznatim važećim podacima opet su isti predmeti kao u prethodnom scenariju, s obzirom na to da su točka polijetanja i inicijalno skeniranje uvijek isti. To su: kauč na dvije lokacije, klupa, monitor, krevet, stol na dvije lokacije i stolica na tri, te dodana vaza s cvijećem na jednoj lokaciji. Uzimajući u obzir sve predmete i njihove lokacije,

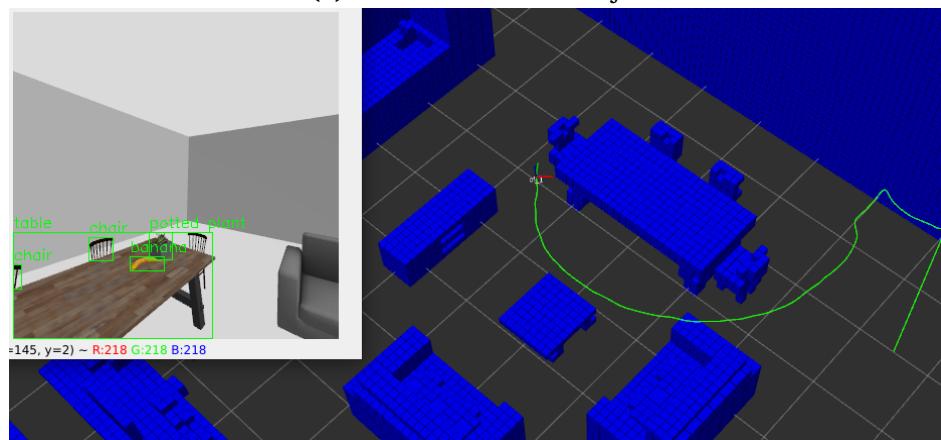
prvo potvrđuje da među njima ima barem jedan relevantan predmet. Zatim kao rezultat vraća stol na lokaciji (2.57, 5.66, 0.71) uz objašnjenje:

"Bananas are often found on tables, especially in dining or kitchen areas. The table at coordinates (2.57, 5.66, 0.71) is near a potted plant and chairs, suggesting a dining or kitchen setting, which increases the likelihood of finding a banana there."

Ovdje je važno skrenuti pažnju na način odabira stola. Naime, LLM je na svoj ulaz dobio dva stola na različitim koordinatama. Zatim je, za svaki od njih, gledao objekte u neposrednoj blizini kako bi im pridodao neko preciznije značenje. S obzirom na to da se u blizini jednog stola nalaze monitor i stolica, a u blizini drugog više stolica i biljka, zaključuje da je drugi stol vjerojatno blagovaonski u prostoru za objed ili kuhinji. Stoga odabire njega kao relevantnog za ciljanu hranu.



(a) Banana skrivena iza biljke



(b) Banana vidljiva iz drugog smjera gledanja

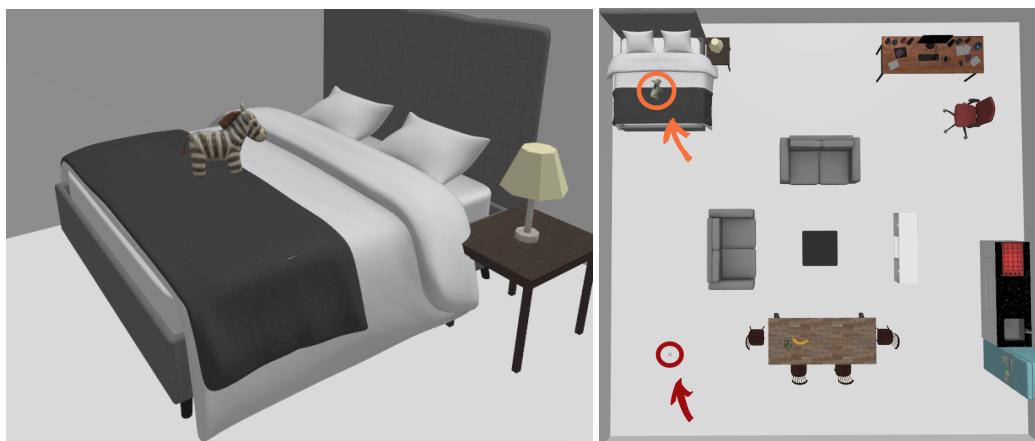
Slika 5.6. Potraga za bananom provedena uz primjenu opisane metode

Letjelica kreće u temeljitu pretragu blizine stola. Po završetku detaljnog skeniranja područja oko stola, sustav analizira popis detektiranih objekata. Ukoliko se traženi predmet nalazi među opaženim objektima, pretraga se zaključuje, a koordinate pronađene banane se evidentiraju i ispisuju:

"Banana found at x: 2.19 y: 5.25 z: 0.72"

Na slici 5.6.a prikazana je situacija iz inicijalnog skeniranja gdje je pogled na bananu zaklonjen vazom s cvijećem. U tom slučaju letjelica ne detektira bananu, ali detektira stol oko kojega detaljnije skenira. Radeći kružnicu oko stola, otvara se pogled na bananu koji nije zaklonjen vazom (Slika 5.6.b) te se potraga završava uslijed uspješne detekcije. Pogled u početnom smjeru je zaklonjen, a taj isti smjer je smjer skeniranja uza zid. Pretraživanje bez uzimanja semantike u obzir zbog toga će trajati duže ili neće niti pronaći ciljni objekt, što je detaljnije opisano rezultatima u poglavljiju 5.5.

5.4. Scenarij III



Slika 5.7. Objekt pretraživanja - igračka zebra na krevetu

5.4.1. Korisničke upute

U ovom scenariju provodi se pretraga prostora s ciljem lociranja plišane igračke u obliku zebre. Ovaj primjer istražuje problem pretraživanja objekta koji može imati više od jednog značenja. Zebra, sama po sebi, teško je semantičko poveziva s objektima u zadanim prostoru, s obzirom na to da se razmatra životinja zebra, stanovnik savana. Stoga je važno, za objekte koji izvorno ne potječu iz trenutnog okruženja, putem korisničkih uputa dati dodatne relevantne informacije o ciljanom objektu. Testirana su oba slučaja:

- Jednostavne upute bez informacija: "*Where is the zebra?*"
- Upute s dodatnim opisom objekta i relevantnim činjenicama: "*I can't find my toy zebra plushie I usually sleep with.*"

LLM obrađuje tekst i definira da je objekt pretrage u oba slučaja *zebra*, ali pri odlučivanju relevantnih objekata koristit će se cjelovite upute korisnika.

5.4.2. Pretraživanje

Bespilotna letjelica ponovno polijeće s početnih koordinata (2, 8), a ciljani objekt u ovom je slučaju lijevo od letjelice (Slika 5.7.). Strategija pretraživanja opisana u poglavlju 4.3 započinje početnim pregledom prostora rotacijom oko vertikalne osi, kao i u prethodnim scenarijima. Po završetku, obrađuje prikupljene podatke, grupira ih, filtrira odstupajuće podatke, usrednjuje njihove koordinate i šalje ih u jezični model.

Među prepoznatim važećim podacima ponovno su isti predmeti kao u prethodnim scenarijima. To su: kauč na dvije lokacije, klupa, monitor, krevet, stol na dvije lokacije, stolica na tri i vaza s cvijećem na jednoj lokaciji. Plišana igračka na krevetu, zbog svoje orijentacije i veličine, nije opažena u inicijalnom skeniranju. Uzimajući u obzir sve predmete i njihove lokacije, razmatra se ima li među njima relevantnih u potrazi za zebrom.

U prvom slučaju s jednostavnim uputama, zaključuje se da relevantnih predmeta nema uz objašnjenje:

"None of the given objects have a strong connection with a zebra, which is typically found in natural environments like savannas or zoos, not indoors near furniture or electronics."

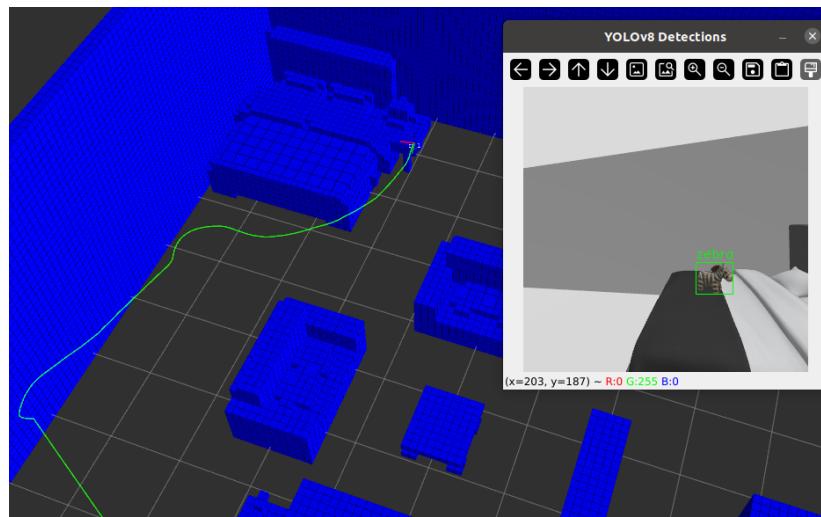
S obzirom na to da nema predmeta vrijednih temeljitije pretrage, letjelica kreće skenirati cijeli prostor prateći zid, isto kao što bi postupila da se ne koristi odlučivanje vođeno semantikom.

U drugom slučaju, LLM koristi dodatne informacije o ciljanom objektu te zaključuje da nije tražena životinja afričkih travnjaka, već plišana igračka s kojom korisnik spava. Jezični model stoga za relevantan objekt izabire krevet na koordinatama (7.5, 8.62, 0.56), uz objašnjenje:

"A toy zebra plushie is most likely to be found near the bed as it is a common place for sleeping and keeping plush toys."

Letjelica kreće u temeljitu pretragu oko kreveta. Po završetku detaljnog skeniranja, sustav analizira popis detektiranih objekata. Ukoliko se traženi predmet nalazi među opaženim objektima, pretraga se zaključuje, a koordinate pronađene plišane zebre se evidentiraju i ispisuju:

"Zebra found at x: 8.48 y: 9.01 z: 0.69"



Slika 5.8. Potraga za zebrom provedena uz primjenu semantičkih metoda

Pretraživanje je također provedeno i bez korištenja semantičke povezanosti, već samo praćenjem zida. Rezultati i usporedbe opisani su u sljedećem poglavljju [5.5.](#)

5.5. Rezultati

Rezultati algoritma dokumentirani su i vrednovani temeljem usporedbe vremena potrebnog za pronalazak predmeta korištenjem semantike u odnosu na neusmjeravano pretrživanje.

U tri prethodno opisana scenarija, prikazane su izvršene trajektorije od polijetanja do pronalaska predmeta koristeći semantičku povezanost i odlučivanje pomoću velikog jezičnog modela. U ovome poglavljju objašnjeni su i komentirani prosječni vremenski periodi potrage u svakom scenariju. Rezultati su uspoređivani s vremenom potrebnim za pronalazak predmeta krećući se samo uza zid u smjeru suprotnom od kazaljke na satu,

prikazano slikom 4.8.b Vrijeme je izraženo u sekundama i predstavlja srednju vrijednost tri pretraživanja za svaki scenarij.

	Scenarij	Bez semantike [s]	Semantika [s]	Ubrzanje [%]
I	Tipkovnica	50.1	48.31	3.4
	Miš	N/A	52.79	∞
II	Banana	79.35	44.38	44.07
III	Zebra	70.21	44.34	36.85

Tablica 5.1. Usporedba vremena pretraživanja sa i bez korištenja semantike

Tablica 5.1. prikazuje prosječne rezultate pretraživanja prostora u potrazi za navedenim objektima. Za svaki ciljani objekt dano je vrijeme u sekundama od polijetanja do njegova pronalaska i evidentiranja njegove pozicije. Vrijeme je mjereno u slučaju kada se nije koristilo semantičko povezivanje objekata koristeći LLM te slučaj u kojem se koristio takav pristup kako bi se potraga ubrzala. Pri pretraživanju bez semantike letjelica leti prateći zid i detektira objekte u svome vidnom polju bez detaljnijeg pretraživanja okoline nekih. U zadnjem stupcu u postotku je predstavljeno ubrzanje razvijene metode koja koristi jezični model u odnosu na pretraživanje koje ga ne koristi.

Analiza

Prema tablici 5.1. jasno je vidljivo da je pretraživanje uz korištenje LLM-a u svakom scenariju rezultiralo bržim pronalaskom predmeta. Korisnost razvijene metode i ušteda vremena uvelike ovise o raznim parametrima.

Prvi faktor koji treba uzeti u obzir je veličina objekata. Što je ciljani predmet manji, to više do izražaja dolazi korisnost velikih jezičnih modela. Primjerice, ako je cilj pronaći monitor, on će biti vidljiv već prilikom inicijalnog skeniranja s drugog kraja sobe, što omogućuje letjelici da odmah evidentira njegovu poziciju i sleti bez potrebe za dodatnim pretraživanjem. U slučaju tipkovnice, iako nije vidljiva tijekom inicijalnog skeniranja, njezina veličina omogućuje da bude uočena s određene udaljenosti prilikom skeniranja uz zid. Zbog toga je ubrzanje pronalaska tipkovnice u prvom scenariju minimalno. Znajljne razlike postaju vidljive tek u slučaju računalnog miša. Zbog svoje male veličine,

miš nije uočen tijekom skeniranja prateći zid. Međutim, detaljnijom pretragom, pri kojoj letjelica analizira područje iz više smjerova i s manje udaljenosti, miš je uspješno detektiran te ovdje ne govorimo o ubrzanju pretrage već o samoj uspješnosti iste.

Drugi ključni faktor je početna pozicija letjelice u odnosu na lokaciju ciljanog objekta. U ovom radu, pretraga prostora bez korištenja LLM-a organizirana je tako da letjelica polijeće, zatim prati zid s njene desne strane, krećući se u smjeru suprotnom od kazaljke na satu. Ovakav način pretraživanja pogoduje objektima smještenima desno od letjelice, budući da ih ona detektira među prvima. Nasuprot tome, objekti koji se nalaze lijevo od početne pozicije bit će uočeni tek pri završetku skeniranja uza zid, kada letjelica dosegne taj dio prostora. U takvim slučajevima, LLM-om vođeno pretraživanje omogućuje značajno ubrzanje detekcije kao u trećem scenariju s plišanom zebrom (Slika 5.7.).

Još jedan važan faktor povezan sa smjerom praćenja zida jest zaklonjenost ciljanih objekata iz tog smjera. U drugom scenariju, primjerice, banana se nalazi iza biljke kada se prostor promatra iz perspektive klasičnog skeniranja, što je prikazano na slici 5.5. Zbog toga banana nije odmah detektirana, unatoč tome što se nalazi u blizini mesta polijetanja letjelice i na njenoj desnoj strani. Detekcija se događa tek u kasnijim fazama trajektorije, kada stol postane vidljiv iz drugog kuta. S druge strane, pretraživanje vođeno LLM-om prepoznaće stol kao relevantan objekt te odmah započinje kružno skeniranje, analizirajući ga iz više smjerova. Time se omogućuje brže uočavanje banane, čime se pretraga ubrzava, a vrijeme pronalaska smanjuje za čak 44%.

Posljednji faktor vrijedan razmatranja jest točnost i opširnost korisničkih uputa. Prilikom odlučivanja o relevantnosti predmeta, LLM se oslanja na cjelovite upute, stoga njihov nedostatak ili netočnost mogu značajno utjecati na njegove odluke. U trećem scenariju, gdje je cilj pronaći plišanu zebru, razmatrana su dva slučaja: u prvom korisnik jednostavno zahtijeva pronalazak zebre, dok u drugom preciznije definira da se radi o plišanoj igrački s kojom spava. Kada upute nisu dovoljno detaljne, LLM ne prepoznaće relevantne objekte jer zebra, bez dodatnog konteksta, ne pripada uobičajenom okruženju. Zbog toga se pretraživanje svodi na klasično praćenje zida, bez usmjerenog pristupa. Nasuprot tome, s preciznijim opisom objekta, letjelica odmah prepoznaće krevet kao logično mjesto pretrage te skraćuje vrijeme pronalaska igračke za gotovo 37%, zahvaljujući bolje definiranim uputama.

6. Eksperiment u laboratorijskom okruženju

Nakon uspješnih rezultata u simulacijama, razvijeni algoritam testiran je na stvarnoj letjelici. Kao što je već spomenuto, riječ je o bespilotnoj letjelici Crazyflie 2.1, opremljenoj AI Deckom koji uključuje monokromatsku kameru. Na vrhu letjelice postavljen je veći reflektirajući marker, što omogućuje lakšu detekciju u OptiTrack sustavu.



Slika 6.1. Korištena letjelica Crazyflie

U prostoru za letenje dimenzija 8x10 metara postavljeno je jednostavno okruženje koje se sastoji od stolice i stola, dok su na stolu smješteni monitor, tipkovnica i računalni miš. Prije samog leta važno je mapirati prostor LiDAR-om te iz tih podataka pomoći Cartographer SLAM-a i OctoMap-a konstruirati oktomapu prostora za letenje. Konstruirana scena i proces skeniranja LiDAR-om vidljivi su na slici 6.2.

Za provođenje eksperimenta, važno je uspostaviti tri načina komunikacije s letjelicom:

- CrazyRadio - slanje upravljačkih naredbi letjelici
- WiFi 2.4GHz veza - primanje slike iz kamere s letjelice
- Ethernet - praćenje pozicije letjelice putem OptiTrack sustava

Antena na AI Deck-u je manjih dimenzija, a zbog ograničene baterije ne posjeduje veliku snagu emitiranja signala. Stoga je preporučljivo isključiti sve veze koje koriste frekven-ciju od 2.4 GHz kako bi se smanjile smetnje, čime se osigurava stabilnost prijenosa slike i sprječava prekidanje signala. U ovom kontekstu, letjelica djeluje kao pristupna točka na koju se uređaji spajaju. Međutim, moguće je postaviti AI Deck tako da sliku šalje na zasebni usmjernik (engl. *router*), čime se povezivanjem na njega može osigurati stabilnija veza. Uz vezu s letjelicom i OptiTrack sustavom, važno je imati i stabilnu internetsku vezu kako bi mogli slati API upite jezičnom modelu.



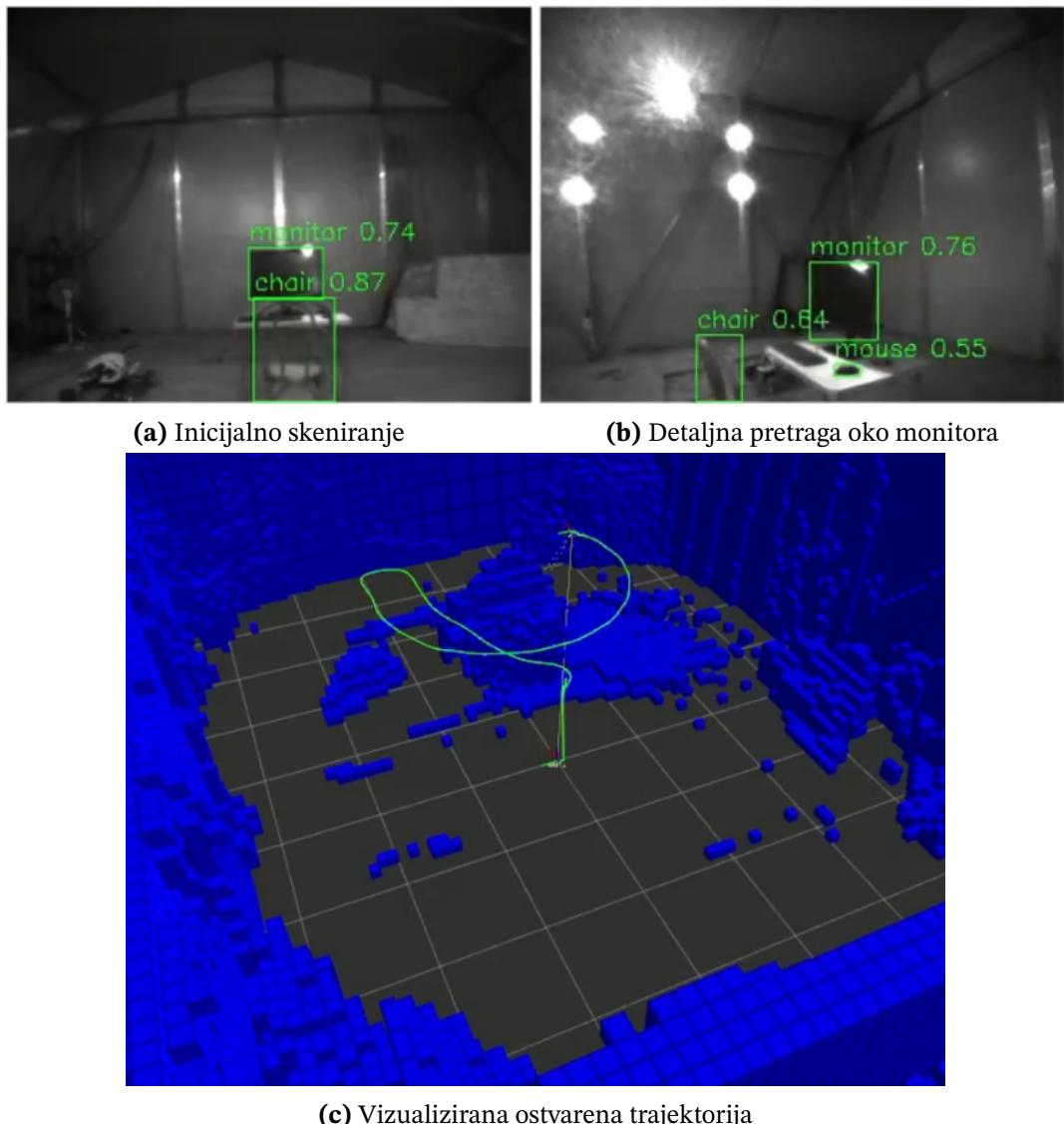
Slika 6.2. Prostor za letenje i njegovo skeniranje

Zadatak letjelice sličan je onome u simulaciji, a korisničke upute glase:

"Where is my computer mouse?",

gdje je važno modelu napomenuti da je riječ o računalnom mišu, a ne glodavcu. U suprotnom, jezični model neće opažene objekte povezivati s računalnim mišem, već će kod odabira relevantnih objekata uzimati u obzir da se traži životinja miš.

Polijetanjem i inicijalnim skeniranjem prostora, letjelica prepoznaje stol, stolicu i monitor, te nakon kraće analize odlučuje provesti temeljitu pretragu u okolini monitora. Tijek pretrage prikazan je na slici 6.3. Slika a) prikazuje pogled iz kamere letjelice neposredno nakon polijetanja. Zbog kuta gledanja, računalni miš nije vidljiv jer je zaklonjen stolicom. Slika b) prikazuje pogled prema monitoru tijekom jednog dijela temeljite pretrage, kada je računalni miš postao vidljiv. Na slici c) prikazana je kružna trajektorija pretrage oko monitora.



Slika 6.3. Proces pretraživanja prostora za letenje

7. Zaključak

Provedeni eksperimenti i testiranja algoritma za pretragu predmeta uz pomoć bespilotnih letjelica pružili su vrijedne uvide u prednosti primjene velikih jezičnih modela (LLM). Rezultati su jasno pokazali da LLM omogućava značajno brže i preciznije lociranje ciljanih objekata u odnosu na tradicionalne metode pretrage, posebno u situacijama kada su objekti manji, zaklonjeni ili smješteni na teško dostupnim mjestima. Osim toga, precizno definirane korisničke upute igraju ključnu ulogu u ubrzanju procesa pretrage, jer LLM bolje interpretira i reagira na jasne smjernice, omogućujući time letjelicu da brže identificira relevantne objekte i poduzme odgovarajuće akcije.

U scenarijima gdje objekti nisu odmah uočljivi zbog smjera pretrage ili zaklonjenosti, korištenje LLM-a omogućuje letjelicu da se usmjereno orientira prema relevantnim područjima, čime se postiže veća efikasnost u pronalaženju predmeta. Također, eksperimentiranjem s različitim vrstama objekata i uvjetima pretrage, utvrđeno je da algoritmi vođeni LLM-om mogu drastično skratiti vrijeme pretrage u kompleksnijim okruženjima.

Nadalje, pokazano je da se algoritam uspješno primjenjuje izvan simulacijskog okruženja, na stvarnoj letjelici, gdje ostvaruje podjednako dobre rezultate. Ovaj uspjeh ukazuje na potencijal algoritma za primjenu u realnim uvjetima, gdje može značajno unaprijediti preciznost i učinkovitost autonomnih sustava u pretrazi prostora.

S obzirom na ove uspjehe, buduća istraživanja trebaju se usmjeriti na daljnju optimizaciju algoritama za pretragu, s naglaskom na primjenu u još dinamičnijim i nepredvidivim okruženjima, te integraciju naprednijih metoda umjetne inteligencije za još učinkovitije donošenje odluka. Korištenje LLM-a može predstavljati ključnu tehnologiju za povećanje autonomije i preciznosti bespilotnih sustava u širokom rasponu, od industrije do robotike i sigurnosnih sustava.

Literatura

- [1] Bitcraze, “Crazyflie 2.1 product page”, pristupano: 10. studenog 2024. [Mrežno]. Adresa: <https://www.bitcraze.io/products/old-products/crazyflie-2-1/>
- [2] K. M. e. a. Wolfgang Hönig, “Crazyswarm2: A ros 2 testbed for aerial robot teams”, pristupano: 13. studenog 2024. [Mrežno]. Adresa: <https://imrclab.github.io/crazyswarm2/>
- [3] Livox, “Mid-360lidar”, pristupano: 13. studenog 2024. [Mrežno]. Adresa: <https://www.livoxtech.com/mid-360>
- [4] W. Hess, D. Kohler, H. Rapp, i D. Andor, “Real-time loop closure in 2d lidar slam”, u *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016., str. 1271–1278.
- [5] Livox, “Cartographer ros”, pristupano: 20. studenog 2024. [Mrežno]. Adresa: <https://google-cartographer-ros.readthedocs.io/en/latest/>
- [6] CloudCompare, “Cloudcompare - 3d point cloud and mesh processing software”, 2025., pristupano: Studeni 2024. [Mrežno]. Adresa: <https://www.danielgm.net/cc/>
- [7] G. Obradović, “Strukture podataka za cad objekte”, diplomski ili magistarski rad, Fakultet elektrotehnike i računarstva, 2008.
- [8] J. Yang, R. Li, Y. Xiao, i Z.-G. Cao, “3d reconstruction from non-uniform point clouds via local hierarchical clustering”, 07 2017., str. 1042038. <https://doi.org/10.1111/12.2281528>
- [9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, i W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees”,

Autonomous Robots, 2013., software available at <https://octomap.github.io>.

<https://doi.org/10.1007/s10514-012-9321-0>

- [10] E. W. Dijkstra, “A note on two problems in connexion with graphs”, *Numerische mathematik*, sv. 1, br. 1, str. 269–271, 1959.
- [11] D. Mellinger i V. Kumar, “Minimum snap trajectory generation and control for quadrotors”, u *2011 IEEE International Conference on Robotics and Automation*, 2011., str. 2520–2525. <https://doi.org/10.1109/ICRA.2011.5980409>
- [12] G. Jocher, A. Chaurasia, i J. Qiu, “Ultralytics yolov8”, 2023. [Mrežno]. Adresa: <https://github.com/ultralytics/ultralytics>
- [13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, i P. Dollár, “Microsoft coco: Common objects in context”, 2015. [Mrežno]. Adresa: <https://arxiv.org/abs/1405.0312>
- [14] OpenAI, “Chatgpt api”, 2024., pristupano: Veljača 2025. [Mrežno]. Adresa: <https://openai.com>
- [15] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Mrežno]. Adresa: <http://www.blender.org>
- [16] P. Min, “binvox”, <http://www.patrickmin.com/binvox> or <https://www.google.com/search?q=binvox>, 2004 - 2019., pristupano: Si-ječanj 2025.

Sažetak

Upravljanje bespilotnom letjelicom za pronađenje objekta u 3D prostoru na temelju semantike prostora

Marin Maletić

U ovome radu razvijen je algoritam koji koristi semantiku okoline u svrhu vođenog pretraživanja prostora. Razvijeni autonomni sustav koristi detekciju objekata pomoću YOLOv8 neuronske mreže i oktalnu reprezentaciju prostora kako bi bespilotna letjelica Crazyflie stvorila semantičku sliku okoline. Na temelju opaženih objekata i njihovih prostornih odnosa, pomoću velikih jezičnih modela stvara se odluka kamo bi se letjelica trebala približiti kako bi se vjerojatnost pronađenja traženog predmeta povećala. Algoritam razmatra semantičku povezanost traženog predmeta i opaženih, ali i korisnikove upute i detaljan opis. Time se omogućuje komunikacija letjelice i korisnika u prirodnom jeziku, a dobivene informacije koriste se u skraćivanju vremena pronađenja predmeta. Metoda je testirana u simulatoru na više kompleksnijih scenarija te na pravoj letjelici.

Ključne riječi: Bespilotna letjelica; YOLO; Autonomno pretraživanje; Semantika prostora; LLM; Crazyflie; 3D navigacija, Računalni vid; LiDAR

Abstract

Control of an Unmanned Aerial Vehicle for Object Detection in 3D Space Based on Spatial Semantics

Marin Maletić

In this work, an algorithm was developed that uses the semantics of the environment for the purpose of guided space research. The developed autonomous system uses object detection using the YOLOv8 neural network and the octal representation of the space to create a semantic image of the environment for the Crazyflie drone. Based on the observed objects and their spatial relationships, using large language models, a decision is made as to where the UAV should approach in order to increase the probability of finding the desired object. The algorithm considers the semantic connection between the requested item and the observed items, as well as the user's instructions and detailed description. This enables communication between the UAV and the user in natural language, and the information obtained is used to shorten the time it takes to find objects. The method was tested in a simulator on more complex scenarios and on a real aircraft.

Keywords: Unmanned aerial vehicle; YOLO; Autonomous search; Space semantics; LLM; Crazyflie; 3D navigation, Computer vision; LiDAR