



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Наташа Перковић

**Једно рјешење одређивања
удаљености објекта од возила
употребом више камера**

МАСТЕР РАД

Нови Сад, 2019.



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :		
Идентификациони број, ИБР :		
Тип документације, ТД :	Монографска документација	
Тип записа, ТЗ :	Текстуални штампани материјал	
Врста рада, ВР :	Завршни (Мастер) рад	
Аутор, АУ :	Наташа Перковић	
Ментор, МН :	Др. Жељко Лукач, доцент	
Наслов рада, НР :	Једно рјешење одређивања удаљености објекта од возила употребом више камера	
Језик публикације, ЈП :	Српски / латиница	
Језик извода, ЈИ :	Српски	
Земља публикавања, ЗП :	Република Србија	
Уже географско подручје, УГП :	Војводина	
Година, ГО :	2019.	
Издавач, ИЗ :	Ауторски репринт	
Место и адреса, МА :	Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, ФО : (поглавља/страна/ цитата/табела/слика/графика/прилога)	7/34/11/2/18/0/0	
Научна област, НО :	Електротехника и рачунарство	
Научна дисциплина, НД :	Рачунарска техника и рачунарске комуникације	
Предметна одредница/Кључне речи, ПО :	Детекција објеката, стереоскопија, камера, <i>ALPHA</i> , <i>ADAS</i>	
УДК		
Чува се, ЧУ :	У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, ВН :		
Извод, ИЗ :	У оквиру овог рада представљено је рјешење за одређивање удаљености објекта од возила користећи више камера. За реализацију рјешења користили смо <i>ALPHA AMV</i> платформу и <i>Vision Software Development Kit</i> окружење. Концепт рјешења јесте да користећи више камера прецизније измјеримо удаљеност објекта од возила.	
Датум прихватања теме, ДП :		
Датум одбране, ДО :		
Чланови комисије, КО :	Председник: др Драган Пејић, ванр. професор	
	Члан: Др Небојша Пјевалица, ванр. професор	Потпис ментора
	Члан, ментор: Др. Жељко Лукач, доцент	



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic publication
Type of record, TR :	Textual printed material
Contents code, CC :	Master Thesis
Author, AU :	Natasa Perkovic
Mentor, MN :	Zeljko Lukac, PhD
Title, TI :	One solution for determining the distance of the object from the vehicle using multiple cameras
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Vojvodina
Publication year, PY :	2019.
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	7/34/11/2/18/0/0
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering and Computer Communications
Subject/Key words, S/KW :	Object detection, stereoscopic pictures, camera, ALPHA, ADAS
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	This paper presents a solution for determining the distance of the object from the vehicle using multiple cameras. To implement the solution, we used the ALPHA AMV platform and the Vision Software Development Kit environment. The concept of the solution is to use two cameras to measure the distance of the object from the vehicle more precisely.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President: Dragan Pejic, PhD
	Member: Nebojsa Pjevalica, PhD
	Member, Mentor: Zeljko Lukac, PhD
	Menthor's sign

Захвалност

Захвалност дугујем ментору др Жељку Лукачу, на стручним савјетима, помоћи и стрпљењу у току израде овог рада.

Такође, захваљујем се свим својим колегама који су ми пружили, прије свега велику подршку и несебичну помоћ при изради овог рада. Једно велико хвала на стрпљењу, стручним савјетима и пријатељству дугујем колеги Огњену Ђекићу.

Највећу захвалност дугујем својој породици, Немањи и својим пријатељима на огромној подршци, љубави и стрпљењу током читавог школовања.

САДРЖАЈ

1. Увод.....	1
2. Теоријске основе	3
2.1 Напредни системи за помоћ возачу – <i>ADAS</i>	3
2.2 Аутономна вожња	5
2.3 <i>ALPHA</i> платформа.....	6
2.3.1 <i>TDA2x SoC</i>	8
2.4 Развојно окужење <i>VSDK</i>	9
2.5 Генерисање и учитавање алгорита	10
3. Концепт рјешења	14
3.1 Стереоскопија	11
3.2 Детекција објеката – пјешака и возила(<i>Object detection</i>)	14
4. Програмско рјешење.....	16
4.1 Процес писања новог <i>use-case-a</i>	16
4.2 Портовање алгорита за детекцију објеката на <i>ALPHA</i> плочи	16
4.3 Прављење новог <i>use-case-a</i> са двије камере	19
4.3.1 Улазни и излазни подсистеми	22
4.4 <i>Distance</i> блок.....	24
4.4.1 Извлачење података из алгорита за детекцију објеката	24
5. Тестирање	26
6. Закључак	31
7. Литература.....	32

СПИСАК СЛИКА

Слика 1: Приказ сензора за прикупљање података	4
Слика 2: Ток проточне обраде података <i>ADAS</i> система.....	4
Слика 3: <i>ALPHA ADAS</i> платформа	7
Слика 4: Шематски приказ <i>ALPHA</i> платформе.....	8
Слика 5: Шематски приказ <i>Texas Instruments TDA2x SoC-a</i>	9
Слика 6: TI Vision SDK slojevita arhitektura softvera	10
Слика 7: Слика објекта (стабла) снимљена са двије камере	12
Слика 8: Рад алгоритма за детекцију објеката	14
Слика 9: Детектовани аутомобили и раздаљина до њих.....	15
Слика 10: Дијаграм тока <i>Object Detection</i> алгоритма	18
Слика 11: Приказ повезивања камера са <i>SCV SoC</i>	19
Слика 12: Дијаграм тока података <i>Object Detection</i> алгоритма са двије камере.....	21
Слика 13: Улазни подсистем.....	23
Слика 14: Приказ радног окружења	26
Слика 15: Приказ тестног окружења са једном камером	27
Слика 16: Приказ видног поља камере у тестном окружењу	28
Слика 17: Приказ видног поља двије камере у тестном окружењу	29
Слика 18: Графички приказ резултата тестирања са једном и са двије камере	30

СПИСАК ТАБЕЛА

Табела 1: Резултати удаљености објекта од камере, добијени коришћењем једне камере27

Табела 2: Резултати удаљености објекта од камере, добијени коришћењем двије камере29

СКРАЋЕНИЦЕ

ADAS	- <i>Advanced Driver Assistance Systems</i> , Напредни системи за помоћ возачу
AMV	- <i>Central Automotive Machine Vision</i> , Системи за обраду видео сигнала за потребе ауто индустрије
DSP	- <i>Digital Signal Processor</i> , Дигитални сигнал процесор
EVE	- <i>Embedded Vision Engine</i> , Јединица за обраду видео садржаја ниског нивоа
HDMI	- <i>High Definition Multimedia Interface</i> , Мултимедијална веза високе резолуције
IPC	- <i>Inter Process Communication</i> , Међупроцесорска комуникација
PCIe	- <i>Peripheral Component Interconnect Express</i> , Периферна комуникациона брза веза
SBL	- <i>The Secondary Bootloader</i> , Иницијални пуњач другог степена
SDK	- <i>Software Development Kit</i> , Окружење за развој програмске подршке
SoC	- <i>System on Chip</i> , Рачунарски систем у оквиру интегрисаног кола
VSDK	- <i>Vision Software Development Kit</i> , Окружење за развој програма компаније <i>Texas Instruments</i>

1. Увод

Последњих деценија можемо примјетити убрзан развој аутомобила. Од 1970. (140 модела), до 2012. (684 модела), број модела аутомобила расте експоненцијално. Повећањем броја аутомобила повећава се и свијест за сигурном возњом. У последњој деценији, убрзаним развојем аутомобилске индустрије, долази до технолошког напретка возила која су у могућности да све више и вјеродостојније спознају властито окружење. Та способност спознаје властитог окружења, даје одређен ниво аутономије у односу на возача и могућност самосталног учествовања у саобраћају. Важну улогу код оваквих аутомобила добијају напредни системи за помоћ возачу (енг. *ADAS – Advance Driver Assistance System*) [1].

ADAS системи спадају у најбрже растуће области у аутомобилској индустрији. Главне примјене *ADAS-a* су диктиране потребом за изградњом сигурнијих возила, смањењем стреса возача и задовољавањем растућег апетита за нове карактеристике савременог потрошача. Јасно је да је у густо насељеним срединама, а посебно великим градовима, број аутомобила знатно већи. У тим срединама долази до проблема у саобраћају, тако да саобраћај умјесто основне сврхе, лакшег превоза људи, постаје и мјесто губљења људских живота. У циљу спрјечавања саобраћајних незгода, неопходно је додатно унаприједити саобраћај, као и повећати сигурности аутомобила. Будућност друмског саобраћаја се своди на аутоматизацију самог саобраћаја, гдје ће људске грешке тежити нули. Аутомобили ће бити у стању да преузму улогу возача у потпуности. Информације којима располаже аутономно возило морају бити поуздане, што прецизније, тачне и достављене у реалном времену, како би возило могло да одреагује у одређеним ситуацијама на најбољи могући начин. До постизања оваквог циља, наилазимо на многе проблеме. Један од проблема јесте искористити и оптимизовати неопходне ресурсе да би

информације које возило добија са сензора биле обрађене у реалном времену и биле прецизне и тачне.

У овом раду ће бити представљена реализација и примјена алгорита за детекцију објеката и израчунавање дистанце између возила и објекта користећи двије камере постављене са предње стране возила. Објаснићемо и зашто је боље имати више сензора, односно зашто је боље имати двије, а не једну камеру, за коју смо такође извршили рачунање удаљености и поређењем увидјели за колико су бољи резултати када рачунамо удаљеност објеката са двије камере у односу са једном камером.

Рјешење је реализовано у склопу *Texas Instruments*-овог развојног окружења *VisionSDK*, (енг. *Vision Software Development Kit*) [2], верзије 02.12.01.00, у програмском језику *C* и тестирано на *AMV ALPHA* [3] (енг. *AMV, Automotive Machine Vision*) платформи, коју је израдио РТ-РК институт.

Рад је организован у шест поглавља:

- Теоријске основе – опис *ADAS* система и *ALPHA* платформе
- Треће поглавље описује концепт рјешења, од проблема до његовог решења
- У четвртном поглављу дат је опис програмске реализације
- У петом поглављу се налазе резултати овог рада
- Шесто поглавље садржи кратак опис шта је урађено у овом раду и који су правци даљег развоја.

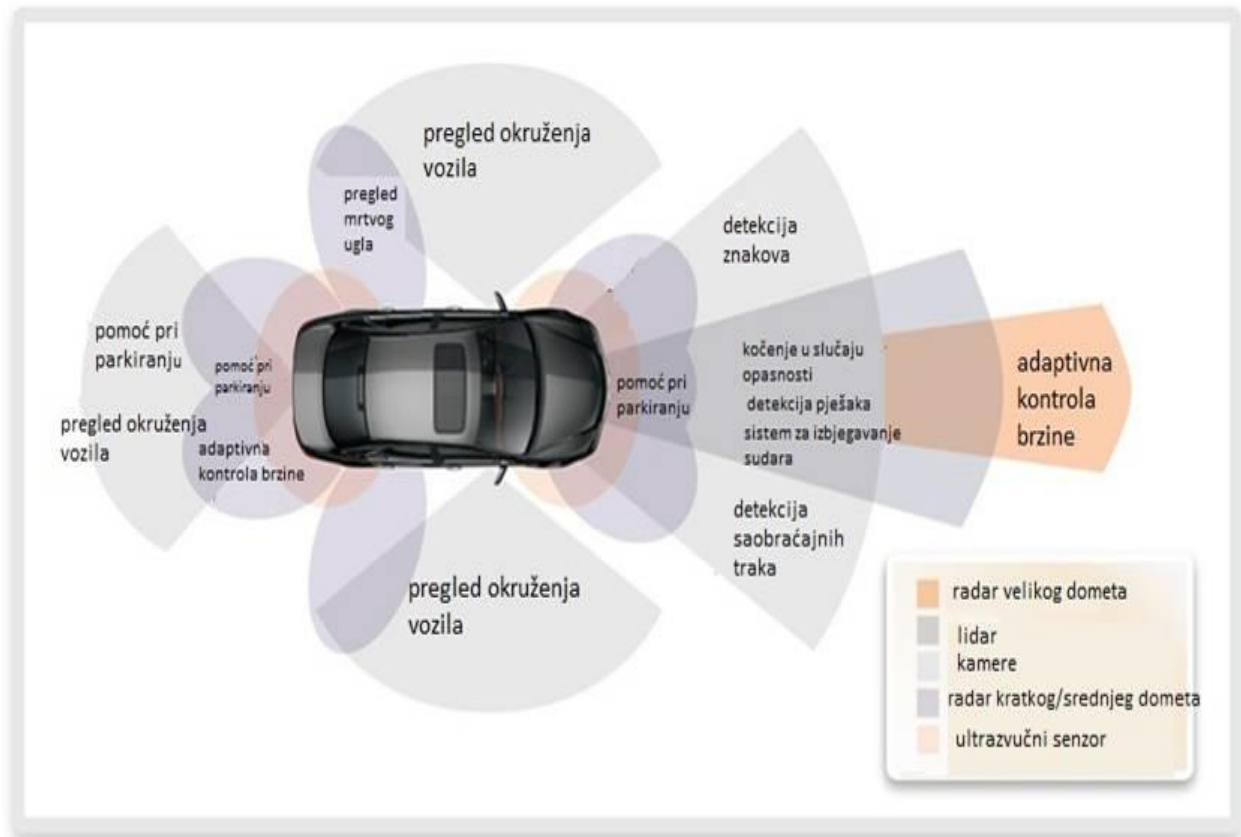
2. Теоријске основе

У овом поглављу су описане теоријске основе на којима је рад заснован. Дат је опис стања у аутомобилској индустрији, приказ напредних система за помоћ возачу *ADAS* система, као и *ADAS* платформе и њених перформанси.

2.1 Напредни системи за помоћ возачу – *ADAS*

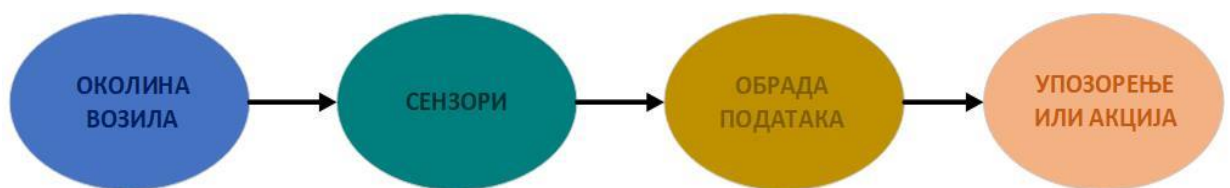
Аутономна вожња представља сет технологија које ће трансформисати аутомобилски свијет у будућност. Људи више неће бринути о вожњи, јер ће тај посао бити препуштен машинама. Мишљења су да ће саобраћај постати "течнији" и сигурнији, јер за разлику од машина, човјек је подложен грешкама, услед умора или недостатка концентрације, као и сваки утицај који може ометати чулно разматрање ситуације током вожње (слаба видљивост, бука....) доводе до разних нежељених последица, па чак и оних фаталних. На основу глобалне статистике, скоро 1.3 милиона људи изгуби живот услед саобраћајне несреће сваке године, а чак 20-50 милиона заврши са озбиљним повредама [4].

Циљ развијања напредних система за помоћ возачу *ADAS*, јесте да вожњу учине сигурнијом, као и комфорнијом. Све више се одговорности препушта рачунарима и смањује се утицај возача. *ADAS* системи су поузданији у односу на природу људи и њихову несавршеност. Ови системи не замјењују у потпуности возача, већ служе као помоћ при управљању машином [5]. Сама функционалност *ADAS* система заснована је на прикупљању података у датом тренутку са уређаја који су постављени са спољашње стране, као и са унутрашње стране аутомобила. Уређаји којима је опремљено возило су разни сензори, нпр камере, радар, лидари, ултразвучни сензори, сензори угла волана, итд. који су задужени за надгледање цјелокупне ситуације током вожње.



Слика 1: Приказ сензора за прикупљање података

По прикупљању података са сензора *ADAS* извршава утврђене кораке њихове обраде - алгоритме. На слици 2. дат је приказ уопштеног тока података у оквиру *ADAS*-а.

Слика 2: Ток проточне обраде података *ADAS* система

Сваки од *ADAS* система користи себи својствене сензоре ради спознавања властитог окружења. Примјери таквих система су [6]:

1. Систем за препознавање саобраћајних знакова (енг. *Traffic Sign Recognition*) који ради на бази обраде података који пристижу самоса камере.
2. Систем за препознавање пјешака (енг. *Pedestrian Detection*) који користе радаре кратког и средњег домета, лидаре и камере.

3. Системи за детекцију судара (енг. *Collision Detection*) користи лидаре, камере и радаре кратког и средњег домета.
4. Систем за аутоматско паркирање (енг. *Automatic Parking*) гдје се поред камере користе и подаци са ултразвучних сензора о растојању околних објеката. Код овог система се користи и сензор угла волана који служи за предикцију трајекторије возила у покрету.
5. Систем за надзор возача (енг. *Driver Monitoring*) користи искључиво камере. Обработом добијених слика, систем спознаје изразе лица возача и адекватно упозорава возача у ситуацијама непажње истог.

Сваки од ових система има неку врсту дефинисаног тока проточне обраде података.

2.2 Аутономна вожња

Постоји скала са нивоима од 0 до 5, коју је усвојио међународно удружење саобраћајних инжињера, *SAE International* [7], које окупља професионалце из света авијације, транспорта и саобраћаја.

Ниво 0 - Без аутономности. Већина возила данас је и даље на овом нивоу. Возач има потпуну контролу над возилом и свим функцијама. Највише што може добити од помоћи су одређена упозорења.

Ниво 1 – "Прва" помоћ возачу. Адаптивни темпомат који контролише брзину и растојање од возила испред су типични пример нивоа 1 аутономне возње. Електроника има само дјелимично могућност да утиче на вожњу, да кочи, да држи правац, односно траку или да мијења правац. Поменуте функције може да ради, али само једна у датом тренутку, односно не може да их комбинује.

Ниво 2 – Дјелимична аутоматизација. Овај ниво је познат и као "ноге с папучице, очи на пут". Овде аутомобил може да комбинује функције из претходног нивоа. Дакле, може да сам убрзава уз истовремено благо окретање волана. Међутим, возача мора бити у приправности да у сваком тренутку преузму команде. Аутоматско паркирање је типични примјер аутомне воћње на овом нивоу.

Ниво 3 – Условна аутоматизација. Систем преузима комплетну контролу, али под одређеним околностима. Тако, нпр, возило на добро обиљеженом аутопуту може само да вози, а возач не мора ни да држи руке на волану, а чак може и да не гледа шта се дешава у саобраћају, ако је довољно храбар. Ипак, мора бити у стању да преузме команде на захтјев система. Аутопилот аутоматски убрзава, успорава, скреће, па чак и избјегава препреке.

Ниво 4 – Напредна аутоматизација. Возачи могу управљати возилом на овом нивоу, али је потпуно непотребно, осим у случају веома лошег времена, попут јаких олуја или мећава. Возило практично ради све самостално. У случају да возач на захтјев не преузме контролу, аутомобил ће се сам безбједно зауставити.

Ниво 5 – Потпуна аутоматизација. Ова возила немају потребу за воланом, јер могу савладати апсолутно сваки задатак или проблем. Једино што ће људи требати да ураде јесте да кажу гдје хоће да путују.

Тренутно аутомобилски свијет је освојио ниво 2 и иде ка следећем нивоу. Технологија већ постоји чак и за ниво 4, али цијелу причу тренутно највише кочи законодавство па се чека да буду усвојени бројни закони који ће омогућити аутономну вожњу. Највећи проблем је сигурност, односно како ће се рјешавати судари која је учинила машина. Због тога је потребно додатно развити сигурносне системе, који подразумјевају бројне камере, сензоре и радаре.

Потребно је објаснити да су нивои 2 и 3 прелазни нивои, наиме, прелази се са нижег нивоа на више нивое аутоматског управљања. Прва три нивоа (0,1,2) подразумјевају да возило захтјева присуство возача, а да се за управљање самог аутомобила користе и помоћни механизми. Од трећег нивоа, следећа три нивоа (3,4,5), подразумјевају да се вожња одвија аутоматски, али и да је у зависности од нивоа потребан возач у већој или мањој мјери. Аутономна вожња је већ заступљена у возилима која су ушла у комерцијалну употребу, попут „Тесла“ и „Ауди“. Ове компаније већ имају аутомобиле другог и трећег нивоа који се користе у Сједињеним Америчким Државама и у Западној Европи [8]. Најновији модел компаније „Ауди“ А8 има потпуну аутономну вожњу до 60km/h, што је и најчешћи случај коришћења у условима градске вожње. План компаније јесте да се већ наредне године брзина за аутономну вожњу знато повећа, тј. да износи 130km/h.

2.3 ALPHА платформа

Данас постоји потреба за мноштво *ADAS* апликација како би учинили вожњу за возача опуштенију и што је још битније сигурнију. Аутомобилска индустрија тежи да постигне један циљ, а то је аутопилот.

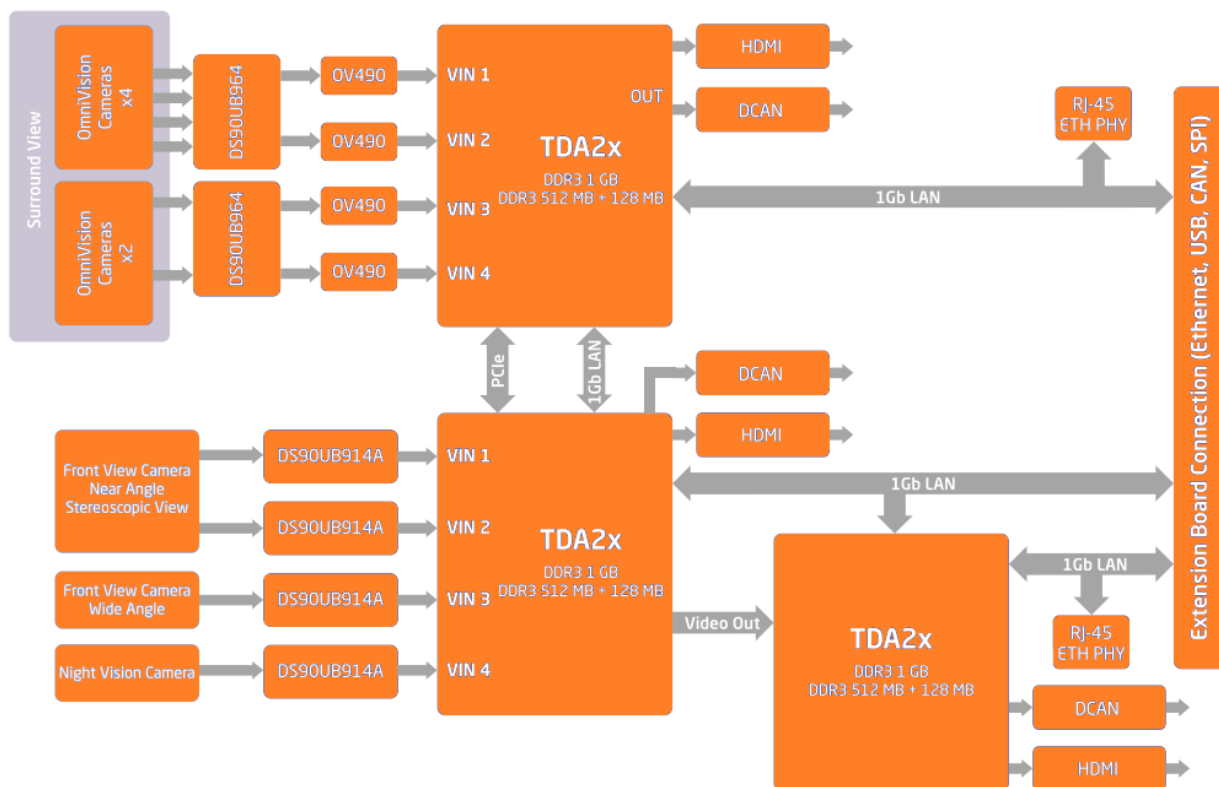
Морамо бити у могућности да сагледамо све догађаје и предмете који се налазе око возила да би дјеловали у свијету какав познајемо. Аутомобили користе разне сензоре (ултразвучне, радар, лидар, камере, идт) како би уочили објекте у свом окружењу. Након добијања информација са свих сензора, потребно их је обрадити и даље прослиједити инструкције возилу, како би знало како у ком тренутку да се понаша. Све податке треба

обрадити у реалном времену. Из тих разлога потребна је јака платформа са више чипова који сваки понаособ у себи садржи интегрисане компоненте једног рачунара (енг. *SoC*, *System on Chip*) који ће да обрађују податаке, а због своје намјене такви чипови су пројектовани да је сваки приближно величине металне кованице а довољно процесорски моћни да обраде симултано огромне количине података.



Слика 3: *ALPHA ADAS* платформа

Платформа се састоји из три *TDA2xx* [9] чипа (енг. *SoC*) произвођача *Texas Instruments-a*. Први *SoC* се назива *SCV* (енг. *Surround Camera View*). Повезан је на шест канала видео улаза (енг. *VIN*, *Video INput*) и као што му и име каже користи се за приказ околине возила. Други *SoC* се назива *FFN* (енг. *Front View Camera Near Angle Stereoscopic View*, *Front View Camera Wide Angle*, *Night Vision Camera*). Он подржава четири видео улазна канала и намјењена је за снимање пута испред возила. Трећи *SoC* се назива *FUS* (енг. *FUSion*). Он се користи за синтезу информација добијених са сензора прикључених на прва два *SoC-a*. У случају када су подаци добијени обрадом на прва два *SoC-a* у противречности, алгоритми који се извршавају на трећем *SoC-у* задужени су да одлуче које податке ће сматрати тачним. Ова три *SoC-a* су међусобно повезани разним магистралама.



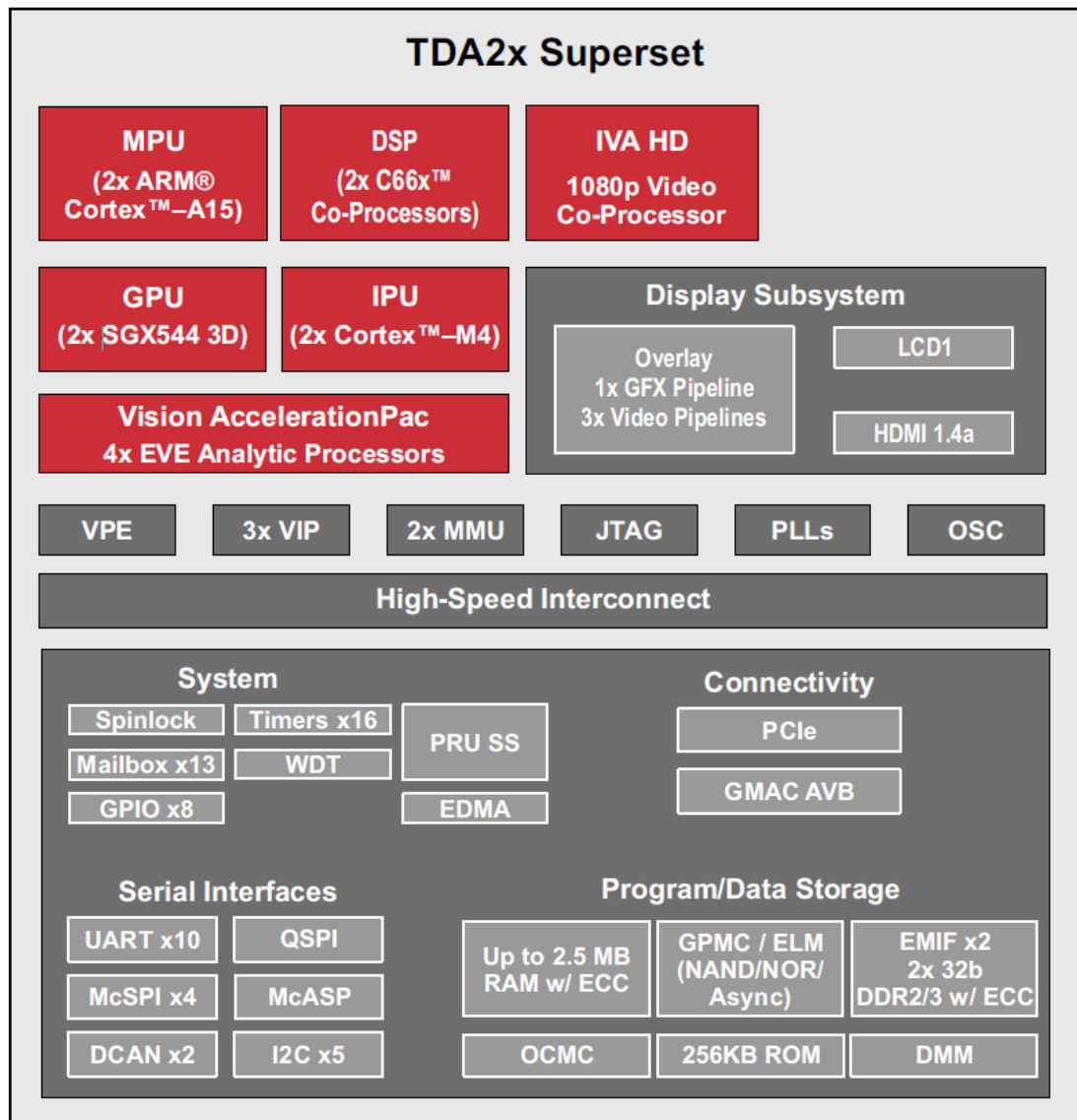
Слика 4: Шематски приказ ALPHA платформе

2.3.1 TDA2x SoC

Са шематског приказа са слике 5. можемо да закључимо да је ријеч о мултипроцесорском систему. Присутна су четири двојезгарна процесора опште намјене, од којих су два архитектуре *ARM Cortex M4*, са ознакама *IPU 1* и *IPU 2* и два архитектуре *ARM Cortex 15*, са ознаком *MPU*. Присутна су и два процесора намјењена дигиталној обради сигнала, архитектуре *Texas Instruments C66x*, са ознаком *DSP*. На *SoC*-у постоје и четири јединице за обраду видео садржаја ниског и средњег нивоа (енг. *Embedded Vision Engine – EVE*). Присутна је и *VPE* (енг. *Video Processing Engine*) компонента, чија је функција обрада видео садржаја.

За комуникацију са рачунаром опште намјене користи се *UART* (енг. *Universal Asynchronous Receiver/Transmitter*) спрега, као и *Ethernet* спрега. *I2C* (енг. *Inter-Integrated Circuit*) спрега се користи за комуникацију са јединицама за десеријализацију података допремљених са камера. Апликација и почетни пуњач се учитавају са *Micro SD* картице. Преко *HDMI* (енг. *High Definition Media Interface*) спреге, добија се излазна слика на монитору. Ради формирања излазне слике користи се дисплеј подсистем (енг. *Display SubSystem*).

Између прва два процесора постоји *PCIe* магистрала која се користи за размјену података. Ова магистрала је супербрза, до 4Gbps за генерацију 2 [10]. Између другог и трећег система на чипу постоји магистрала преко које се може преносити видео у HD (енг. *High Definition* – висока резолуција) резолуцији (1280x720).



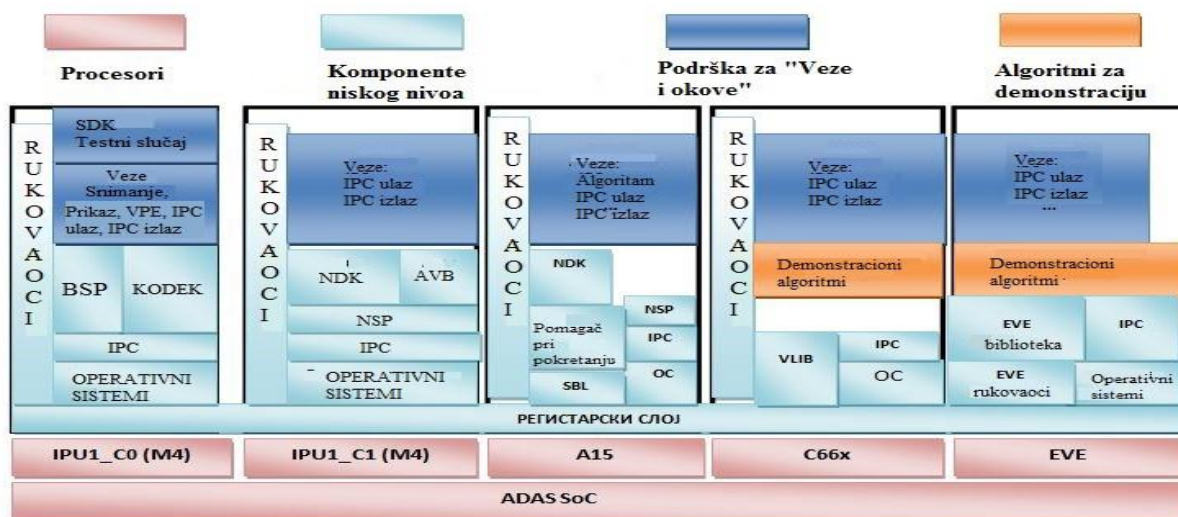
Слика 5: Шематски приказ *Texas Instruments TDA2x SoC-a*

2.4 Развојно окружење *VSDK*

Развојно окружење које је кориштено за развој апликација јесте окружење за развој програмске подршке за *TDA2x* чип (енг. *VSDK, Vision Software Development Kit*). *VSDK* је мултипроцесорска платформа за развој софтвера за *ADAS*. *VSDK* садржи *DSP* и *EVE* библиотеке са основним софтверским драјверима и алгоритмима за анализу и обраду

сигнала, као и поједине примјере апликација. Као такав представља погодно развојно окружење. *DSP* и *EVE* библиотеке нуде оптимизоване функционалности на ниском и средњем нивоу обраде слике у развијању и имплементацији алгорита на апликативном слоју. Софтверски радни оквир дозвољава кориснику да развија креирање различитих токова података за *ADAS* апликације што укључује прихватање видео сигнала, препроцесирање видео сигнала, алгорита за анализу и видео приказ.

VSDK је базиран на радном оквиру „*Везе и ланци*“ (енг. *Links and Chains*). *VSDK* инсталациони пакет укључује све алате и компоненте потребне за покретање апликација, а то подразумијева алате за генерисање кода, *BIOS* (енг. *Binary Input Output System*), *IPC* (енг. *Inter Processor Communication*), руковаоце (енг. *starterware*), *BSP* драјвер (енг. *Board Support Package*), мрежне стекове, кодеке као и кернел алгоритме. На слици 6. је дат детаљан опис *VSDK* слојева.



Слика 6: TI Vision SDK слојевита архитектура софтвера

2.5 Генерисање и учитавање алгорита

Да би се алгоритам пустио у рад, неопходно је одрадiti пар корака. Први од њих јесте генерисање почетног пуњача (енг. *bootloader*). *Bootloader* је рачунарски програм који учитава оперативни систем или неки други софтвер за рачунар. У процесу покретања учитава се и покреће софтвер. *Bootloader* се учитава у главну меморију из интерне меморије. Затим, почетни *bootloader* учитава и извршава процесе који довршавају покретање. Код покретања долази до предефинисаности локације. Ако је та локација ограничена, из неког разлога, тај примарни *bootloader* позива иницијални *bootloader* другог

степенa (енг. *Secondary Bootloader*). У нашем случају *bootloader* другог степена се назива *MLO*.

Следећи корак јесте генерисање *AppImage-a*. *AppImage* је бинарни фајл у коме се налазе изгенерисани сви алгоритми спремни за рад.

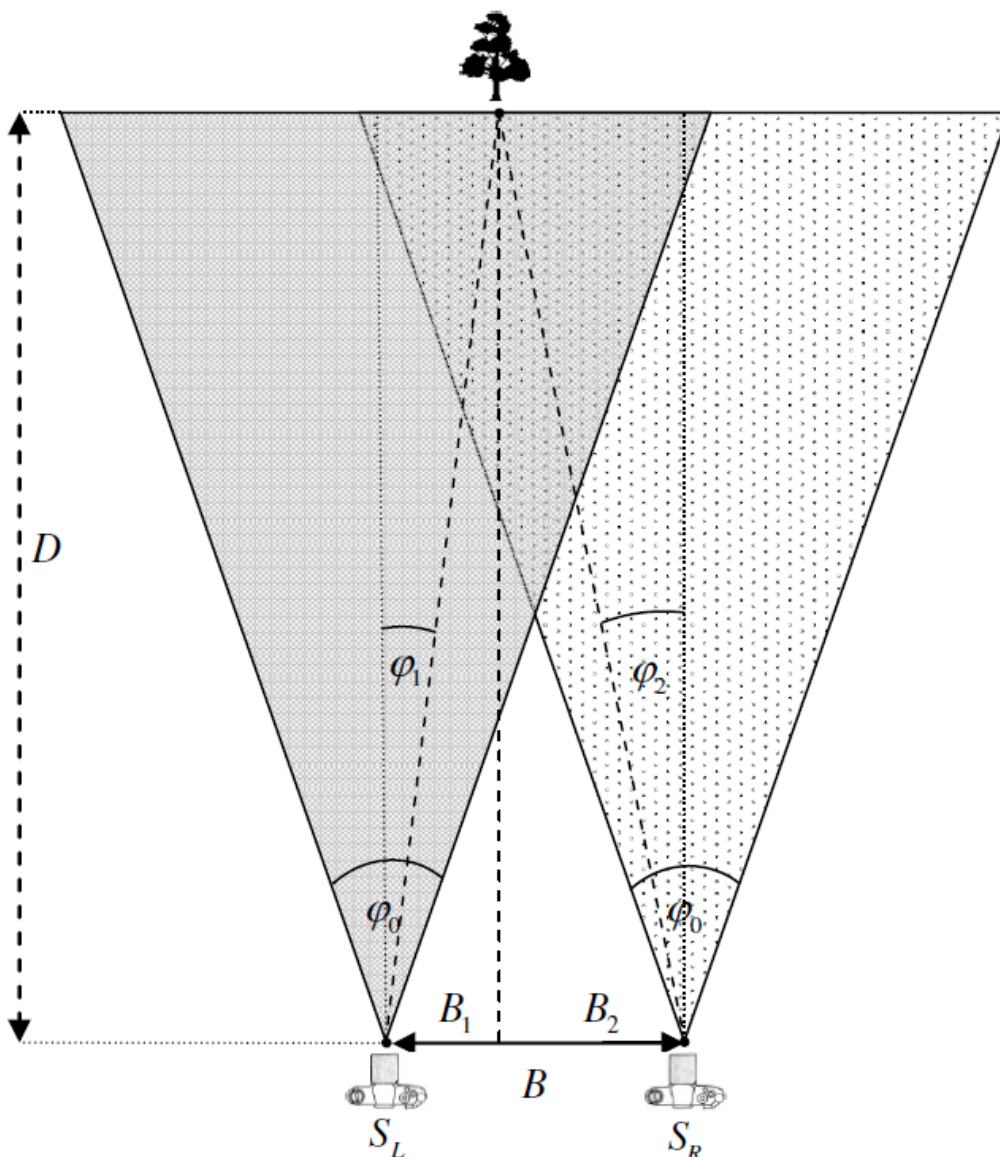
Ова два фајла заједно се спусте на *SD* картицу и на тај начин покреће се *ALPHA* плоча.

2.6 Стереоскопија

Стереоскопија [11] је техника која се користи за снимање и представљање стереоскопске 3Д слике. То може створити илузију дубине користећи двије слике снимљене са само мало другачијим положајем. Постоје два начина снимања стереоскопских слика: коришћењем специјалних стерео камера са два објектива или система са двије камере спојене са једним објективом. Стереоскопске слике допуштају манипулацију угловима да бисмо израчунали удаљеност од камере до изабраног објекта унутар слике. На рачунање удаљености утичу разни технички подаци као што су жижна даљина, удаљеност између камера, различити углови праволинијске дистанце у односу на објекат које камере снимају. Тачност позиције зависи од резолуције слике, оптичког изобличења и удаљености између камера.

Методe за мјерење удаљености до неких објеката могу бити подјeљене на активне и пасивне. Активне методе су мјерење удаљености слањем сигнала према објекту (нпр. ласерски зрак, ултра звук итд.) док пасивни примају информације о објектима (обично свјетлом). Међу пасивним, најпопуларнији су они који се ослањају на стереоскопски метод мјерења. Главна карактеристика методе је да се користе двије камере. Удаљеност објекта може се израчунати из релативне разлике позиције објеката на обје камере.

Британски научник Чарлс Ветстон (енг. *Charles Wheatstone*) направио је стереоскопске слике и уређај за гледање. Стереоскопска слика може бити сликана са паром камера, које подјећају на људске очи. Правила која треба да се испоштују да бисмо добили пар стереоскопских слика су: камере треба да су постављене хоризонтално слике треба да буду направљене истовремено са обје камере, посебно ако причамо о видеу, а не статичним сликама. Стереоскопска слика нам омогућава да израчунамо дистанцу између камера и објекта који се налази испред њих. У овом случају причамо о аутомобилу и објектима (пјешаци, бициклисти, друга возила и остали учесници у саобраћају) који се налазе испред возила.



Слика 7: Слика објекта (стабла) снимљена са двије камере [11]

На слици 7. приказано је како се објекат види са двије камере. Нека прва слика буде снимљена са локације S_r и лијева слика на локацији S_l . B представља удаљеност између камера, а φ_0 представља угао погледа који има камера. Позиција објекта, дистанца D , може да се израчуна на основу геометријских извођења. Дистанцу између камера, B , можемо да изразимо као суму дистанце [11] B_1 и B_2 :

$$B = B_1 + B_2 = D \tan \varphi_1 + D \tan \varphi_2:$$

У случају да су оптичке осе паралелне, гдје φ_1 и φ_2 представљају углове између оптичке осе сочива камере и одабраног објекта. Дистанцу рачунамо на следећи начин[11]:

$$D = \frac{B}{\tan \varphi_1 + \tan \varphi_2}$$

Ако су познати дистанца између камера (B), број хоризонталних пиксела (x_0), угао гледања камера (φ_0) и хоризонтална разлика између два иста објекта на обје слике ($x_L - x_D$), онда дистанца објекта може да се израчуна[11]:

$$D = \frac{Bx_0}{2 \tan\left(\frac{\varphi_0}{2}\right)(x_L - x_D)}$$

У овом раду ће бити представљена имплементација једног оваквог алгоритма, који је тестиран на *ALPHA* плочи, користећи двије камере и алгоритам за детекцију објеката.

3. Концепт рјешења

У овом раду је представљено рачунање дистанце између аутомобила и објекта испред аутомобила, користећи двије предње камере (енг. *Front view camera*). Техника која је примјењена да би се дошло до што прецизнијих резултата јесте стереоскопија.

У овом раду ће бити реализовани и разлози зашто је боље користити двије предње камере, а не само једну, која је разлика у рачунању дистанце са једном камером, и са двије, као и прецизност добијене вриједности дистанце са двије камере.

За реализацију овог рада користили смо алгоритам за детекцију објеката (енг. *Object detection*). У току рада ове апликације и прослеђивањем неходних параметара, имамо могућност да изачунамо прецизнију вриједност дистанце између објекта и возила.

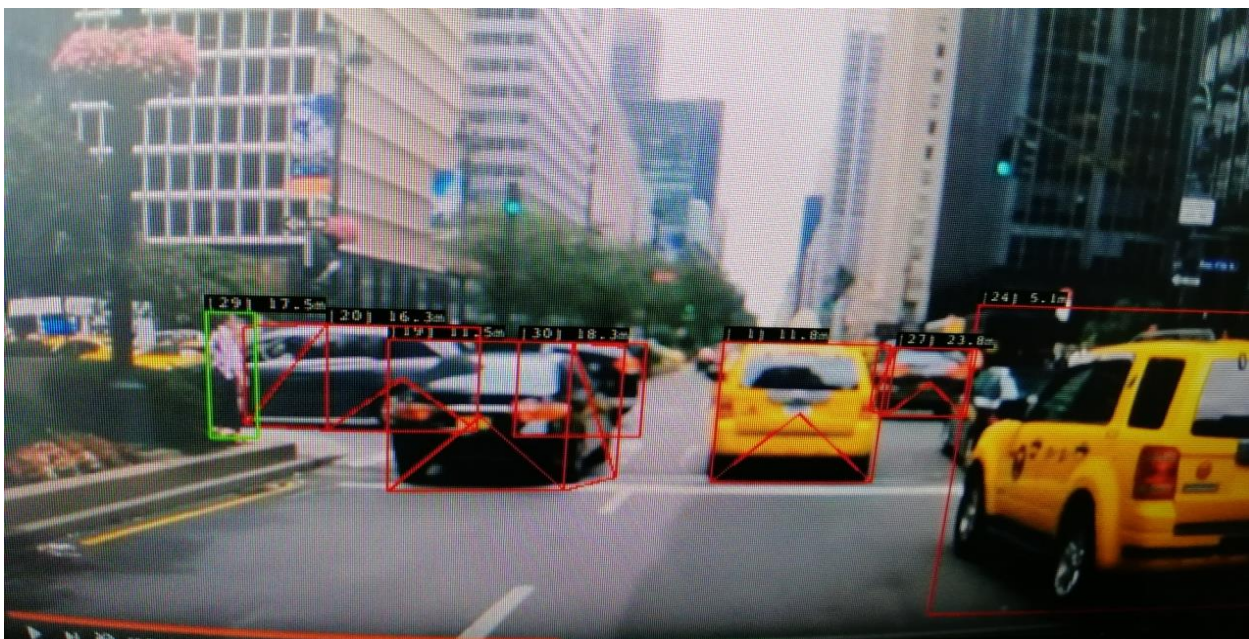
3.1 Детекција објеката – пјешака и возила (*Object detection*)

Детекција објеката је један од алгоритама који су дио *ADAS-a*. Постоје различити алгоритми који у зависности од жељених резултата могу препознавати пјешаке, бицикле, мотоцикле, аутомобиле, аутобусе, теретна возила, као и друге објекте који се налазе на путу или крај пута (животиње, грање...). Такође, ови алгоритми могу да препознају и саобраћајне знакове као и семафоре. Неки од ових алгоритама могу бити посебно развијени, па ће нпр. детекција саобраћајних знакова бити дио цјелокуног система, али независно од детекције пјешака. Слично је и за семафоре и детекцију задњих свјетала на аутомобилу, гдје се алгоритам обучава и прилагођава другачијем типу објеката који се детектују.



Слика 8: Рад алгоритма за детекцију објеката

Алгоритам за детекцију објеката је веома сложен. Резултати овог алгоритма се могу видјети на екрану као правоугаоником уоквирени детектовани објекти. Детектовани објекат може бити различитог типа као што смо већ спомињали (пјешак, бициклиста, аутомобил, аутобус, камион...). У зависности који је тип објекта, боја правоугаоника којим се исцртава објекат биће другачија, нпр. зелена за пјешаке, црвена за аутомобиле итд. Такође, овај алгоритам поред информације о типу објекта, садржи и информацију о удаљености објекта од камере. Ову информацију, такође, можемо да видимо исписану на екрану. На слици 9. виде се детектовани аутомобили и детектован пјшак алгоритмом који је коришћен у раду.



Слика 9: Детектовани аутомобили и раздаљина до њих

Информације које су значајне јесте удаљеност објекта од камере. У овом раду ће бити приказан начин рачунања удаљености објекта користећи двије камере, а не само једну што је пракса. На основу добијене информације о удаљености, из самог алгоритма за детекцију објеката, и помоћу претходно објашњене стереоскопије, добићемо нови алгоритам за детекцију објеката. Овакав алгоритам ћемо тестирати на *ALPHA* плочи, као и крајњи резултат овог рада. Моћи ћемо да упоредимо резултате удаљености које добијемо са једном и са двије камере.

4. Програмско рјешење

У овом поглављу биће приче о процесу писања новог *use-case-a* и блока *Distance* који садржи функцију *distance* за израчунавање дистанце на основу добијених података са двије камере. Нови *use-case* обједињава претходно поменућу апликацију *Object detection* чији се садржај провлачи кроз блок *Distance*, како би добили прецизније резултате удаљености аутомобила и објекта.

4.1 Процес писања новог *use-case-a*

Проучавањем свих блокова којима располаже *VSDK* окружење, долазимо до закључка да би требало да је најбоље рјешење да помоћу двије камере повезане на *ALPHA* плочи, чије податке проследимо кроз алгоритам за детекцију објеката, а затим такав добијени садржај провући кроз блок *Distance*, који ће прецизније да израчуна удаљеност возила и детектованог објекта. Читав овај дијаграм тока се завршава са *Display* блоковима, који служе да на крају обраде алгоритма добијемо слику на монитору.

У окружењу, какав је *VSDK*, за генерисање новог *use-case-a*, потребно је написати *.txt* фајл. Тај фајл садржи написане блокове, оним редом како желимо да се наш алгоритам извршава, нпр:

```
Capture -> Split -> Select -> Dup -> Display_Video
```

Након писања овог фајла, окружење генерише остале неопходне фајлове уз помоћ *use_case_gen* алата. Овај алат нам помаже да брзо и лако повежемо блокове у функционалну цјелину. Преостаје да напишемо сам алгоритам, као и подешавања генерисаних блокова.

4.2 Портовање алгоритма за детекцију објеката на *ALPHA* плочи

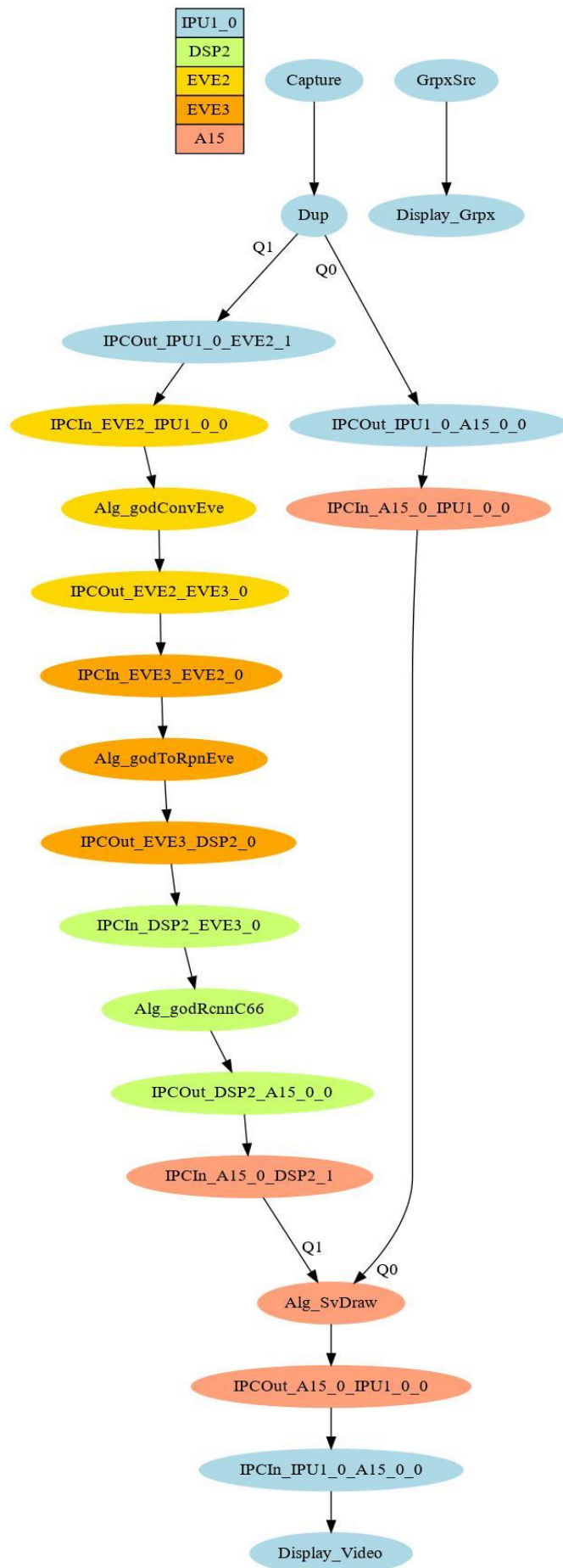
Алгоритам који је кориштен за детекцију објеката је истрениран на неуронским мрежама. То су математички алгоритми за обраду слике, односно видеа. Хардверска платформа и њене особине морају нам добро бити познате да би се овај алгоритам могао извршавати.

Овај алгоритам је напредни алгоритам, и због таквог неопходно нам је да имамо сазнање о подацима и типу система на чипу на који се алгоритам портује да би што боље

искористили ресурсе које дата хардверска платформа посједује. Алгоритам за детекцију објеката је портован на хардверске платформе, гдје је *TDA2x* главни систем на чипу, али није био портован на *ALPHA* платформу. С тим у вези неопходне су одређене модификације, прилагодити специфичностима хардверске платформе и тек онда видјети ефекат његовог рада.

Једна од измјена коју је увела *ALPHA* платформа јесте коришћење других камера за снимање околине. Убачени су десеријалајзери и серијалајзери који омогућују комуникацију камере са системом на чипу.

Као улаз користи се камера, које се налазе на предњем дијелу аутомобила. Подаци добијени са камере се обрађују даље кроз алгоритам, на *DSP*, *A15*, *IPU*, *EVE* процесорима. Тако обрађени подаци се шаљу на екран. На слици 10. приказан је дијаграм тока података система *VSDK-a* за *Object detection* апликацију, од преузимања са камера до формирања излазне слике.

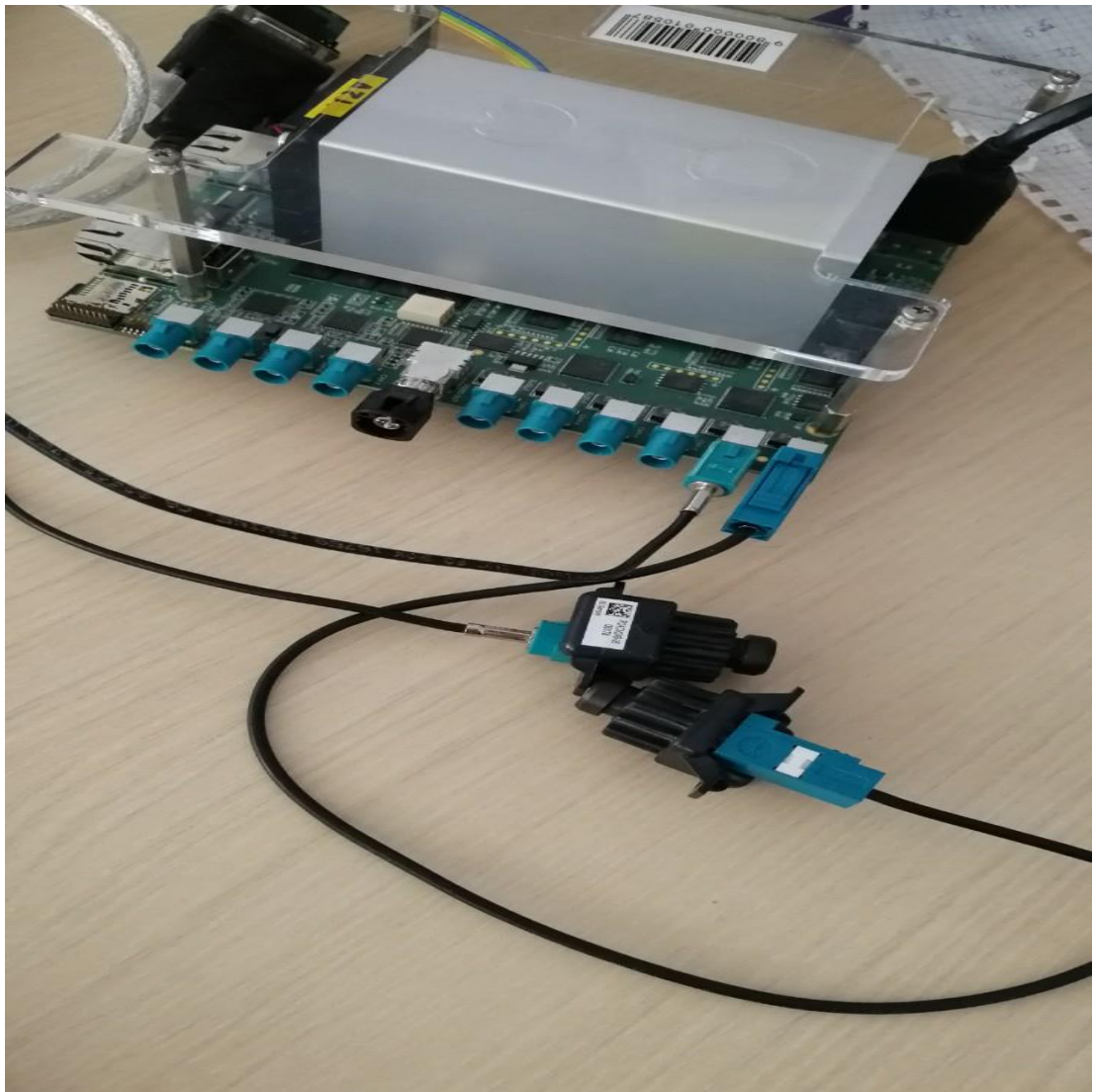


Слика 10: Дијаграм тока *Object Detection* алгоритма

Сваки блок у дијаграму представља нит (енг. *thread*), која се извршава на неком од расположивих процесорских језгара унутар чипа. У горњем лијевом углу представљени су процесори који се користе у том алгоритму, *IPU1_0*, *DSP2*, *EVE2*, *EVE3*, *A15*. Стрелице представљају прослеђивање података између нити, које се врши помоћу блокова података, бафера.

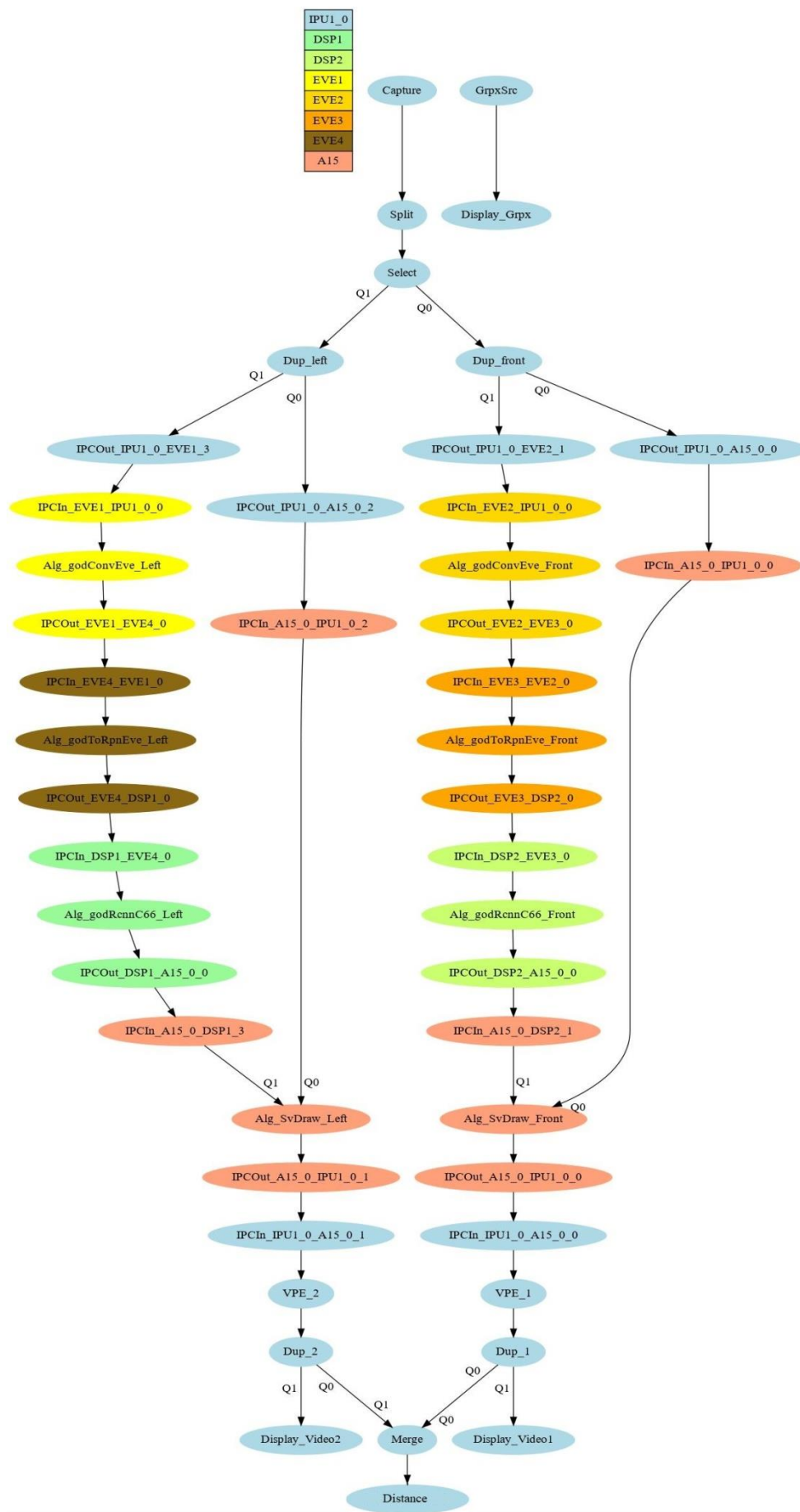
4.3 Прављење новог *use-case-a* са двије камере

Идеја овог рада, како смо већ о томе причали, јесте да одредимо удаћеност објекта од возила, користећи двије камере. Неопходно за ову редализацију јесте *ALPHA* платформа која, како смо већ спомињали, има три чипа. За нашу реализацију користимо први *SCV* чип, на који су повезане двије камере, као што је приказано на слици 11.



Слика 11: Приказ повезивања камера са *SCV SoC*

Use_case_gen алат, о коме је било ријечи на почетку овог поглавља, нам помаже да брзо и лако повежемо блокове у функционалну цјелину. На слици 12. приказан је дијаграм тока података система *VSDK-a*. Подаци добијени са камере се обрађују даље кроз алгоритам, на *DSP1*, *DSP2*, *A15*, *IPU1_0*, *EVE1*, *EVE2*, *EVE3*, *EVE4*, *A15* процесорима, што можемо видјети на слици у лијевом горњем углу.



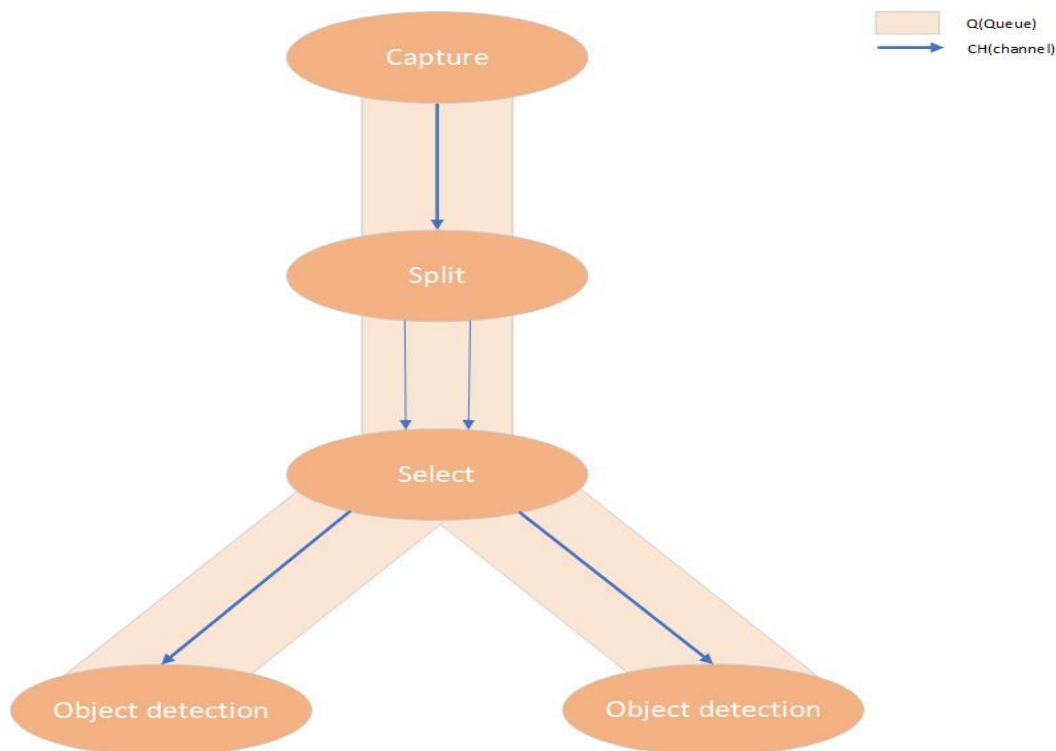
Слика 12: Дијаграм тока података *Object Detection* алгоритма са двије камере

4.3.1 Улазни и излазни подсистеми

Као што смо могли видјети у претходном поглављу, алгоритам за детекцију објеката креће од *Capture* блока, који како смо већ навели, прикупља податке са камера и прослеђује на обраду. Обзиром да се наш проблем заснива на рачунању дистанце на основу двије камере, у току реализације програмског рјешења, неопходно је да податке прикупљене и са једне и са друге камере обрадимо кроз алгоритам за детекцију објеката. С тим у вези, на један *use-case* би требали да имамо два блока *Capture*, гдје би један представљао улаз података са једне камере, а други блок улаз података са друге камере. Овде наилазимо на први проблем, платформа има своја ограничења. Један од тих ограничења јесте да један *use-case* не може имати више од једног *Capture* блока. Самим тим потребно је додатно измијенити улазне блокове у ове алгоритме. *Capture* блок је први блок који потребно другачије конфигурисати. Он поред подешавања мноштва параметара, укључује и избор камера путем функције:

```
Void ChainsCommon_Amv_Start_Camera(
    Amv_ChipType chipType,
    Chains_CaptureSrc pCaptureSr,
    CaptureLink_CreateParams *pPrm,
    UInt32 captureInWidth,
    UInt32 captureInHeight,
    AMV_VideoSensor_Id portId[],
    UInt32 numOfSensor,
    Bsp_VidSensorConfigRegs **vidSensorConfigRegs,
    UInt32 *vidSensorConfigRegsSizes,
    Bsp_VidSensorFlipParams flipParams[]
),
```

На овај начин се прослеђује идентификациони број камера које желимо да укључимо, у овом случају то су прве двије камере (*SC_VIDEO_SENSOR_1_2*).



Слика 13: Улазни подсистем

На слици 13. су представљени улазни блокови у заједничком *use-case-y*. Између блокова се налазе *Q*-ови. Сваки од ових *Q*-ова садржи одређен број канала. Канали нису ништа друго него подаци добијени са камера.

Прикупљени подаци са камера шаљемо даље на *Split* блок. *Split* добија један канал, прикупљен са камера. У каналу се налазе слике од обје камере. Затим их овај блок „сплитује“, тј. рашчлањује на засебне канале. У овом случају имамо два излазна канала (*CH1-CH2*). Таква два канала од *Split* блока иду до *Select* блока. *Select* је задужен да примљене канале проследи на одговарајући излаз (*Q0* или *Q1*), у зависности од тога како је конфигурисан. Оба канала пролазе пролазе кроз исти алгоритам за детекцију објеката, као што се види на слици 12. и 13. Са десне стране (*Q0*) се прослеђују и обрађују подаци са прве камере (канал *CH1*), док са лијеве стране (*Q1*) се прослеђују и обрађују подаци са друге камере (канал *CH2*).

Свака грана је задужена за посебну обраду тог канала. У овом случају једна грана представља канал добијен са прве камере, тј. податке добијене са прве камере, док друга грана представља канал са друге камере. Обје гране су обрађене на исти начин, обрада података је обрађена кроз алгоритам за детекцију објеката. Сваки од ових канала дуплира бафере помоћу блока *Dup*. Овај садржај иде даље и обрађује се кроз задати алгоритам за детекцију објеката и прослеђује се на екран.

Као што се види са слике 12. сви блокови се не извршавају на једном процесору. Међупроцесорска комуникација (енг. *IPC – Inter Processor Communication*) се користи за сарадњу различитих задатака који се покрећу на различитим процесорима на чипу. *IPCIn* је блок намијењен достављању бафера података са различитог процесорског језгра, а *IPCOut* је блок намијењен прослеђивању бафера другом процесорком језгру. На крају дијаграма се налази *Display* блок, који обрађен садржај прослеђује на видео канал, који се даље прослеђује на излаз.

У десном дијелу на слици налази се блок *GrpхSrc* (енг. *Graphics Source*), извор графичког садржаја. Након формирања, бафери са овим садржајем прослеђују се блоку *Display_Grpх*, који их даље прослеђује дисплеј подсистему, тачније графичком каналу. На екрану добијамо слику на којој су приказани сви објекти који учествују у саобраћају, уоквирени у правоугаоник са различитим бојама, гдје свака боја представља један скуп објеката (пјешаци, аутомобили...). Изнад самих оквира се налазе подаци о удаљености објекта.

4.4 *Distance* блок

Резултати детекције објеката биће приказани на излазу на екрану, гдје ће бити исцртани и приказане информације о удаљености објекта од возила са ког се снима. Будући да је то дио гдје се информације добијају, неопходно је било ући у саму имплементацију линка за исцртавање графике и у њему пронаћи све информације.

4.4.1 Прослеђивање података из алгоритма за детекцију објеката

Сам линк за исцртавање је .с фајл под називом *SvDrawLink_algPlugin.c*. Посједује функције којима од добијених резултата из алгоритма издваја корисне податке од података који нису поуздани. На тај начин, он врши филтрирање резултата који ће бити исцртани. Један од тих филтера јесте и тај да ли се објекат налази у видном пољу камере, тако да се он може потпуно уоквирити правоугаоником.

Од свих података, од највећег значаја је информација о ком типу објекта је ријеч, дистанца детектованог објекта и положај детектованог објекта. Структура у коју се смијештају подаци садржи поља *Int x*, *Int y*, *Int h*, *Int w* која представљају координате објекта, односно исцртавање правоугаоника око датог објекта у тренутку његовог детектовања, висина и ширина правоугаоника.

Ова поља су нам од великог значаја. Прослеђивањем ових података на *Distance* линк можемо да одредимо позицију објекта у односу на обје камере и да тај податак примјенимо

у формули за стереоскопију. Ако погледамо у форму то је дио ($x_L - x_D$). *VSDK* располаже разним уграђеним алатима, као што је на примјер системски бафер (енг. *System buffer*), који служи да прослеђујемо информације између различитих линкова. Користећи могућности које пружа *System buffer* искористили смо да структуру системског бафера проширимо за ове параметре како би успјешно проследили неопходне информације до *Distance* линка, која изгледа овако:

```
typedef struct System_Buffer_s {
    UInt32 w;
    UInt32 x;
    UInt32 y;
    UInt32 h;
} System_Buffer;
```

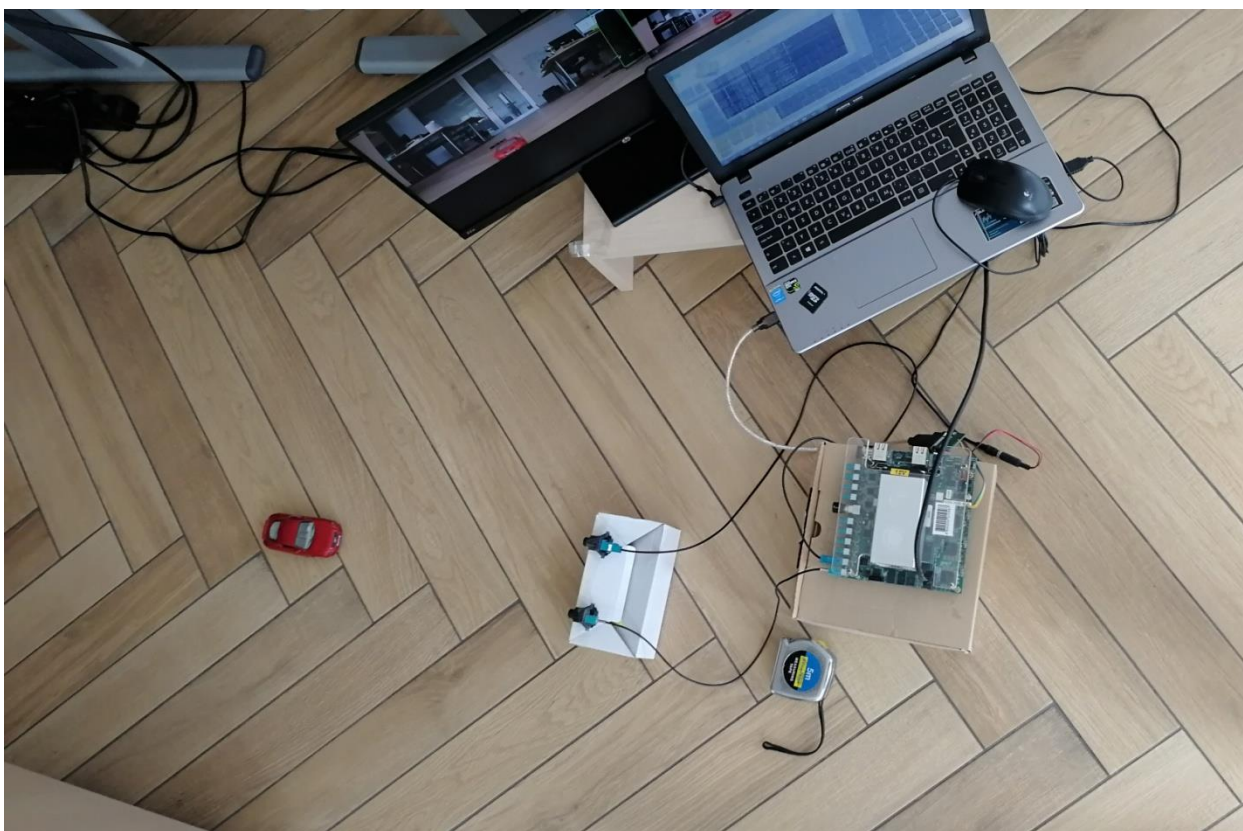
Када смо проследили праве информације, следећи корак јесте да израчунамо положај детектованог објекта. Да би дошли до резултата, на основу прослеђених параметара рачунамо средину ширине правоугаоника који се приликом детекције објекта исцртава. Примјеном аритметичке средине на ове вриједности добијамо средину ширине оквира око препознатог објекта на објекту слике, а самим тим и тачну позицију објекта израчунату у пикселима. Дио кода који ради рачунање дистанце изгледа овако:

$$Distance = (B * x_0) / (2 * \tan(camera_angel/2) * (w_1/2 - w_2/2));$$

Где је B размак између камера, x_0 је ширина улазне слике у пикселима, $camera_angel$ је угао видног поља камере, w_1 и w_2 представља ширину правоугаоника који се исцртава око објекта. На пола тих ширина јесту пиксели на којима се налазе детектовани објекат.

5. Тестирање

У овом поглављу биће приказано тестно окружење, начин тестирања и добијени резултати. Упоредиће се тестни случајеви и шта је добијено. Тестирање и реализација обављани су на *ALPHA* плочи, на првом *SCV* чипу. Циљ овог рада јесте да се представе двије камере као боље рјешење за рачунање удаљености возила од објекта. За потребе тестирања, користили смо макету аутомобила, двије камере повезане на платформи, што можемо да видимо са приложене слике 14.

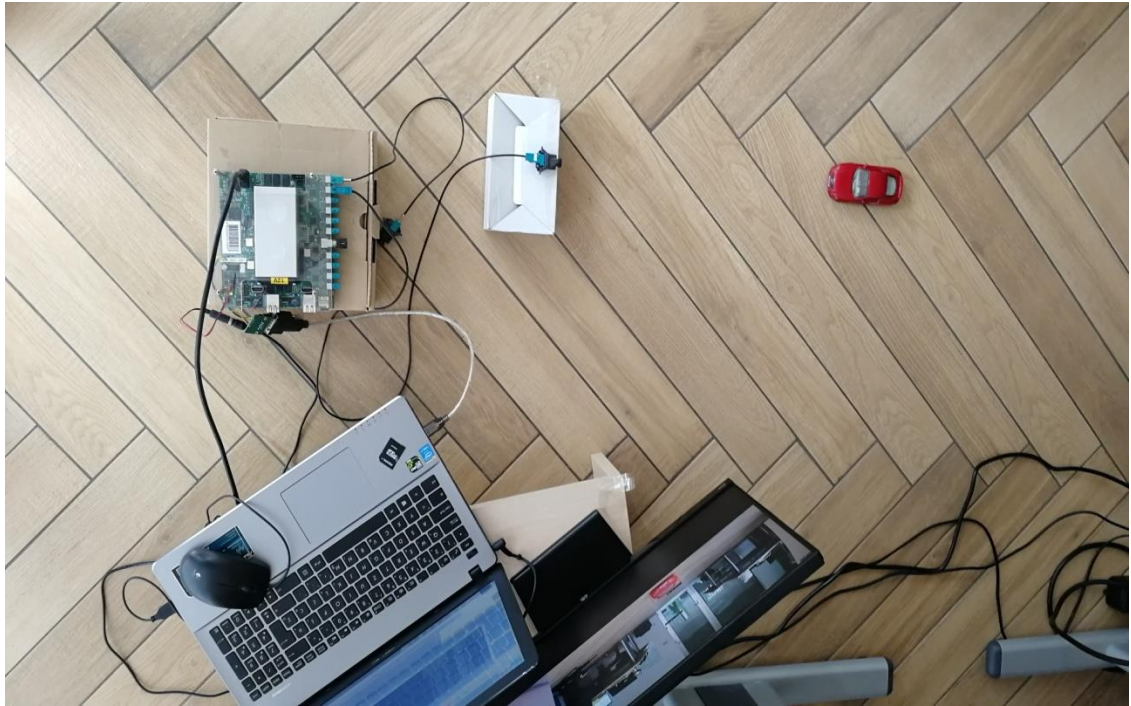


Слика 14: Приказ радног окружења

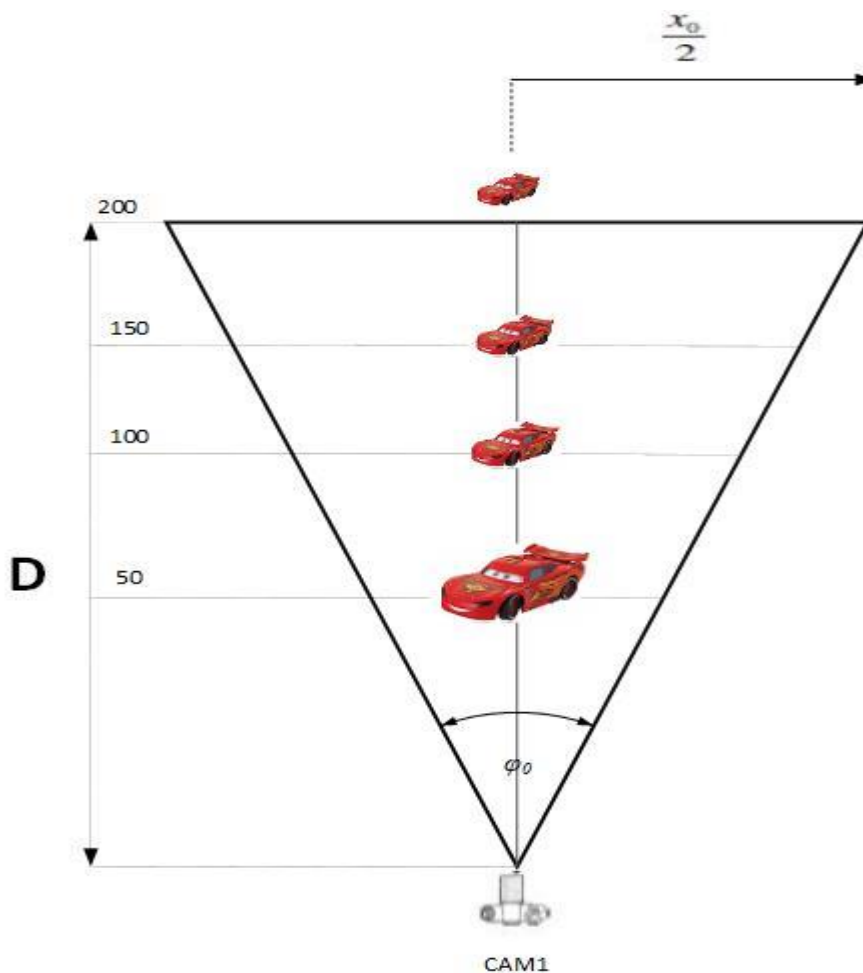
Први корак кроз ово тестирање јесте утврђивање како алгоритам за детекцију објеката ради са једном камером када детектујемо играчку у облику аутомобила коју користимо у сврху тестирања. У табели 1. приказани су резултати удаљености објекта од камере, које добијемо коришћењем једне камере. Тестирање је извршавано на различитим удаљеностима између камере и објекта (50цм, 100цм, 150цм, 200цм, 250цм).

D[cm]	50	100	150	200	250
1CAM	56	116	164	217	242

Табела 1: Резултати одређивања удаљености за различите удаљености објекта од камере, добијени коришћењем једне камере



Слика 15: Приказ тестног окружења са једном камером



Слика 16: Приказ видног поља камере у тестном окружењу

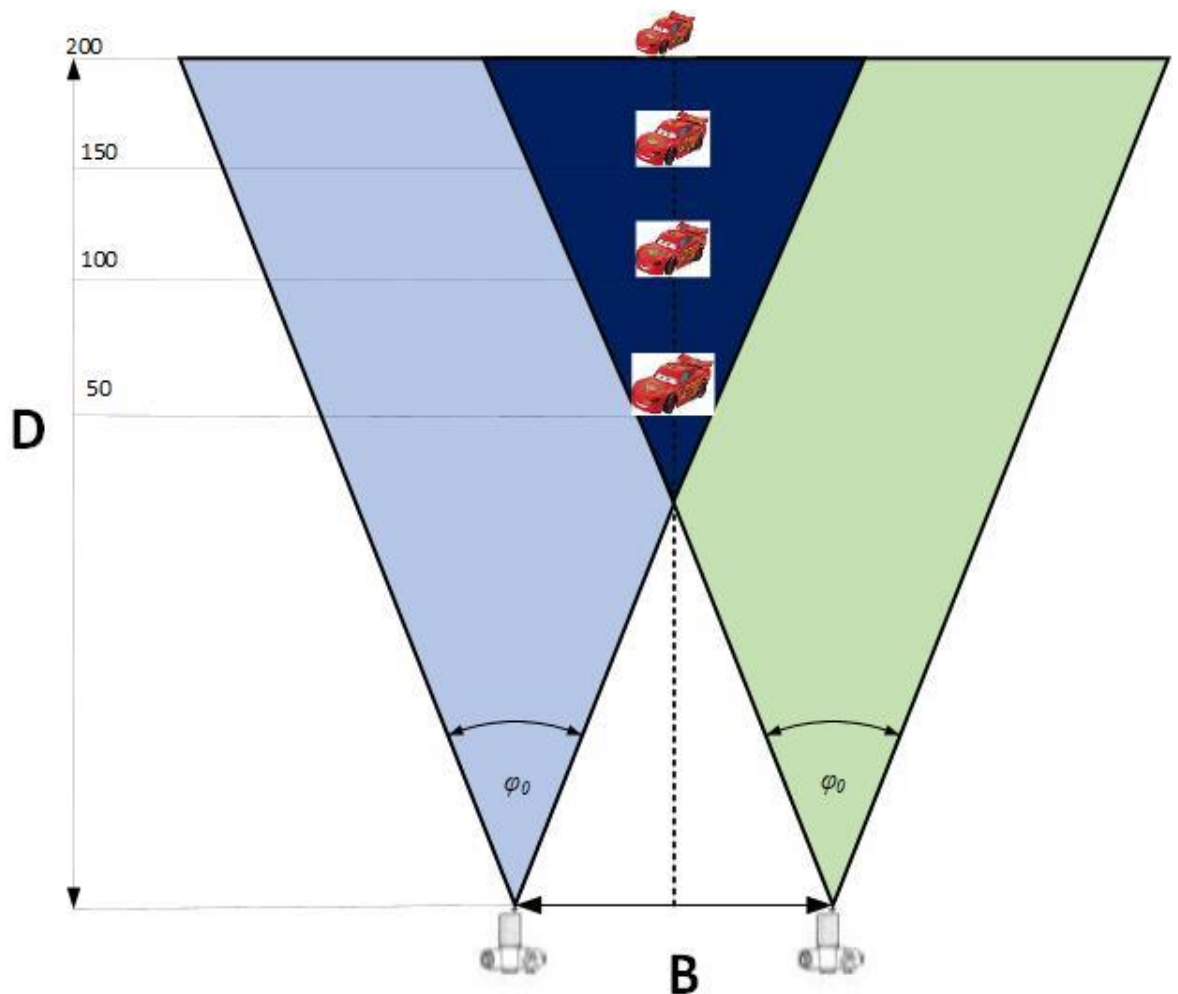
На слици 15. приказано је тестно окружење за тестирање алгорита за детекцију објекта са једном камером. Док на слици 16. дат је сликовит приказ поменутог тестног окружења. Дат је приказ снопа видљивости камере. Сlike аутомобила представљају тачке удаљености аутомобила од камере (50цм, 100цм, 150цм, 200цм, 250цм) на којима смо тестирали алгорита за детекцију објеката користећи једну камеру.

У табели 2. дати су резултати удаљености објекта од камере, које добијемо коришћењем двије камере. Да би могли што прецизније да дамо резултате, и ово тестирање је извршено на истим удаљеностима објекта од камера (50цм, 100цм, 150цм, 200цм, 250цм).

D[cm]	50	100	150	200	250
B[cm]-2CAM					
10	55	111	158	189	245

20	54	105	152	195	X
30	46	102	148	180	X
40	57	89	131	176	X

Табела 2: Резултати одређивања удаљености за различите удаљености објекта од камере, добијени коришћењем двије камере

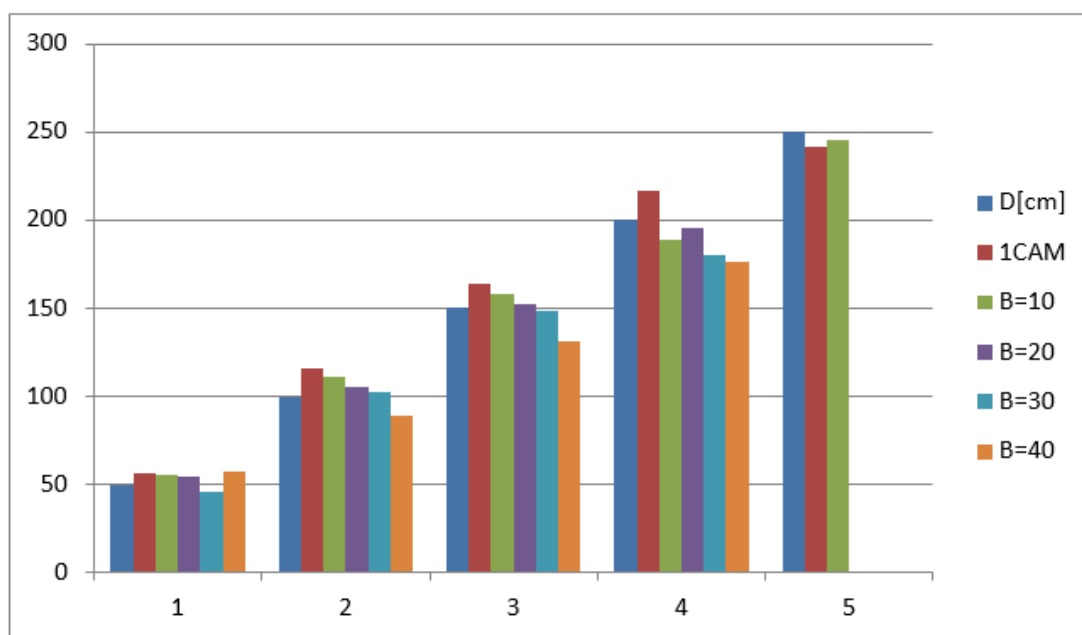


Слика 17: Приказ видног поља двије камере у тестном окружењу

На слици 17. приказана је илустрација видних поља обје камере. Са слике видимо да ове камере имају један дио који им је заједнички (тамно плави троугао). Над објектима који су видљиви са обје камере, као што је било приче у претходним поглављима, примјењена је стереоскопија, на основу добијених података. Сlike аутомобила, као и у тестирању са једном камером, представљају тачке удаљености аутомобила од камере (50цм, 100цм,

150цм, 200цм, 250цм) на којима смо тестирали алгоритам за детекцију објеката користећи двије камере. Пратећи резултате из табеле 2. и слику 17, можемо да размотримо добијене резултате и положај макете аутомобила у односу на двије камере. У случају када се макета аутомобила налази на удаљености од 50цм добијамо одступања од 8% до 14%. Са слике 17. видимо да се у том случају макета аутомобила налази на самој граници гдје се снопови видљивости обје камере укрштају. Када се макета аутомобила налази на удаљености од 100цм и 150цм, видимо да су и резултати прецизнији, одступања су од 2%, када је $B = 20$ цм и $B = 30$ цм, док када је $B = 10$ цм и $B = 40$ цм одступања иду до 13%. Са илустрације можемо да видимо да се на тој удаљености макета аутомобила налази на много бољем положају у односу на камере, и сама детекција макете је прецизнија, у односу када се макета налази на самој граници ова два снопа видљивости. Када се макета налази на удаљености од 200цм, одступања је од 2,5%, када је $B = 20$ цм, до 11%, када је $B = 40$ цм.

Упоредивањем резултата добијених са једном камером и са двије камере можемо да анализирамо добијене резултате. Као што се види из табеле 1, одступања измјерених од стварних вриједности се крећу у опсегу од 8% до 16%. Док из табеле 2, одступања се крећу у опсегу од 2% до 14%. Из приказаних тестних резултата, као најбоље рјешење се показао размак између камера од 20цм и 30цм и то на удаљености објекта од 100 – 150цм. За удаљености објекта већој од 200цм алгоритам није у могућности да га детектује. Ова одступања су приказана и на следећем графику.



Слика 18: Графички приказ резултата тестирања са једном и са двије камере

6. Закључак

У овом раду је приказан начин на који се рачуна удаљеност објекта од возила користећи више камера у *VSDK*-у. Циљ формирања једне овакве апликације, јесте да се што прецизније израчуна удаљеност.

Кроз тестирања могли смо да видимо коју вриједност удаљености објекта од возила даје када се користи само једна камера, а до којих смо резултата дошли користећи двије камере, примјењивањем стереоскопје. У овом раду смо приказали како се долази до тачнијих вриједности удаљености објекта од возила, користећи методу стереоскопије. Обзиром да је тестирање извршено над макетом аутомобила било би корисно ако би се тестирање спровело у реалном саобраћају и са савршено калибрисаним камерама, јер би се тек тада видјела стварна поузданост самог алгоритма. Овај рад приказује само једно од могућих рјешења, када говоримо о прецизнијим и тачнијим вриједностима које нам, као свој резултат, дају алгоритми. Даљи развој рјешења би могао да се креће правцу побољшања и оптимизације кода, као и тестирање над аутомобилима и у реалном саобраћају. Тиме би се добио поуздан и употребљив алгоритам који би потенцијално могао да се извршава у реалном времену и самим тим користи у аутомобилској индустрији.

Како смо све ближе тренутку када неће бити потребе за воланом и возачем, ово је једна од тема и проблема са којима је суочена аутомобилска индустрија. Комплекснија рјешења, која ће бити имплементирана у пракси, даће само додатно проширење тезама изнијетим у раду и потврдити смјер даљег развоја аутономне вожње.

7. Литература

- [1] M. Broy, “Challenges in automotive software engineering”, *Proceedings of the 28th international conference of software engineering*.
- [2] TI Vision SDK, Optimized Vision Libraries for ADAS Systems, <http://www.ti.com/lit/wp/spry260/spry260.pdf>, приступљено: јун 2019.
- [3] Automotive machine vision alpha reference board on Texas Instruments SoCs, <http://www.rt-rk.com/news/195-rt-rk-presents-automotive-machine-vision-alpha-reference-board-on-texas-instruments-socs>, приступљено: јун 2019.
- [4] *Annual Global Road Crash Statistics*, <http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics>, приступљено: јун 2019.
- [5] Paul A., Chauhan R., Srivastava R., Baruah M., *Advanced Driver Assistance Systems*, SAE Technical Paper 2016-28-0223, 2016.
- [6] A. Hakanovic, „Sistemi za pomoc vozacu“, Sarajevo, 2011.
- [7] Society of Automotive Engineers, <https://www.sae.org>, приступљено: јун 2019.
- [8] *Audin says 2019 A8's Level 3 self-driving tech capable of doubling speed*, <https://www.cnet.com/roadshow/news/audi-says-2019-a8-level-3-self-driving-tech-capable-of-doubling-speed/>, приступљено: јун 2019.
- [9] TDA2x ADAS System-on-Chip, <http://www.ti.com/lit/ml/sprt681/sprt681.pdf>, приступљено: јун 2019.
- [10] *PCI Express (PCIe)*, <https://www.lifewire.com/pci-express-pcie-2625962>, приступљено: јун 2019.
- [11] J. Mrovlje, D. Vrancic, “Distance measuring based on stereoscopic pictures”, Izola, 2008.