

# Implementacija stereovizije korišćenjem dve kamere na ugrađenom hardveru

Mihailo Žarković, Dušan Veselinović, Sofija Filipović,  
Goran Ferenc, Dragoje Timotijević  
Syrmia d.o.o.  
Belgrade, Serbia  
[mihailo.zarkovic@syrmia.com](mailto:mihailo.zarkovic@syrmia.com),  
[dušan.veselinovic@syrmia.com](mailto:dušan.veselinovic@syrmia.com),  
[sofija.filipovic@syrmia.com](mailto:sofija.filipovic@syrmia.com), [goran.ferenc@syrmia.com](mailto:goran.ferenc@syrmia.com),  
[dragoje.timotijevic@syrmia.com](mailto:dragoje.timotijevic@syrmia.com)

Danijela Simić  
Faculty of Mathematics, University of Belgrade,  
Belgrade, Serbia,  
[danijela.simic@matf.bg.ac.rs](mailto:danijela.simic@matf.bg.ac.rs)

**Sažetak**— Ovaj rad istražuje implementaciju algoritma za merenje udaljenosti objekata koristeći dve kamere. Fokusira se na teorijsko razumevanje stereoskopskog vida i matematičkih modela, pružajući praktičan vodič za kalibraciju, ekstrakciju tačaka i merenje udaljenosti. Eksperimentalni rezultati pokazuju performanse algoritma u različitim uslovima, doprinoseći unapređenju sistema pomoći vozaču i razvoju autonomnih vozila.

**Ključne riječi**— ključna riječ 1; C++ 2; TensorFlow 3; MobileNet SSD 4; Mbedded hardware 5; Stereovizija 6; OpenCV

## I. UVOD

Saobraćaj je kompleksan sistem kretanja vozila i pešaka na putevima, koji uključuje mnogo faktora kao što su vremenski uslovi, gustina saobraćaja, kvalitet puteva, i mnogi drugi. U poslednjih nekoliko decenija, saobraćajna industrija doživljava značajne promene, a jedna od najvećih inovacija u ovom sektoru su autonomna vozila. Ovakva vozila su napravljena sa ciljem da povećaju efikasnost, bezbednost i održivost saobraćaja, a upotreba senzora na ovim vozilima igra ključnu ulogu. Senzori koji se koriste u autonomnim vozilima se mogu podeliti u tri kategorije: senzori percepcije okoline, senzori kretanja vozila i senzori navigacije.

Senzori percepcije sredine su ključni za autonomna vozila, jer omogućavaju vozilu da prepozna svoju okolinu i proceni situaciju na putu. Ovi senzori uključuju kamere, radare, lidare i ultrazvučne senzore. Senzori kretanja vozila se koriste za merenje brzine, obrtaja, ubrzanja, nagiba i drugih parametara kretanja u vožnji. Senzori navigacije se koriste za održavanje kursa automobila i za navigaciju na putu, a uključuju GPS senzor.

Cilj ovog rada je predavljanje nove aplikacije sistema percepcije okoline, koja može da smanji cenu izrade automobila time što će zameniti radare, lidare i ultrazvučne senzore. Rešenje se oslanja na korišćenje kamere, a sam pristup omogućava korišćenje kamere sa proizvoljnim specifikacijama.

U poglavlju II biće objašnjen algoritam izračunavanja distance i implementacija, u poglavlju III predstavljamo korišćeno razvojno okruženje, dok će u poglavlju IV biti predstavljeni rezultati eksperimenata.

Monovizijski sistemi koriste jednu kameru u vidu senzora kao i algoritme za pozicioniranje objekta unutar slike. Ovo su podaci od značaja koje koristimo za dalje proračune. Da bi se odredila udaljenost objekta pomoću jedne kamere bez dodatnih senzora, potrebno nam je znanje o nekoj referentnoj tački ili dimenzijama nekog objekta unutar slike. Skladno tome, postoje neki vidovi rešenja koji koriste algoritme mašinskog učenja kako bi precizno klasifikovali objekte i time došli do njihovih dimenzija. Ovakav pristup zahteva veliku bazu podataka kao i dosta vremena pri razvijanju algoritama mašinskog učenja, što su mane ovog rešenja. Preklapanje objekata unutar slike dovodi do otežanog merenja kao i do grešaka pri merenju.

### A. Sistem stereovizije

Sistem stereo vizije [1] jeste sistem kompjuterske vizije zasnovan na stereoskopskim tehnikama za izračunavanje udaljenosti objekta. Ovakav sistem koristi dve kamere kao jednu, čime daje utisak dubine slike kao i mogućnost preciznog izračunavanja udaljenosti različitih objekata. Rad ovog sistema prikazan je na dijagramu sa Sl. 1.



Slika 1. Dijagram toka

#### 1) Metod izračunavanja

Stereovizija je tehnika koja koristeći dve kamere izvlači informacije o dubini snimljene scene. Kamere su horizontalno poravnate i vertikalno slobodne u prostoru, slično ljudskim

očima, i koriste se kako bi se dobila dva različita pogleda iste scene. Udaljenost objekta se može izračunati ako se isti nalazi u vidnim poljima kamera.

Na Sl. 2 može se videti kako su kamere horizontalno postavljene na poznatom rastojanju  $A$ , gde  $h$  predstavlja željenu udaljenost koju merimo. Kako bi izračunali udaljenost  $h$  potrebni su nam sledeći parametri:

- $B$  – rastojanje između objekta i leve kamere
- $C$  – rastojanje između objekta i desne kamere
- $\alpha; \varphi; \theta$  – uglovi trougla formiranog od objekta i dveju kamera, kao na Sl. 2.

Prema trigonometrijskim funkcijama i vrednostima označenim na Sl. 2 znamo da važi sledeće:

$$\sin \theta = \frac{h}{c} \quad (1)$$

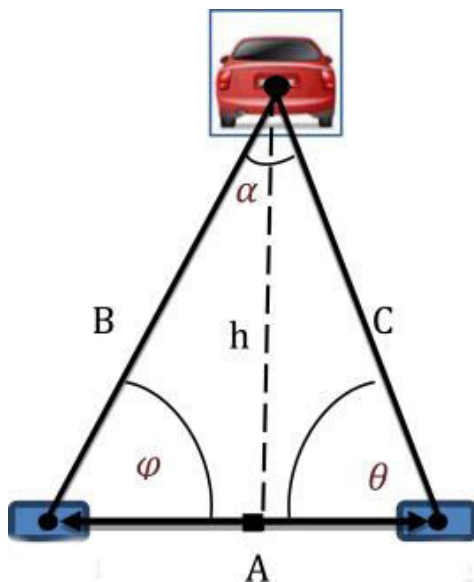
Iz toga možemo izvesti da je:

$$h = B \times \sin \varphi = C \times \sin \theta \quad (3)$$

Prema sinusnoj teoremi važi sledeće:

$$\frac{A}{\sin \alpha} = \frac{B}{\sin \theta} \quad (4)$$

Sledi:



Slika 2 Položaj kamera i objekta čija se udaljenost računa

$$B = \frac{A \times \sin \theta}{\sin \alpha} \quad (5)$$

Sada udaljenost  $h$  možemo predstaviti na sledeći način:

$$h = \frac{A \times \sin \theta \times \sin \varphi}{\sin \alpha} \quad (6)$$

Prema pravilima geometrije znamo da je zbir unutrašnjih uglova trougla sledeći:

$$\alpha + \theta + \varphi = 180^\circ \quad (7)$$

Kada dođemo do uglova  $\theta$  i  $\varphi$  možemo zaključiti koliko iznosi  $\alpha$ . Kao što je označeno na Sl. 3 poznate su nam sledeće vrednosti:

- $\omega_1, \omega_2$  – uglovi vidljivosti leve i desne kamere, respektivno  $\alpha$  (2)
- $H_1, H_2$  - broj horizontalnih piksela leve i desne kamere, respektivno
- $P_1, P_2$  - pozicija objekta u kamerama;  $P_1$  je rastojanje u pikselima između centra objekta i kraja preklapajućeg dela leve kamere;  $P_2$  je rastojanje u pikselima između centra objekta i početka preklapajućeg dela desne kamere.

Sada potrebne uglove možemo izračunati sledećim formulama:

$$\theta = O_2 + \beta_2 \quad (8)$$

Prema Sl. 4 važi:

$$\beta = \frac{180^\circ - \omega}{2} \quad (10)$$

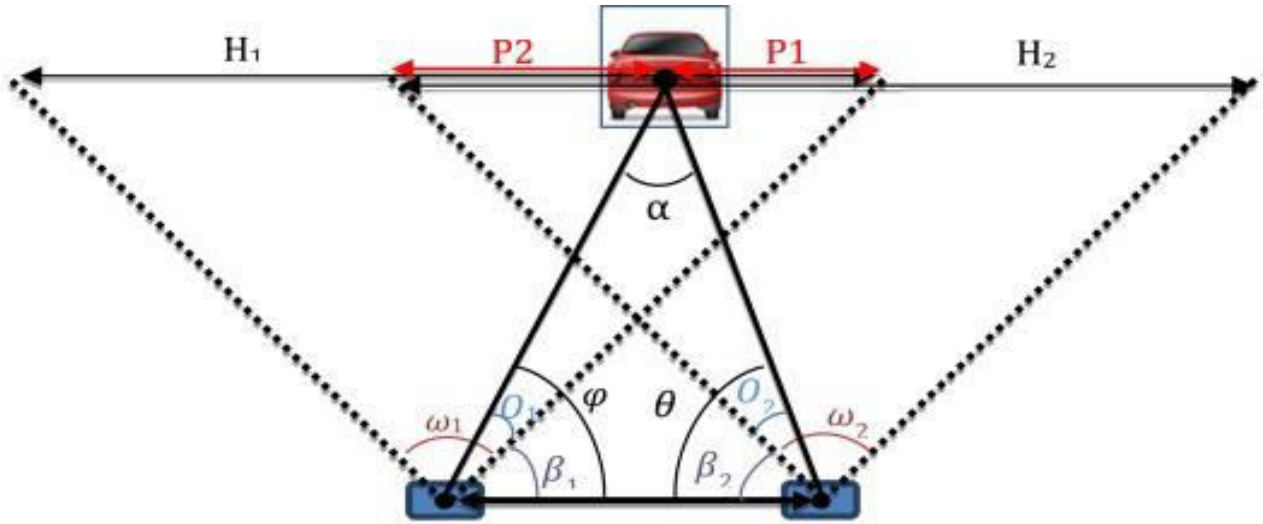
Znajući uglove  $\beta_1$  i  $\beta_2$ , potrebno je odrediti uglove  $O_1$  i  $O_2$  koji predstavljaju proizvod pozicije objekta ( $P_1$  i  $P_2$ ) i ugla jednog piksela kamere, respektivno  $Ap_1$  i  $Ap_2$ .

$$\beta_1 = \frac{180^\circ - \omega_1}{2} \quad (11)$$

$$\beta_2 = \frac{180^\circ - \omega_2}{2} \quad (12)$$

$$O_1 = P_1 \times Ap_1 \quad (13)$$

$$O_2 = P_2 \times Ap_2 \quad (14)$$



Slika 3. Položaj kamere i objekta sa dodatim poznatim vrednostima

Stoga potrebno je izračunati uglove  $Ap_1$  i  $Ap_2$  koji predstavljaju količnik vidnog ugla kamere i horizontalnog broja piksela iste.

$$Ap_1 = \frac{\omega_1}{H_1} \quad (15)$$

$$Ap_2 = \frac{\omega_2}{H_2} \quad (16)$$

Prema prethodno izvedenim formulama možemo predstaviti uglove  $\theta$ ,  $\varphi$  i  $\alpha$  na sledeći način:

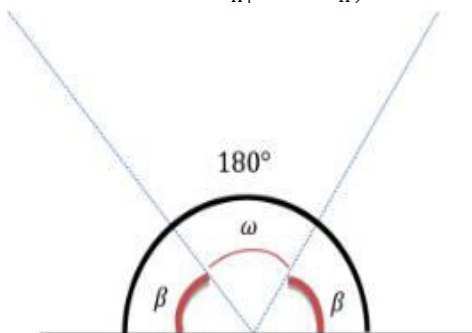
$$\varphi = P_1 \times \frac{\omega_1}{H_1} + \beta_1 \quad (17)$$

$$\theta = P_2 \times \frac{\omega_2}{H_2} + \beta_2 \quad (18)$$

$$\alpha = 180^\circ - (P_1 \times \frac{\omega_1}{H_1} + \beta_1 + P_2 \times \frac{\omega_2}{H_2} + \beta_2) \quad (19)$$

Sada željenu udaljenost možemo izračunati uz formulu:

$$h = \frac{A \times \sin(P_1 \times \frac{\omega_1}{H_1} + \beta_1) \times \sin(P_2 \times \frac{\omega_2}{H_2} + \beta_2)}{\sin(180^\circ - (P_1 \times \frac{\omega_1}{H_1} + \beta_1 + P_2 \times \frac{\omega_2}{H_2} + \beta_2))} \quad (20)$$



Slika 4. Uglovi kamere

## II. PROGRAMSKA IMPLEMENTACIJA

Aplikacija je implementirana u programskom jeziku C++, biblioteka *OpenCV* je korišćena za rad sa kamerama, a *TensorFlow* biblioteka je upotrebljena za upošljavanje *MobileNet SSD* modela sa zadatkom detekcije objekata.

### A. Tehnologije i alati

Korišćenje C++ [2] za razvoj stereovizijske aplikacije na ugrađenom Linux-u donosi visoke performanse, efikasno upravljanje resursima, podršku za objektno-orijentisano programiranje i bogat ekosistem biblioteka poput *OpenCV*. Programski jezik C++ omogućava direktnu kontrolu nad hardverom, lako prenosiv kod između platformi, dobru integraciju sa Linux sistemima.

*OpenCV* [3] pruža obiman set algoritama za obradu slika, kompjutersku viziju i detekciju objekata. Kao open-source biblioteka, podržava objektno-orijentisano programiranje, optimizovane implementacije u C++ za brzu obradu slika, te podržava različite platforme, uključujući Windows, Linux, macOS, Android i iOS. Ima široku primenu u industrijama kao što su medicina, robotika i bezbednost, *OpenCV* je podržan od strane aktivne zajednice korisnika, što olakšava integraciju i pruža resurse za lakše razumevanje i upotrebu. Ova biblioteka je posebno korisna za real-time aplikacije koje zahtevaju brze odzive i efikasnu obradu slika.

Klasa *cv::VideoCapture* kao deo *OpenCV* biblioteke koristi se za pristup video fajlovima, nizovima slika i kamerama, takođe omogućava programeru učitavanje video okvira iz različitih izvora u realnom vremenu.

Klasa *cv::VideoWriter* iz biblioteke *OpenCV* koristi se za snimanje video fajlova na disku, podržava snimanje videa u različitim formatima i kvalitetima.

Prostor imena *cv::dnn* sadrži sve potrebne konstante, funkcije i klase potrebne za rad sa dubokim neuronskim mrežama.

TensorFlow [4] je popularna biblioteka otvorenog koda korišćena u mašinskom učenju, razvijena od strane kompanije Google. Ova biblioteka omogućava korisnicima da kreiraju i treniraju svoje modele dubokog učenja, kao i da primenjuju iste u različitim zadacima, u ovom slučaju u detekciji objekata.

Detekcija objekata je važan zadatak u kompjuterskoj viziji i podrazumeva identifikaciju i lokalizaciju objekta unutar slike i video zapisa. Postoji nekoliko metoda za detekciju objekata, uključujući *YOLO*, *Faster R-CNN* i *SSD*.

*SSD (Single Shot MultiBox Detector)* je jedan od popularnih metoda za detekciju objekata zasnovan na dubokom učenju. Ovaj model je specijalno dizajniran za brzu i preciznu detekciju i koristi konvolucijske neuralne mreže (*CNN*) za ekstrakciju bitnih informacija iz slike. *SSD* modeli koriste mrežu sa više nivoa apstrakcije, što omogućava detekciju objekata različitih veličina i oblika. Ova metoda je efikasna zbog jednog prolaza kroz mrežu (engl. *Single Shot*), što je karakteristično za ovu vrstu modela. Zbog brzih i efikasnih detekcija, *SSD* modeli se koriste u aplikacijama autonomnih vozila, nadzornim aplikacijama i drugim oblastima gde su efikasnost i brzina od značaja [4].

### III. RAZVOJNO OKRUŽENJE

U procesu implementacije, najbitniji hardverski elementi obuhvatali su *Open-Q 865XR Development Kit* i *Open-Q 5165RB SOM* u kombinaciji sa *IMX258* kamerama, čineći ključnu kompjutersku jedinicu za izvršavanje aplikacija. Sa druge strane *EURO RC* automobil [5] sa *Clicker 4* pločom simulira ponašanje stvarnog automobila.

#### A. *Open-Q 865XR Development Kit*

*Open-Q 865XR* [6] razvojni komplet je hardverska i softverska platforma koju je razvio *Intrinsyc Technologies Corporation*, a namenjena je pomoći programerima za razvoj visoko performantnih i naprednih AI (engl. *Artificial Intelligence*) aplikacija. Razvojni komplet uključuje sveobuhvatni set hardverskih komponenti, uključujući razvojnu ploču, kamere, displej modul, kao i razne vrste periferija. Komplet je dizajniran da bude kompatibilan sa širokim spektrom operativnih sistema kao što *Android*, *Windows* i *Linux*. Sveukupno, *Open-Q 865XR* razvojni komplet je snažna i fleksibilna platforma koja omogućava programerima da jednostavno naprave napredne AI aplikacije, što ga čini kao idealan izbor za širok spektar primena, uključujući robotiku, dronove, pametne kamere i mnoge druge

#### B. *Open-Q 5165RB SOM*

*Open-Q 5165RB SOM* (engl. *System on module*) je sistem na modulu dizajniran da pruži visoku performansu, pouzdanost i skalabilnost u ugrađenim aplikacijama, uključujući pametne uređaje, industrijske kontrolere i druge IoT (engl. *Internet of Things*) uređaje. Modul pruža brzu obradu podataka i pouzdanu bežičnu povezanost, ima integrisani *DSP* koji podržava obradu zvuka i slike. Takođe, modul ima interfejsa za kamere, ekran, zvučnike, kao i druge senzore i periferije.

#### C. *IMX258 kamera*

*IMX258* kamera je digitalna kamera koja koristi *CMOS* (engl. *complementary metal-oxide semiconductor*) senzor, ima rezoluciju od 13 megapiksela, što omogućava kvalitetne slike i oštru detaljnost. Kamera podržava različite funkcije, kao što su automatsko fokusiranje, detekcija lica, i druge funkcije koje poboljšavaju kvalitet slike. Prednosti *IMX258* kamere su niska potrošnja energije i velika brzina snimanja, što omogućava korisnicima da snimaju brze pokrete kao i veliki broj fotografija bez prekida.

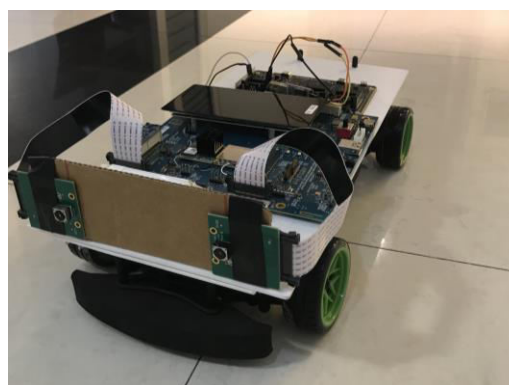
#### D. *EURO RC*

Odlučili smo se za realističnu kopiju Ford Mustanga RTR Spec 5 [7]. Ovaj Mustag je izrađen u razmeri 1/10, opremljen najnovijim HPI Racing Touring Car šasijom. Posедуje stalni pogonom na sva četiri točka, potpuno zatvoren pogonski sklop, vodootporna 2.4 GHz elektroniku i radio opremu. Takođe poseduje regulator brzine koji može biti korišćen sa modernim litijum-polimer (LiPo) baterijama. Upravljanje se vrši servo motorom koji se kontroliše *PWM* singalom. Takođe drugi *PWM* signal se prosledjuje regulatoru brzine koji upravlja *FIREBOLT 15T* motorom.

Kao interfejs za kontrolu automobila iskorišćena je *Clicker 4* ploča od Mikroelektronike. Njena namena je: generisanje *PWM* signala za upravljanje servo motorm, generisanje *PWM* signala za upravljanje regulatorom brzine i pružanje 4 *microBus* interfejsa za različite senzore.

Pomenuta ploča dobija komande putem *UART* komunikacije od *Open-Q 865XR* razvojnog kompleta sa *Open-Q 5165RB SOM*-om, čiji je operativni sistem *Linux*. Na *Linux*-u se izršavaju svi algoritmi za autonomnu vožnju i generišu se upravljačke komande, koje se dalje propagiraju od ploče zadužene za upravljanje automobilom.

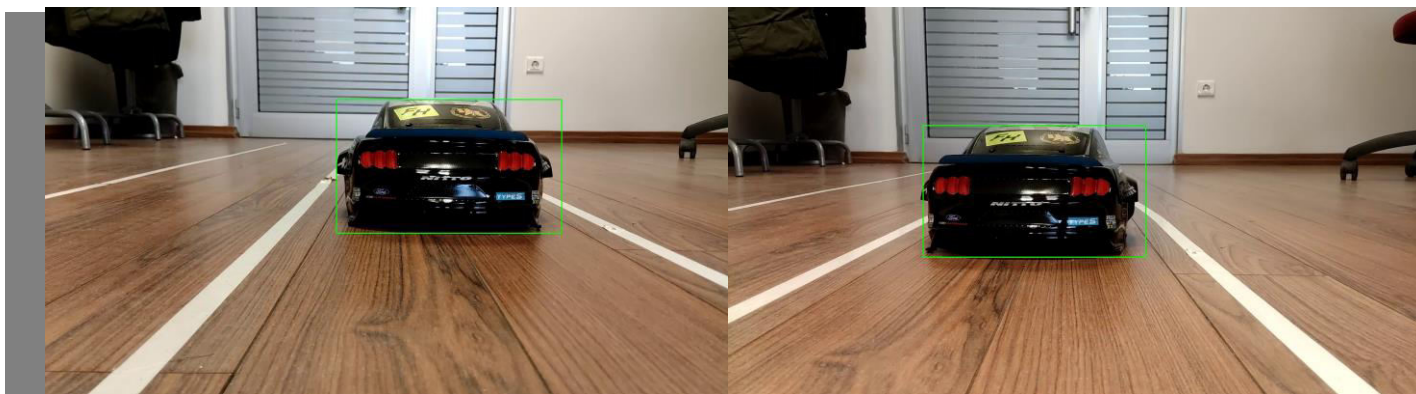
Celokupan izgled automobila sa hardverom dat je na Sl. 5.



Slika 5. Izgled automobila

### IV. REZULTATI

Dobijeni rezultati su postignuti u laboratorijskim uslovima (Sl. 6) i predstavljeni su u tabeli 1. Udeo greške u merenju zavisi od kvaliteta detekcije objekata, kao i od distance između dve kamere koja je pri ovom merenju bila 10 cm. Za



Slika 6. Slika sa leve i desne kamere (respektivno)

postizanje boljih performansi aplikacije potrebno je utvrditi najoptimalniju distancu između kamera i odabrati najoptimalniji model detekcije u zavisnosti od primera upotrebe aplikacije. Planirano je dalje testiranje na otvorenom i mernje udaljenosti objekata na većim distancama.

Tabela 1. Merenja

	<i>Procenjena distanca (m)</i>	<i>Prava distanca (m)</i>
1	0,40	0,42
2	0,91	0,94
3	1,35	1,36

## V. ZAKLJUČAK

U ovom radu predstavljena je nova aplikacija sistema senzora percepcije sredine. Opisan je hardver optimizovan za obradu signala i korišćenje algoritama mašinskog učenja i objašnjen je algoritam izračunavanja distance pomoću stereovizije. Ovaj rad predlaže novo rešenje za računanje udaljenosti objekata, a zbog samog načina rada algoritma dolazimo do zaključka da je moguće znatno smanjiti cenu izrade sistema senzora autonomnih vozila.

Dalje razvijanje aplikacije podrazumeva razvijanje bržih i preciznijih algoritama mašinskog učenja za detekciju objekata, testiranje rada aplikacije u realnom svetu i dodatna optimizacija programskog rešenja kako bi sistem brže radio.

## ZAHVALNICA

Autori zahvaljuju Fondu za inovacionu delatnost Republike Srbije koji je delom omogućio izradu ovog rada u okviru projekta broj 50301 pod nazivom "Platform for Remote development of Autonomous Driving algorithms in realistic environment-READ".

## LITERATURA

- [1] Abdelmoghith Zaarane, Ibtissam Slimani, Wahban Al Okaishi, Issam Atouf, Abdellatif Hamdoun, "Distance measurement system for autonomous vehicles using stereo camera", LTI Lab, Department of Physics, Faculty of Sciences Ben M'Sik, University Hassan II Of Casablanca, Morocco, 2020.
- [2] C++ dokumentacija, internet adresa: <https://en.cppreference.com/w/>, pristupano maj 2023.
- [3] OpenCV dokumentacija, internet adresa: <https://docs.opencv.org/4.7.0/>, pristupano maj 2023.
- [4] TensorFlow internet adresa: <https://www.tensorflow.org>, pristupano maj 2023.
- [5] Mihailo Zarkovic, Goran Ferenc, Stefan Stefanovic, "An autonomous vehicle emulator in the cloud", XXII međunarodni simpozijum INFOTEH-JAHORINA, 2023.
- [6] Lantronix internet adresa <https://www.lantronix.com/products/open-q-865xr-som-development-kit/>, pristupano maj 2023.
- [7] F1TENTH dokumentacija, internet adresa: <https://f1tenth.org/build.html>, pristupano oktobar 2023.

## ABSTRACT

This paper explores the implementation of an algorithm for measuring the distance between objects using two cameras. We focus on the theoretical understanding of stereoscopic vision and mathematical models, providing a practical guide for calibration, point extraction, and distance measurement. Experimental results demonstrate the algorithm's performance under various conditions, contributing to the improvement of driver assistance systems and the development of autonomous vehicles.

## IMPLEMENTATION OF STEREOVISION USING TWO CAMERAS ON EMBEDDED HARDWARE

Author name(s)