Tehnička škola Čakovec

ELABORAT ZAVRŠNOG RADA

SUSTAV ZA BEŽIČNO NAVIGIRANJE ROBOTSKIM VOZILOM

Mentor:
Mirko Jambrošić, mag. ing. inf. et comm. techn.

Učenik:
Štefan Granatir, 4.RT

Čakovec, svibanj 2023.

Tehnička škola Čakovec

Prosudbeni odbor za završni rad

Učenik: **Štefan Granatir**

Razred: **4.RT**

Školska godina: **2022./2023.**

Obrazovno područje: **računalstvo**

Zanimanje: **Tehničar za računalstvo**

Naziv zadatka: **SUSTAV ZA BEŽIČNO NAVIGIRANJE ROBOTSKIM VOZILOM**

Opis zadatka: Učenik će predstaviti programsko rješenje kojim će omogućiti robotskom vozilu da navigira kroz 2d labirint

Učenik će se za konzultacije obratiti svojem mentoru.

Zadatak zadan:	Rok predaje pisanog rada:	Predviđen datum obrane:
	17.05.2023.	

Mentor:

Mirko Jambrošić, mag.ing.el.

SADRŽAJ

UVOD

Umjetna inteligencija (UI) predstavlja jedno od najaktualnijih područja razvoja tehnologije u današnjem svijetu. Svojim sposobnostima da obrađuje velike količine podataka, otkriva skrivene uzorke, te pruža precizne i pouzdane rezultate, UI je postala središnja tema istraživanja i primjene u brojnim industrijama. Jedna od primjena UI koja je postala sve važnija u posljednjih nekoliko godina je njezina primjena u prijevozu. UI može se primjeniti u različitim aspektima prijevoza, od upravljanja prometom i poboljšanja sigurnosti vozača, do povećanja efikasnosti prijevoza i smanjenja troškova poslovanja.

U ovom radu biti će obrađena tema navigiranja vozila kroz 2d prostor implementacijom UI. Rad se sastoji od dva dijela, programskog i sklopovskog. Programski dio sastoji se od programa za obradu slike napisanog u Pythonu i programa za upravljanje vozilom napisanog u C++-u. Sklopovski dio sastoji se od raznih elektroničkih komponenti koje vozilu daju mogućnost navigiranja kroz prostor.

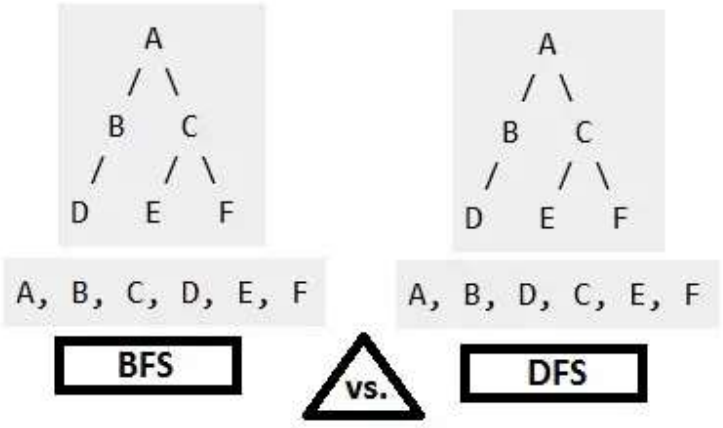
Osobno mi se tema čini zanimljivom. Temu sam odabrao zbog osobne zanimacije prema autonomnim vozilima.

RAZRADA

Programski dio za obradu slike, rješavanje labirinta i komunikaciju s vozilom pisan je u programskom jeziku Python. Kod izrade programskog rješenja korištene su razne biblioteke otvorenog koda, neke od njih su OpenCV, Numpy, PySerial. Programski dio za upravljanje vozilom pisan je u C++ programskom jeziku, i korištene su biblioteke SoftwareSerial i AccelStepper. Fizički dio vozila dizajniran je u Autodesk Fusion 360 programu za 3d modeliranje.

ODABIR ALGORITMA ZA RJEŠAVANJE LABIRINTA

Za rješavanje labirinta korišten je BFS(engl. Breadth first search) algoritam. To je algoritam kojeg odlikuju laka implementacija i srednja razina kompleksnosti. Radi tako da se iz unesene prve točke širi na susjedne sve dok ne dođe do krajnje točke. To ga čini efikasnim za labirinte u kojima su početna i krajnja točka relativno blizu jedna drugoj za razliku od DFS algoritma koji se rekurzivno širi u određenom smjeru. Njeggova jedina mana je prisutna kad su točke na većim udaljenostima jer se broj provjerenih točaka eksponencijalno povećava, što rezultira njegovom malom implementacijom u stvarnom svijetu.



OBRADA SLIKE – UNOS SLIKE

Funkcija cv2.imread() je dio biblioteke OpenCV za obradu slike u programskom jeziku Python. Ova funkcija se koristi za čitanje slike iz datoteke na disku i pretvaranje u format koji se može koristiti za daljnju obradu. Sintaksa funkcije je sljedeća:

```
cv2.imread(putanja_do_slike, flagovi)
```

Ovdje je 'putanja_do_slike' relativna ili apsolutna putanja do datoteke slike koju želimo unjeti, a 'flagovi' su opcionalni argumenti koji se koriste za određivanje načina učitavanja slike. Funkcija se može koristiti u nekoliko formata, uključujući JPEG, PNG, BMP i druge.

Neki od uobičajenih 'flagova' koji se koriste uključuju:

- cv2.IMREAD_COLOR: učitava sliku u boji(3 kanala, BGR)
- cv2.IMREAD_GRAYSCALE: učitava sliku u crno-bijelom kanalu
- cv2.IMREAD_UNCHANGED: učitava sliku u boji s alfa kanalom

Nakon što se slika učita, može se dalje obrađivati koristeći druge funkcije iz OpenCV biblioteke. na primjer, funkcija cv2.imshow() se može koristiti za prikazivanje slike na zaslon, dok funkcija cv2.imwrite() sprema određenu sliku na disk.

Važno je napomenuti da funkcija cv2.imread() može vratiti 'None' ako putanja do datoteke nije ispravna ili se datoteka ne može učitati. Stoga je važno provjeriti povratnu vrijednost funkcije prije nego što se nastavi s daljnjom obradom slike.

OBRADA SLIKE – TRANSLACIJA

Funkcija cv2.goodFeaturesToTrack() u biblioteci OpenCV korištena je za pronalaženje rubnih točaka na slici. Ova funkcija koristi se na razne načine, kao što su praćenje objekata u pokretu, izračunavanje optičkog toka ili izrada panoramskih slika.

Korištenjem ove funkcije mogu se dobiti koordinate rubnih točaka. Funkcija radi tako da analizira intenzitete piksela u blizini svake točke na slici i izračunava kutnu razliku između vektora koja pokazuje smjer promjene intenziteta.

placeholder za sliku primjera

Sintaksa funkcije je sljedeća:

```
rubovi=cv2.goodFeaturesToTrack(slika, Max_rubova, razina_kvalitete,
Min_udaljenost, veličina_blokova, HarrisDetector, k)
```

Ovdje, ‘slika’ predstavlja ulaznu sliku u kojoj se traže rubne točke, ‘Max_rubova’ je maksimalni broj točaka koje će se pronaći, ‘razina_kvalitete’ predstavlja prag kvalitete rubnih točaka, ‘Min_udaljenost’ je minimalna udaljenost između dvije točke, ‘veličina_blokova’ je veličina bloka za izračunavanje gradijenata, ‘HarrisDetector’ se koristi za odlučivanje hoće li se koristiti Harris detektor ruba ili ne, dok se ‘k’ koristi za izračunavanje odgovarajućih pragova za Harris detektor.

Za izračunavanje matrice transformacije perspektive korištena je funkcija cv2.getPerspectiveTransform(), funkcija kao argument prima dva seta koordinata, od kojih prvi set predstavnja koordinate točaka u ulavnoj slici, a drugi set predstavnja koordinate točaka na izlaznoj slici. Na temelju tih koordinata, funkcija izračunava matricu transformacije perspektive koja se dalje koristiti za transformaciju perspektive slike.

Sintaksa funkcije je sljedeća:

```
pts1=np.float32([[RubneTocke[1].x,RubneTocke[1].y],[RubneTocke[0].x,RubneT
ocke[0].y],[RubneTocke[2].x,RubneTocke[2].y],[RubneTocke[3].x,RubneTocke[3
].y]])

pts2=np.float32([[najmanjaX.x, najmanjaX.y-konstUdaljenost], [najvecaX.x,
najvecaX.y-konstUdaljenost], [najmanjaX.x, najmanjaX.y], [najvecaX.x,
najvecaX.y]])

matrix=cv2.getPerspectiveTransform(pts1, pts2)
```

Funkcija cv2.warpPerspective() se koristi za primjenu matrice transformacije perspektive na ulaznu sliku kako bi se dobila transformirana izlazna slika. Ova funkcija prima matricu transformacije perspektive koju smo izračunali korištenjem cv2.getPerspectiveTransform(), kao i dimenzije izlazne slike.

placeholder za sliku primjera

OBRADA SLIKE – PRONALAŽENJE KLJUČNIH TOČAKA

Funkcija cv2.moments() koristi se za izračunavanje različitih karakteristika kontura slike. Funkcija prima binarnu sliku koja sadrži konture i vraća rječnik s karakteristikama kontura. Jedna od karakteristika koju vraća je središte kontrue, koja se ovdje koristi za automatizirano pronalaženje početne i krajnje točke labirinta.

Da bi se pronašlo središte konture, prvo je potrebno pronaći konture slike pomoću funkcije cv2.findContours(), Nakon pronalaska kontura, moguće je primjeniti funkciju cv2.moments() na svaku konturu kako bi se dobila njezina svojstva. u rječniku karakteristika konture, središte konture predstavljeno je kao vrijednost ‘m10’ i ‘m01’, koje su zatim korištene za izračun koordinata središta konture.

```
for i in contours:
    M = cv2.moments(i)
    rect=cv2.minAreaRect(i)
    area=cv2.contourArea(i)
    print(area)
    if M['m00'] != 0:
        cx = int(M['m10']/M['m00'])
        cy = int(M['m01']/M['m00'])
        if(area>500)and(area<1200):
            brojac+=1
            pixelVal = image[cx-8,cy-8]
            print(cx,cy)
            print("boja=",pixelVal )
            cv2.circle(image, (cx, cy), 3, (0, 255, 255), -1)

            cv2.putText(image, str(brojac), (cx - 20, cy - 20)
,cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 2)
```

placeholder za primjer - slika

KOMUNIKACIJA S VOZILOM – PYTHON STRANA

PySerial je Python biblioteka koja se koristi za komunikaciju sa serijskim uređajima poput Arduino mikrokontrolera, senzora i drugih perifernih uređaja. PySerial je popularna biblioteka u području ugrađenih sustava, interneta stvari i drugih sličnih područja. Korištenjem PySerial biblioteke moguće je ostvariti bluetooth komunikaciju između računala i vozila putem serijskog porta.

```
ser = serial.Serial("COM4", 38400, timeout=1)
```

Ova funkcija prihvaća tri argumenta: naziv porta, brzinu prijenosa i opcionalne postavke za serijsku vezu. Nakon otvaranja porta, mogu se koristiti funkcije write() i read() za slanje i primanje podataka preko serijske veze.

```
ser.write(bytes(str(f'poruka'), 'utf_8'))#slanje
input_data=ser.readline() #primanje
```

KOMUNIKACIJA S VOZILOM – C++ STRANA

Za bluetooth komunikaciju između računala i vozila koristi se HC-05 modul. HC-05 je Bluetooth modul koji se često koristi s Arduino platformom za bezžičnu komunikaciju. Međutim, kako Arduino nema ugrađen softverski serijski port, može biti teško uspostaviti vezu između HC-05 modula i Arduina. U tom slučaju, SoftwareSerial biblioteka se često koristi za stvaranje virtualnog softverskog serijskog porta koji se može koristiti za uspostavljanje veze između HC-05 modula i Arduina. Kod komunikacije između Arduino mikroupravljača i modula koristi se UART protokol, što znači da je potrebno spojiti Rx, Tx, VCC i GND pinove za uspješnu komunikaciju.

```
#include <SoftwareSerial.h>
SoftwareSerial serial_con(Rx_pin,Tx_pin);
```

Budući da UART protokol ne koristi sinkronizacijsku liniju potrebno je definirati baud rate (broj bitova po sekundi).

```
serial_con(9600);
```

Kod čitanja sa serijskog porta potrebno je dodatno definirati veličinu buffera, kako ne bi došlo do učitavanje prevelike količine podataka u jednom trenutku. Veličina buffera varira od komponente do komponente, u ovom slučaju, na Arduino Nano mikroupravljaću, ona iznosi 64 bajta.

```
#define BUFFER_SIZE 64
byte byte_count=serial_con.available();
  if(byte_count){
    for(i=0;i<byte_count;i++){
      inChar=serial_con.read();
      indata[i]=inChar;
    }
    indata[i]='\n';
  }
```

IZRADA VOZILA

Fizički dio vozila izrađen je prema načelima tehničke dokumentacije. Zbog korištenja manje količine ulazno-izlaznih pinova i težnje prema kompaknosti cijelokupne konstrukcije, korišten je mikrokontroler Arduino Nano. On upravlja cijelim procesom i izvršava sve naredbe zadane programom. Komunikaciju s računalom omogućuje HC-05 bluetooth modul koji se na mikrokontroler povezuje pomoću UART protokola. Robot se kreće pomoću " NEMA11 koračnih motora koje kontroliraju zasebni A4988 upravljački čipovi. Smješteni su na prednju stranu vozila. Kako bi vozilo bilo stablinije, na zadnju stranu smješten je 360 kotač. Kako A4988 zahtijeva minimalno 8V za napajanje koračnog motora, Za napajanje cijelog vozila koristie se baterija koja se sastoji od 3 serijski povezanih 18650 članaka. Baterija daje 12V napona što je dovoljno za dugotrajnu upotrebu vozila. Konstrukcija vozila u potpunosti je dizajnirana u Autodesk Fusion 360 programu za 3d modeliranje i 3d printana PLA filamentom. Dodatnu stabilnost vozilu daju metalne matice i šarafovi.

ZAKLJUČAK

LITERATURA

POPIS

KONZULTACIJSKI LIST IZRADE ZAVRŠNOG RADA

Ime i prezime učenika: **Štefan Granatir**

Razred: 4.RT

Program-zanimanje: **tehničar za računalstvo**

Mentor: **Mirko Jambrošić, mag.ing.inf. et comm. techn.**

Red. br.	Datum konzultacije	Sadržaj rada	Potpis mentora
1.	10.1.2023	Demonstracija programa za rješavanje labirinta , te dane daljnje upute o izradi završnog rada	
2.	31.1.2023	Pregled napredka, te dane daljnje upute o izradi završnog rada	
3.	3.4.2023	Prikazan fizički prototip vozila. Pregled napredka, te dane daljnje upute o izradi završnog rada	
4.			

Završni rad ocijenjen pozitivno

DA / NE

_____

(potpis mentora)