

Darbības programma: "Uzņēmējdarbība un inovācijas" ;
Prioritātes numurs un nosaukums: 2.1. "Zinātne un inovācijas"
Pasākuma numurs un nosaukums: 2.1.1. "Zinātne, pētniecība un attīstība"
Aktivitātes numurs un nosaukums: 2.1.1.1. aktivitāte "Atbalsts zinātnei un pētniecībai"

PROJEKTS Nr.2010/0250/2DP/2.1.1.1.0/10/APIA/VIAA/086

**Projekta nosaukums: "Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija"
(MITS)**

Līguma noslēgšanas datums: 10.12.2010.g.

Projekta sākuma datums: 01.01.2011.g.

Projekta beigu datums: 31.12.2013.g.

Eiropas Reģionālās attīstības fonda finansējuma saņēmējs: Elektronikas un datorzinātņu institūts (EDI)

ZINĀTNISKĀ PĒTĪJUMA PROGRESU APLIECINOŠĀ DOKUMENTĀCIJA

Pārskata numurs Nr.4. par periodu no 1.11.2011.g līdz 29.02.2012.g.

Projekta zinātniskais vadītājs: Leo Seļāvo, Ph.D., vad. pētnieks.

Projekta izpildītāju saraksts: Vad.pētn. Dr.sc.comp. Kārlis Freivalds
Vad.pētn. Dr.sc.comp. Mihails Broitmans
Zin.asist. Mārtiņš Liepiņš
Zin.asist. Aivars Ševerdaks
Zin.asist. Roberts Kadiķis
Progr.inž. Normunds Šilinskis

Pētnieks Uldis Grunde /Profesionāla rakstura darbs bez atlīdzības/

Zin.asist. Artis Mednis /Profesionāla rakstura darbs bez atlīdzības/

Zin.asist. Andris Gordjušins /Profesionāla rakstura darbs bez atlīdzības/

Saturs

Ievads	2
MITS	2
1 MITSSen grupa	3
1.1 Ievads	3
1.2 Īstenotie risinājumi	3
1.2.1 Transporta līdzekļ ātruma noteikšana izmantojot magnētiskos sensorus	3
1.2.2 Ciparu magnētiskie sensori	6
1.2.3 Bezvadu datu pārraides izveide	7
1.2.4 Paveiktaisi automašīnas kustības noteikšanā izmantojot mikrofonu matricu	7
1.3 Kopsavilkums un secinājumi	8
1.4 Nākotnes perspektīvas	8
2 MITSimPro grupa	9
2.1 Paveiktais	9
2.2 Īstenotie risinājumi	9
2.2.1 Algoritma īstenošana C++ kodā	9
2.2.2 Atklāšanas pazīmju klāsta paplašināšana	11
2.2.3 Adaptīvu sliekšņu iegūšana	12
2.2.4 Ātruma noteikšana izmantojot optisko plūsmu	13
2.2.5 Atklāto automašīnu attēlu saglabāšana	14
2.3 Eksperimenti un testi	14
2.4 Kopsavilkums un secinājumi	14
2.5 Nākotnes perspektīvas	15
3 MITSRegNo grupa	16
3.1 Ievads	16
3.2 Automašīna valsts numura zīmes noteikšanas tehnoloģijas apraksts	16
3.3 AVRN noteikšanas piedāvātās tehnoloģijas etapu darbības rezultātu apraksts	18
3.4 AVRN noteikšanas piedāvātās tehnoloģijas pielietošanas statistiskie rezultāti	22
3.5 Secinājumi un turpmāko darbu plāns	27

Ievads

Anotācija

”Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija” (MITS) ir ERAF atbalstīts projekts. MITS projekta mērķis ir modulāra sensoru informācijas savacēja, analīzes un komunikācijas punkta izstrāde. Šī tehnoloģija palīdz drošas un efektīvas satiksmes organizēšanā to monitorējot, komunicējot un koordinējot luksoforus atkarībā no satiksmes stāvokļa, kā arī ziņo par detektētajiem notikumiem vadības punktam.

Projekta mērķu sasniegšanai ir izveidotas sekojošas apakšgrupas: MITSView – sistēmas arhitektūra un komunikācijas; MITSSen – sensoru sistēmu pielietojumi; MITSImPro – attēlu analīze; MITSRegNo – automobiļa reģistrācijas numura noteikšana.

Projekta zinātniskajā progresa pārskatā izklāstītas zinātniskās pētniecības aktivitātes un rezultāti augšminētajās grupās. Projektu atbalsta Eiropas Reģionālās attīstības fonds, līguma Nr. 2010/0250/2DP/2.1.1.1.0/10/APIA/VIAA/086.

”Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija” (MITS)

MITS grupas galvenais pētniecības virziens ir rast inovatīvu risinājumu satiksmes drošības un efektivitātes uzlabošanai. Projekta laikā tiek izstrādāts modulārs un multifunkcionāls punkts, kas saņem dažādu sensoru datus, veic informācijas sapludināšanu un analīzi, kā arī pieņem lēmumus par potenciālas avarijas situācijas un dažādu konstatēto notikumu nodošanu vadības punktam.

MITS projektā ir sekojošas apakšgrupas saskaņā ar pētnieku individuālo pieredzi un interesēm:

- MITSView – sistēmas arhitektūra un komunikācijas;
- MITSSen – sensori viedām transporta sistēmām;
- MITSImPro – attēlu apstrāde satiksmes analīzei;
- MITSRegNo – reģistrācijas numuru automātiska atpazīšana;

Turpmākajās nodaļās ir apkopoti galvenie sasniegumi minētajos virzienos. MITS projekta 4. etapā darbi ir veikti rūpnieciskā pētījuma un eksperimentālās izstrādes aktivitātēs. Zinātniskās atskaites 1. nodaļā aprakstītie rezultāti attiecas uz abām minētajām aktivitātēm, 2. nodaļā veiktie uzdevumi attiecas uz eksperimentālās izstrādes aktivitāti, bet 3. nodaļā aprakstītie rezultāti - uz rūpniecisko pētījumu aktivitāti.

Nodaļa 1

MITSSen grupa

Anotācija

Sensoru grupa MITS projekta ceturtajā ceturksnī turpināja darbus pie magnētisko sensoru izveides, bezvadu sakaru risinājumiem, kā arī mikrofonu matricu realizācijas. Darbs tiek veikts shēmtehnisko risinājumu izveidē, programmatūras izstrādē un signālu apstrādēs jomā. Sensoru grupas veicamie darbi ietver rūpnieciskā pētījuma un eksperimentālās izstrādes aktivitātes. Pie rūpniecisko pētījumu aktivitātes attiecas magnētiskā signāla filtrācijas izpēte un eksperimentālā testa maketa izveide ciparu magnētiskajiem sensoriem, kā arī testa maketu izveide mikrofonu matricām.

1.1 Ievads

Iepriekš veikto darbu uzdevumos magnētisko sensoru jomā signāla filtrācijai tika izmantota loga filtrācija un filtrācija izmantojot spektrālo atdalīšanu (FFT-IFFT). Šajā etapā magnētiskos sensorus testēja transporta līdzekļu ātruma noteikšanai, kas ieviesa papildus prasības apstrādājamā signāla trokšņainumam. Uz brauktuves novietojamo sensoru jomā ir izveidota programmatūra msp430f16 mikrokontrolierim, lai testētu HMC 5883 trīsasu magnētisko sensoru un noteiktu maksimālo iespējamo magnētiskā lauka lasīšanas frekvenci. Bezvadu datu sūtīšanas risinājumu izstrādē veikti darbi sūtīšanas sinhronizācijas nodrošināšanai. Mikrofonu matricu izstrādē veikta ADC spiestād plates izstrāde, kā arī testa moduļa izveide datu saņemšanai uz datora.

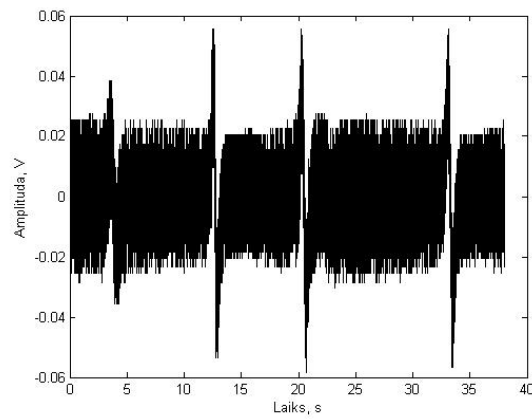
1.2 Īstenotie risinājumi

1.2.1 Transporta līdzekļ ātruma noteikšana izmantojot magnētiskos sensorus

Ir izveidots eksperimentāls makets, kurš sastāv no diviem magnētisko sensoru moduļiem (HMC1001 un HMC 1002). Maktes redzams 1.1 attēlā. Attālums starp sensoru moduļiem tika izvēlēts 0,5 m. Sensoru signāla diskretizācijas frekvence tika izvēlēta 200 Hz. Dati no sensoriem tiek savākti izmantojot NI datu savācējiem. Pie šādiem nosacījumiem transporta līdzeklis, braucot ar ātrumu 90 km/h, attālumu no viena līdz otram sensoram veiks 0,02 sekundēs, kas laika ziņā, kamēr transporta līdzeklis sasniedz otru sensoru, atbilst 4 nolsēm. Pirmajā piegājiena transporta līdzekļa ātrums tika noteikt izmantojot līmeņu metodi - abu sensoru signāla šķērsojot kādu noteiktu līmeni tiek atzīmēts laika atskaites punkts pēc kura nosaka pārvietošanās ātrumu. Eksperimentu gaitā tika noskaidrots, ka šāda pieeja nav labāks risinājums, jo zems magnētiskā lauka kropļojumu izmaiņas straujums, ko rada transporta līdzeklis, var katram sensoram būtiski atšķirtis dēļ analogo komponentu parametru izkliedes. Magnētisko sensoru signālu augšanas straujumu ietekmē arī pretējā joslā braucošie transporta līdzekļi. Tika secināts, ka, lai uzlabotu ātruma noteikšanu precizitāti, nepieciešama cita metode - maksimumu meklēšana magnētiskā sensora signālā. Salīdzinot laika starpību starp diviem sensoriem minētā veidā, analogo elementu parametru izkliede rezultātu ietekmē ievērojami mazāk. Tika izveidota programmatūra Matlab vidē iepriekš ierakstītas transporta plūsmas magnētiskā paraksta maksimumu meklēšanai. Darba gaitā tika noskaidrota, ka tikai loga filtrācijas izmantošana neļauj noteikt magnētiskā lauka sensora signālā maksimumus, jo netiek nofiltrēti visi trokšņi, kas traucē atrast maksimumu vienas svārstības robežas. 1.2 attēlā redzams magnētiskā sensora signāls x asij.

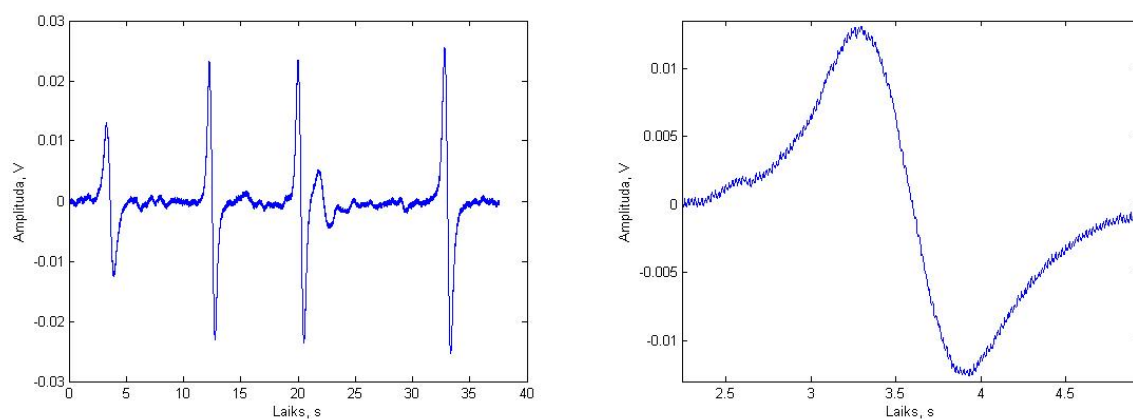


Att. 1.1: Magnētiskā sensora prototips



Att. 1.2: Magnētiskā sensora x kanāla RAW dati

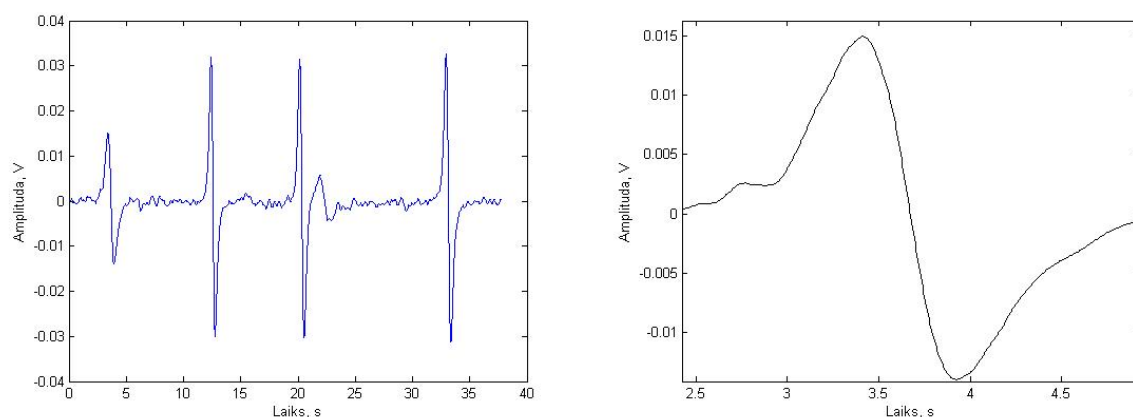
Veicot šāda signāla trokšņa filtrāciju, izmantojot loga filtru ar garumu 90 nolases, iegūst 1.3 attēlā redzamo signālu. Liela daļa no trokšņiem ir nofiltrēta, kas ļauj skaidri saskatīt un reģistrēt transporta



Att. 1.3: Ar loga filtru filtrēts magnētiskā sensora x kanāla signāls

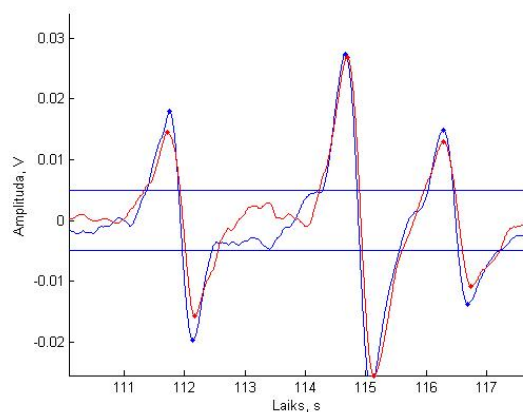
līdzekļus. Lai izmantotu maksimuma noteikšanas algoritmus, šāds signāls nav derīgs, jo ir palicis neliels troksnis, kurš redzams 1.3 attēlā. Šādam signālam nosakot maksimumus, viena izvārstījuma robežas

tie būs vairāki. Var izstrādāt sarežģītus algoritmus ar nosacījumu izpildi pie kuriem tiek noteikts maksimuma punkts, taču tas nav optimāli. Lai iegūtu signālu bez trokšņa, var izmantot papildus filtrāciju. Lai algoritmu būtu vienkārši implementēt mikrokontrolerī, var izmantot loga filtrāciju ar divām iterācijām. Pirmo iterāciju veic ar loga garumu 40 nolases, otro iterāciju ar loga garumu 20 nolases (minimālie pieļaujamie logu garumi tika noskaidroti eksperimentālā ceļā veicot signāla filtrāciju ar dažādiem logu garumiem un visuāli novērtējot iegūto rezultātu). Šāda algoritma izmantošanas rezultāts redzams 1.4 attēlā. Loga filtrācijas izmantošana ir viegli implementējama mikrokontrolerī,



Att. 1.4: Filtrēts signāls izmantojot loga filtru divās iterācijās

kā arī neprasa ievērojamus resursus, jo vienas filtrētas izejas nolases iegūšanai nepieciešams veikt divas atņemšanas, divas summēšanas darbības, kā arī divas dalīšanas darbības. Pie tam kontroliera atmiņā nepieciešams saglanāt tikai abu loga garumu nolāšu skaitu. Lai iegūtu ekvivalentu vai pat labāku rezultātu, var izmantot FFT filtrāciju, kas vienkāršo signāla līdzkomponentes noņemšanu, taču tai pat laikā ir ievērojami resursu ietilpīgākas proces. Veicot filtrāciju ar FFT - IFFT algoritmu, ieguvums trokšņu samazinājumam ir ievērojami lielāks, ja signāla realizācija ir gara (vismaz 1400 nolases). Lai šādus aprēķinus veiktu reālā laikā, ar mikrokontroliera resursiem ir par maz. Konkrētā uzdevuma veikšanai loga filtrācijas divās iterācijās izmantošana dod pilnīgi apmierinošus rezultātus. Kad signāls ir nofiltrēts, var veikt ātruma novērtējuma aprēķinu. Iepriekš tika minēts, ka etapa laikā tika apskatītas divas metodes transporta līdzekļa ātruma noteikšanai - sliekšņa metode un maksimuma metode. Sliekšņa metodes darbības princips ir sekojošs: tiek izvēlēts sliekšnis, kuru signālam sasniedzot, tiek noteikt laika moments. Abiem sensoru signāliem šis sliekšnis ir vienāds pēc amplitūdas. Laika moments, kad tiek sasniegta sliekšņa vērtība abiem sensoriem ir atšķirīgs. Ņemot vērā laika nobīdi starp laika momentiem, tiek aprēķināts transporta līdzekļa ātrums. Ir izveidota programmatūra labview vidē transporta līdzekļu ātruma noteikšanai reālā laikā. Eksperimentu gaitā tika secināts, ka šāds risinājums dod ievērojamu kļūdu. Galvenais iemesls ir magnētiskā sensora analogo elementu parametru izkliede, kas veido katram sensoram dažādu signāla amplitūdas stāvumu. Tāpat arī citi faktori, piemēram, transporta līdzeklis pretējā brauktuves joslā, ietekmē signāla augšanas straujumu. Šāds gadījums, ka viena transporta līdzekļa magnētiskajam parakstam ir dažādi augšanas strauji no dažādiem sensoriem redzams 1.5 attēlā. Kā var redzēt no grafika, jekurā no četriem punktiem, kur signāls šķērso līmeni, viena magnētiskā paraksta ietvaros attālumi starp abiem magnētiskajiem parakstiem ir atšķirīgi, lai gan teorētiski tiem jābūt vienādiem. Viens no risinājumiem ir izmantot visu četru sliekšņa šķērsošanas punktu summāro nobīdi, kas kļūdu nedaudz izlīdzinātu. Taču ņemot vērā ka signāla augšanas nevienmērīgumam ir gadījuma raksturs, šāds risinājums nav optimāls. Cits risinājums ir palielināt sliekšņa vērtību, pie kura tiek noteikts laika moments, taču tad netiks noteikta laika starpība mazākas amplitūdas transporta līdzekļu magnētiskajiem parakstiem. Pie tam kā var redzēt no 1.5 attēla magnētiskā paraksta augšanas straujuma nevienādība var būt jebkurā liknes reģionā. Mazāku blakus faktoru ietekmi uz transporta līdzekļa ātruma noteikšanu dod maksimumu meklēšana signālā. Taču arī šai metodei ir savi trūkumi, jo filtrācijas rezultātā signāla "īstais" lokālais maksimums trokšņa ietekmē var nobīdīties, taču ievērojami mazāku ietekmi rada analogi elementu parametru izkliede. Lokālo maksimumu meklēšana metodes pamatā ir izšķirt maksimums, kuri atrodas virs kāda noteikta sliekšņa. Maksimumu meklēšanai ir izveidots eksperimentāls programnodrošinājums Matlab vidē, kurš veic maksimumu meklēšanu iepriekš ierakstītam signālam (šis uzdevums netiek veikts reālā laikā). Transporta līdzekļa magnētiskā paraksta realizācija ar

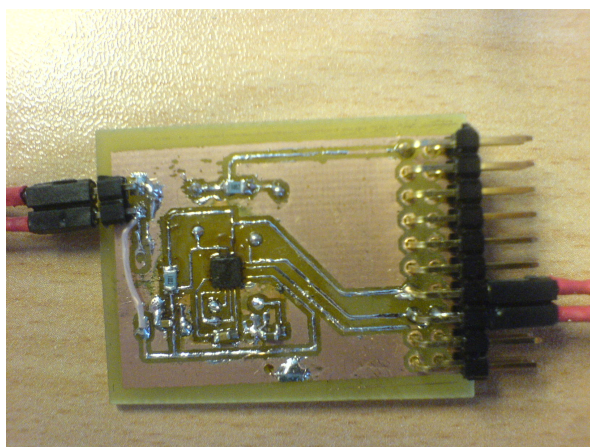


Att. 1.5: Magnētiskā sensora signāla sliekšņošana

atrastiem maksimumiem redzama 1.5 attēlā. Izmantojot atrastos maksimumus eksperimentāli noteikts transporta līdzekļa ātrums, kura precizitāte ir augstāka kā sliekšņa gadījumā. Transporta līdzekļu ātruma noteikšana izmantojot brauktuves malā novietotus magnētiskos sensorus dod ievērojamu kļūdu ātruma noteikšanā. Lai uzlabotu ātruma noteikšanas precizitāti, nepieciešams palielināt signāla diskretizācijas frekvenci, ko savukārt ierobežo bezvadu datu pārraides nosacījumi.

1.2.2 Ciparu magnētiskie sensori

Projekta iepriekšējo etapu laikā tika testēts trīsasu ciparu magnētiskā lauka sensori HMC5843. Eksperimentu laikā tika noskaidrots, ka konkrēto sensoru lasīšanas frekvence nav pietiekama, lai nodrošinātu drošu transporta līdzekļu detektēšanu. Projekta ceturtajā etapā tika iepirkti jauni ciparu magnētiskie sensori HMC 5883, kuriem signāla lasīšanas frekvence nepārtrauktā režīmā sasniedz 75 Hz, bet viena nolasījuma režīmā 160 Hz. Iepriekš izveidotā programmatūra tika pielāgota HMC5883 sensoram. Darba mērķis bija noskaidrot maksimālo iespējamo datu lasīšanas frekvenci. HMC5883 sensori tika iepirkti kā čipi, tādēļ vispirms tika izveidot PCB (1.6 attēls). Izveidotais risinājums ir ievērojami lielāks pēc izmēriem par maksimāli mazāko. Šāds risinājums izvēlēts tādēļ, jo mērķis



Att. 1.6: HMC 5883 PCB plates prototips

šiem sensoriem ir izveidot bezvadu datu pārraidi. Gala risinājumā ir paredzēts izmantot NRF9E5 raiduztvērēju, gan datu sūtīšanai, gan sensora konfigurēšanai. Gatavi bezvadu datu pārraides moduļi ir izstrādāti jau iepriekš, tādēļ laika taupīšanas nolūkā, lai nebūtu jāizstrādā jaunas PCB plate, tiek izmantoti jau esošie moduļi. Testa nolūkos datu saņemšanai un pārsūtīšanai uz datoru tika izmantots MSP430f1611 mikrokontrolieris. HMC5883 sensora datu sūtīšana tiek nodrošināta izmantojot I2C

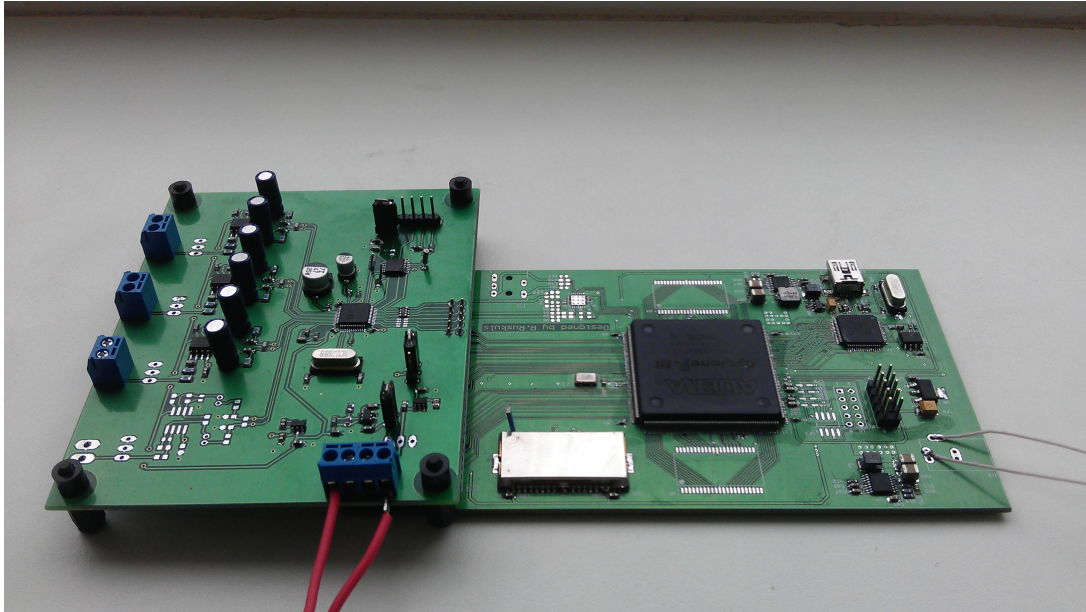
datu parraidi. MSP430f1611 mikrokontrolierim ir nodrošināts atbalsts I2C perifērijas iekārtām, kas ievērojami atvieglo sazināšanos ar sensoru. Dati uz datoru tiek pārsūtīti izmantojot UART (testa risinājumā tiek izmantots iepriekš izveidots modulis ar FTDI 2322 mikrosihēmu). Viena lasījuma režīmā HMC5883 sensors ziņo par jaunu datu ierakstīšanu reģistros noraustot RDY izejas kāju vai arī ierakstot attiecīgu vērtību reģistrā. Konkrētā risinājumā tika izmantota datu ierakstes reģistra lasīšana. Datu sūtīšanai caur UART tika uzstādīts standarta datu pārraides ātrums 9500 biti sekunde. Vēicot eksperimentus tika konstatēts, ka maksimālā iespējama magnētiskā sensora datu nolasīšanas frekvence ir 120 Hz, lai gan datu gatvības reģistrs uzstādījās ievērojami agrāk. Tika secināts, ka lielu laiku no datu lasīšanas/sūtīšanas procesa aizņem datu pārsūtīšanu caur UART datu kopni, tādēļ tika pārveidota mikrokontroliera programmatūra datu sūtīšanai uz maksimālo iespējamo datu pārraides ātrumu, kuru nodrošina MSP430f1611 mikrokontrolieris - 115200 biti sekundē. Izmantojot šādu risinājumu iespējamā datu lasīšanas frekvence palielinājās līdz 200 Hz. Ņemot vērā, ka dati uz datoru tiek sūtīti izmantojot ASCII simbolus, datu pārraides laiku var samazināt, sūtot RAW datus heksadeciālā formā (pārveidojot datus uz ASCII formu sūtāmo simbolu skaits dubultojas). Vēicot šādu vienkāršojumu tika iegūts maksimālā stabilā datu lasīšanas frekvence 230 Hz. Tika secināts, ka optimāli veidot nelielu aizturi starp datu lasīšanas seansi, un tikai tad pārbaudīt datu gatvības reģistra stāvokli. Neveidojot aizturi, var veidoties situācija, ka datu gatvības reģistrs tiek pārbaudīts divas vai trīs reizes (tik ilgi līdz magnētiskā lauka dati ierakstīti datu reģistros), kas pazemina datu lasīšanas frekvenci. Ieviešot nelielu aizturi starp datu lasīšanas seansi, tiek nodrošināta stabila lasīšanas frekvence. Datu plūsmas laika samazināšanai tika arī mēģināts izmantot I2C protokola darbību 400 kHz frekvencē, taču konkrētais PCB risinājums nespēja nodrošināt atbilstošas prasības galvenokārt dēļ dažādiem pārejas procesiem ķēdēs, kas noveda pie programmatūras "uzkāšanās" datu saņemšanas režīmā.

1.2.3 Bezvadu datu pārraides izveide

Turpināts darbs pie bezvadu sensoru tīkla izveides magnētisko sensoru datu pārsūtīšanai uz datoru izmantojot Nordic Semi nRF9E5 raiduztvērēju 433MHz frekvenču joslā. Datu pārraides realizācijai Keil uvision vidēt C valodā tika izveidots sinhronizācijas algoritms priekš nRF9E5 raiduztvērēja, kurš nodrošina takts impulsa pārsūtīšanu, kā arī pakešu sūtīšanas laika dali. Tīkla pamatā ir četri sensoru moduļu raiduztvērēji un viens savācējkārtas modulis. Savācējkārtas modulis sastāv no raiduztvērēja nRF9E5 un USB UART pārveidotāja. Savācējkārtas modulis sūta sensoru moduļiem takts impulsus, kuri sinhronizē kopējo sistēmu un nodrošina visu sensoru datu vienlaicīgu nolasīšanu, kas ir svarīgi tālākā datu apstrādē. Sākuma stāvoklis visiem sensoriem ir vienāds, tie atrodas uztveršanas režīmā un gaida sinhronizācijas datu paketes pienākšanu, kuru sūta savācējkārta. Pēc tā saņemšanas notiek visu sensoru ADC vienlaicīga analoģo signālu diskretizācija no magnētiskajiem sensoriem, savukārt savācējkārta pārslēdzas uztveršanas režīmā un gaida datu paketes no sensoru moduļiem. Pēc sensoru datu nolasīšanas tiek veikta datu pakešu formēšana un identifikācijas numura pievienošana datu paketei. Savācējkārtā datu pakas tiek identificētas pēc katra sensoram piešķirtā identifikācijas numura. Gadījumā, ja tiek pazaudēta kāda no sensoru sūtītām pakām, savācējkārta to atpazīst un sūtīto paku secība tiek saglabāta aizpildot iztrūkstošo paku ar nulles datiem. Uz doto brīdi bezvadu sistēma tika notestēta ar diviem magnētiskiem sensoriem, kuriem pieslēgti raiduztvērēji. Takts impulsi tiek sūtīti ar 185Hz atkārtotās frekvenci, kas uz doto brīdi ir maksimāli sasniegtais. Takts impulsu sūtīšanas biežumu ierobežo raiduztvērēju ātrdarbība un bezvadu datu pārraides ātrums.

1.2.4 Paveiktais automašīnas kustības noteikšanā izmantojot mikrofonu matricu

Turpinot iepriekšējā periodā paveikto tika izstrādāta savācējkārtas analoģā ciparu pārveidotāja iespietplate ar vadības mikrokontrolieri skaņas signālu ierakstīšanai (1.7 attēls). Visa savācējkārta sastāv no vadības mikrokontroliera MSP430F2013, AD1974 četru kanālu analoģa- ciparu pārveidotāja, kurš pieslēgts pie Altera Cyclone III FPGA izmantojot I2S un FT232 USB UART pārveidotāja. Ar MSP mikrokontrolieri tiek veikta AD1974 iekšējo reģistru konfigurēšana. Altera pārsūta datus uz FT232 USB UART pārveidotāju 8 bitu paralēlā režīmā. Šī savācējkārta bija nepieciešama, jo iepriekš izmantotā savācējkārta NI-6009 nenodrošināja nepieciešamo analoģo signālu diskretizācijas frekvenci. Izmantotais analoģais ciparu pārveidotājs ļauj mainīt diskretizācijas frekvenci robežās no 24kHz - 192kHz. Ar izveidoto sistēmu tika veikti testa ieraksti ar vienu pieslēgtu mems WM7110 mikrofonu ar mērķi pārbaudīt sistēmas darbību. Ieraksta testi tika veikti laboratorijas apstākļos. Tiek turpināts darbs pie programmatūras izstrādes, lai varētu pārsūtīt visu trīs mikrofonu signālus uz datoru caur USB UART



Att. 1.7: Mikrofonu matricas signālu savācējiekārta, pa kreisi izstrādātā iespiedplate ar AD1974 analogo ciparu pārveidotāju

pārveidotāju.

1.3 Kopsavilkums un secinājumi

Etaļa laikā ir noskaidrota transporta līdzekļu ātruma noteikšanas metodika, kā arī izveidoti principi pēc kuriem iespējams veikt ātru magnētiskā sensora signāla filtrāciju ar iespēju risinājumu implementēt mikrokontrolerī. Veikti eksperimenti ar brauktuves malā novietotiem magnētiskiem sensoriem, kā arī izveidots testa risinājums HMC5883 sensoram. Izveidots testa risinājums mikrofonu matricām datu sūtīšanai uz datoru, izmantojot I2S datu kopni, kā ar izveidota un salodēta PCB signāla diskretizācijai un pastiprināšanai. Izveidots risinājums bezvadu datu pārraides sinhronizācijai vairāku sensoru gadījumā.

1.4 Nākotnes perspektīvas

Turpmākiem darbiem nepieciešams izveidot HMC 5883 bezvadu datu sūtīšanas risinājuma programmnodrošinājumu, kas ietver arī sensora aktivizāciju, izmantojot NRF9E5 raiduztvērēju. Jāstrādā pie brauktuves malā novietojamo magnētisko sensoru datu implementācijas mikrokontrolerī un automatizētas transporta līdzekļa ātruma noteikšanas programmatūras, kā arī automātiskās kalibrācijas nodrošināšanas. Bezvadu datu sūtīšanas jomā nepieciešams veikt testus reālos apstākļos un palielināt takts impulsu sūtīšanas frekvenci, kā arī nodrošināt 4 sensoru datu pārsūtīšanu, kā arī praktiski nomērīt datu pārraides attālumu.

Nodaļa 2

MITSimPro grupa

Anotācija

Grupa MITSimPro turpina izstrādāt video apstrādes sistēmu automašīnu atklāšanai. Ceturtajā gada ceturksnī izstrādātais algoritms, kas iepriekš tika īstenots un testēts MATLAB Simulink vidē, tika uzrakstīts objektorientētā C++ kodā. Konkrētā programmēšanas valoda ir arī paredzētā valoda gala iekārtai. Algoritma atsevišķie posmi tika uzlaboti, piemēram, pie mašīnu atklāšanas ņemot vērā jaunas pazīmes, kuru atrašanai pielietoti adaptīvi sliekšņi, kā arī piešķirot atsevišķajām pazīmēm svarus, kas nosaka, vai konkrētā pazīme izteikti, vidēji vai vāji norāda uz automašīnas esamību.

Kodā iestrādāta iespēja saņemt ieejas datus no tīkla kameras tiešsaistes režīmā, kā arī šajā režīmā veikti pirmie testi.

2.1 Paveiktais

Šajā ceturksnī algoritma īstenošana, kā arī uzlabošana notika C++ kodā, izmantojot atvērtā koda bibliotēku OpenCV:

- Iepriekš Simulink vidē īstenotais algoritms pilnībā pārrakstīts uz C++ valodu (nodaļa 2.2.1).
- Izstrādes vajadzībām uzlabota programmas darbības vizualizācija (nodaļa 2.2.1).
- Pētīts un paplašināts pazīmju klāsts, kas nosaka, vai tiek detektēta automašīna (nodaļa 2.2.2).
- Dažādās pazīmes apvienotas mašīnām atbilstošo intervālu slēgšanai (nodaļa 2.2.2).
- Iestrādāta automātiska adaptīvu sliekšņu noteikšana katrai pazīmei (nodaļa 2.2.3).
- Īstenota alternatīva metode aptuvenai automašīnas ātruma noteikšanai, izmantojot optiskās plūsmas noteikšanu (nodaļa 2.2.4).
- Iestrādāta atklāto automašīnu attēlu saglabāšana uz cietā diska (nodaļa 2.2.5).
- Algoritma darbināšana tiešsaistes režīmā ar tīkla kameru (nodaļa 2.3).

2.2 Īstenotie risinājumi

2.2.1 Algoritma īstenošana C++ kodā

Iepriekšējā ceturksņa atskaitē tika piedāvāts pret skaitļošanas jaudām maz prasīgs algoritms automašīnu atklāšanai un skaitīšanai izmantojot video plūsmu. Tas tika īstenots MATLAB Simulink vidē, kā arī testēts uz iepriekš ierakstītiem satiksmes video failiem. Salīdzinājums ar iepriekš apskatītajām metodēm, kas apstrādāja visu kadru, parādīja, ka algoritms ir perspektīvs risinājums, tāpēc rūpnieciskā pētījuma rezultātā turpmākai izlabošanai izvēlēts algoritms, kas apstrādā nelielu kadra daļu – ceļa virzienam perpendikulāru līniju.

Šajā gada ceturksnī piedāvātais algoritms tika pārnests uz C++ valodas kodu. Tas ne tikai ļauj veikt testus tiešsaistes režīmā, jo uz C++ īstenots algoritms kļūst ātrāks nekā darbinot to kopā ar MATLAB programmu, bet arī ir solis uz priekšu intelligentā transporta sistēmas mezgla izstrādē, kura programmatūras daļu paredzēts veidot tieši C++ valodā. Īstenojot algoritmu tika lietota atvērtā koda bibliotēka OpenCV, kurā iestrādātas biežāk lietojamās attēlu apstrādes funkcijas, piemēram, zemāk apskatītā Otsu metode automātiskai sliekšņu atrašanai, atrastā sliekšņa pielietošanas funkcija, krāsu telpas konvertēšanas funkcija u.c.

Algoritms tika rakstīts objektorientētā kodā, cenšoties to padarīt ērti lasāmu un korigējamu. Tika ieviestas vairākas klases. *VideoProcessor* klase satur pētāmās līnijas (vai vairāku līniju) koordinātes. Klases metode *openCamera* nodrošina video ievadi no datora cietā diska vai kameras. Šajā klasē tiek veidoti kadru klases *Frame* objekti, kā arī izsauktas to metodes. Šie objekti ir kadri, un to metodes ir dažādas operācijas ar kadriem, piemēram, kadru konvertēšana no krāsainiem uz melnbaltiem, interesējošā apgabala izdalīšana no kadriem. Kadru objekts tiek lietots arī kameras noraustīšanās fiksēšanai.

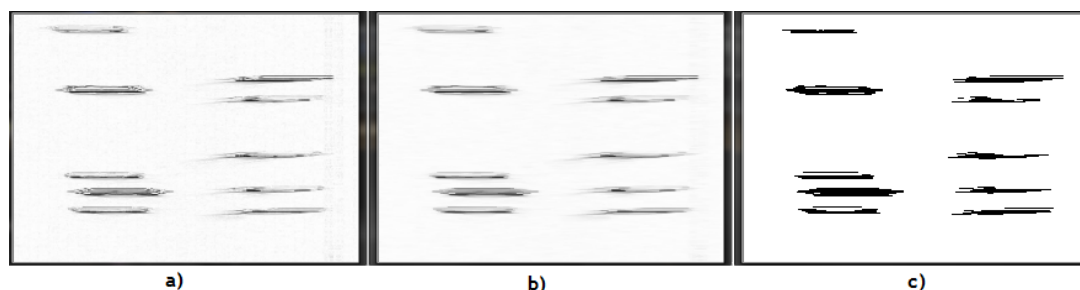
VideoProcessor klase satur metodi *run*, kur ciklā tiek apstrādāts katrs pienākošais kadrs. Vispirms tiek apstrādāti *Frame* objekti, līdz tiek iegūta divu melnbaltu blakus kadru starpības absolūtās vērtības. Tad ciklā tiek izveidoti citas klases - *FewLines* – objekti. Šie objekti iekļauj vairākas blakus esošas paralēlas līnijas (kadra rindas), kuras tiek pētītas – klases objektu var būtībā uzskatīt par detektoru, kas pētāmajā apgabalā detektēs mašīnu esamību. *FewLines* objekts iekļauj tādas konstantes, kā, piemēram, minimālo detektējamo automašīnu platumu, lukturu aptuveno izmēru intervālu u.c.

VideoProcessor ciklā tiek izsauktas *FewLines* metodes, kur nozīmīgākā ir *processLine* metode. Šajā metodē tiek apstrādāti *Line* un *Interval* klašu objekti. *Line* objekti ir attēli (matricas) ar vienu rindu. Šie attēli var būt oriģinālā kadra intensitātes pikseli uz pētāmās līnijas, krāsu komponentes, robežas, blakus kadru starpības uz līnijas vai citi parametri. Dažādi parametri var papildināt viens otru, automašīnu atklāšanas procesā, kas sīkāk parādīts nodaļā 2.2.2. Katra parametra līnija ir tādā atsevišķs *Line* klases objekts. *FewLines* klasē izsauktās *Line* metodes apstrādā dažādo parametru līnijas, piemēram, veic gludinošo filtrāciju, uzkrāj līnijai atbilstošu, adaptīvo, tukšam ceļam atbilstošu fona līniju (viens no *Lines* mainīgajiem), atņem parametru līniju no atbilstošā fona, atrod starpības absolūto vērtību, atrod adaptīvu sliekšni, pielieto to līnijas un fona starpībai u.c.

Viens no izmantotajiem parametriem automašīnu atklāšanai ir blakus kadru starpība. Pēc šīs starpības līnijas apstrādes klasē *FewLines* tiek izsaukta metode *findIntervals*, kas jau sliekšņotajā *Line* objektā atrod savienotus intervālus - *Interval* klases objektus. Intervāli sastāv no to sākuma un beigu koordinātēm uz līnijas, no to pastāvēšanas ilguma, kā arī to klasē ir vairāki citi mainīgie, kas rāda, piemēram, cik intensitātes līnijas pikseli ir pārvarējuši sliekšni konkrētā intervāla apgabalā. Atrastie intervāli tiek saglabāti vektorā, kur jaunus intervālus salīdzina ar iepriekš atrastajiem un apvieno tos, ja tie ir vienas un tās pašas automašīnas intervāli.

Interval klases objekti tālāk tiek apstrādāti *FewLines* metodē *processLine* katrā kadrā pēc tam, kad veikta nepieciešamā apstrāde ar visām parametru līnijām. Visi šie parametri nosaka, vai konkrētu intervālu šajā kadrā būtu jāslēdz, jo mašīna ir pilnībā izbraukusi cauri līnijai. Katram intervālam tiek secīgi saskaitīti sliekšņoto līniju (dažādo parametru līniju) sliekšni pārvarējušie pikseli, tad tiek izsaukta *Interval* klases metode, kas apvieno informāciju no dažādu parametru līnijām un pieņem galīgo lēmumu par intervāla slēgšanu. *Interval* klasē ietilpst arī metodes, kas nosaka aptuveno automašīnas ātrumu (šeit gan metodes ieejas parametram jābūt kādam lielākam kadra apgabalam nekā tikai viena līnija), kā arī metode atklātās automašīnas bildes saglabāšanai uz cietā diska un vairākas citas.

Vēl viena klase algoritma kodā, ir *Visualization* klase algoritma darbības vizualizēšanai. Šajā klasē iestrādātās metodes ļauj izvadīt *Line* klases objekta izmaiņas laikā, tādējādi, piemēram, var vizuāli salīdzināt līnijas izskatu pirms gludinošās filtrēšanas, pēc tās, pēc sliekšņu pielietošanas (attiecīgie invertētie attēli 2.1.a, 2.1.b, 2.1.c), kā arī pēc jebkuras citas modifikācijas. Attēlos vertikālais virziens atbilst laikam - pirmā rinda parāda līnijas izskatu šī brīža kadrā, otrā rinda rāda izskatu vienu kadru iepriekš, pēdējā rinda atbilst izskatam pirms 200 kadriem.



Att. 2.1: Vizualizācijas klases rezultātu izvade a) blakus kadru līniju starpības izmaiņas laikā; b) starpība pēc gludinošās filtrēšanas; c) rezultāts pēc sliekšņošanas

Vizualizācijas klase ļauj arī iegūt uzskatāmu rezultāta izvadišanu – uz krāsainā ieejas video kadra tiek

uzzīmēta pētāmā līnija (vai vairākas), uz kuras savukārt tiek zīmēti šobrīd detektētie intervāli. Virs katra intervāla tiek izvadīts aptuvenais intervālam atbilstošās automašīnas ātrums (nobraukto pikseļu skaits kadrā), zem intervāla tiek parādīts uz kādu parametru rēķina (intensitātes, robežas, krāsas) intervāls vēl netiek slēgts.

2.2.2 Atklāšanas pazīmju klāsta paplašināšana

Nodaļā 2.2.1 jau pieminēts, ka intervālu slēgšanas nosacījumus veido vairāki parametri. Pirmais parametrs ir pietiekami zīmīga atšķirība uz līnijas blakus kadrus. Algoritmā tika arī paredzēts novērtēt tukšam ceļam neatbilstošu robežu esamību šobrīd eksistējošā intervāla apgabalā. Tāpat intervāls netika slēgts, ja tas pēdējo 3 kadru laikā ir mainījis savu izmēru, vai tā apgabalā ir bijis atklāts kāds jauns, pietiekami plats intervāls. Šajā gada ceturksnī algoritms papildināts ar jauniem parametriem.

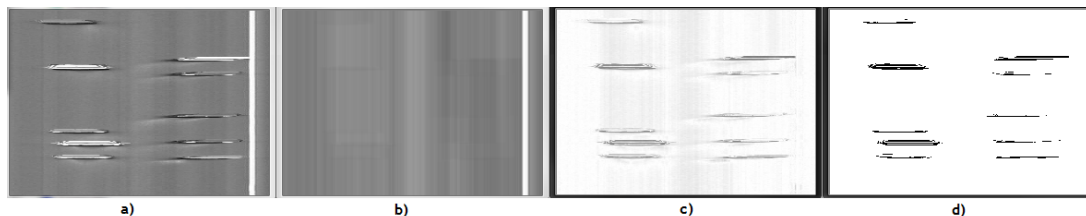
Par mašīnu var liecināt melnbalto pikseļu intensitātes atšķirība no tukša ceļa pikseļu intensitātes. Tātad ir jāiegūst tukša ceļa intensitātes līnija, un mainīgo āra apstākļu dēļ, tai jābūt adaptīvai. Fona iegūšana realizēta tādējādi, ka laikā, kamēr uz līnijas nav atrasts kāds intervāls, fons ir vienāds ar šī brīža līniju. Pretējā gadījumā, katram fona pikselim tiek nedaudz modificēta vērtība pēc formulas 2.1.

$$F_{i,k} = (1 - a_{i,k}) \cdot (0.98 \cdot F_{i,k-1} + 0.02 \cdot P_{i,k}) + a_{i,k} \cdot F_{i,k-1}, \quad (2.1)$$

kur $a=0$, ja pikselis i kadrā k atrodas šobrīd eksistējošā intervāla apgabalā, un $a=1$, ja pikselis atrodas ārpus intervāliem. P ir šī brīža (kadra k) līnijas pikseļa vērtība, F – uzkrātā atbilstošā fona pikseļa vērtība. Vienādojums 2.1 rāda, ka pat, ja ir atrasts un balstoties uz detektēšanas parametriem, tiek uzturēts kāds intervāls, tā apgabala fons tiek lēnām modificēts, ņemot vērā jaunās vērtības. Šāda pieeja nodrošina, ka, ja gadījumā fons ir strauji un zīmīgi izmainījies, bet šīs izmaiņas novedušas pie viltus intervāla detektēšanas un uzturēšanas, tad laika gaitā tomēr tukša ceļa attēls pielāgosies jaunajiem apstākļiem un viltus intervāls tiks slēgts. Attēlā 2.2.a parādīta intensitātes līnijas izmaiņa 200 kadru laikā, savukārt 2.2.b parādīta atbilstošā fona izmaiņa šajā laikā.

Kad iegūts fona attēls, tas tiek atņemts no šī brīža intensitātes līnijas un starpībai ņemta absolūtā vērtība – invertēts rezultāts parādīts 2.2.c. Tad starpības līnijai tiek noteikts un pielietots adaptīvs sliekšnis, kas piešķir visiem pikseliem tikai divas no iespējamām vērtībām – 0 vai 1. Sliekšņotas intensitātes līnijas izmaiņas laikā redzamas 2.2.d.

Katram atrastajam intervālam tiek atrasta to apgabalā esošo pikseļu summa, kas tātad būs lielāka, ja vairāki pikseļi šajā apgabalā pārvarējuši sliekšni un ieguvuši vērtību 1.

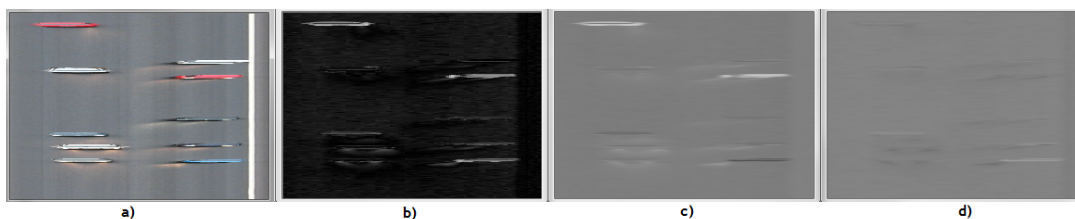


Att. 2.2: Pikseļu intensitātes izmantošana par intervālu slēgšanas parametru

Līdzīgā veidā kā melnbalto pikseļu intensitāte, tika pētīta iespēja lietot krāsu kā papildus parametru. Tika apskatīts dažādu krāsu telpu pielietojums. Piemēram, pārveidojot RGB krāsu telpas attēlu, kurā attēls sastāv no 3 divdimensiju matricām – R, G un B, par HSV (krāsātspozmes) telpas attēlu, kurš tāpat sastāv no 3 komponentēm, iespējams izdalīt krāsu piesātinājuma komponenti S. Attēlā 2.3.b parādīta 2.3.a attēlam atbilstošā krāsu piesātinājuma izmaiņa laikā.

Cita krāsu telpa YCrCb ļauj izdalīt sarkanās Cr un zilās Cb krāsas objektus, kas parādīts attēlos 2.3.c un 2.3.d. Salīdzinot ar piesātinājumu, redzams, ka krāsu komponentes ir mazāk trokšņainas, tāpēc YCrCb telpas izmantošanai dota priekšroka salīdzinājumā ar HSV. Izdalītajām Cr un Cb līnijām tiek pielietotas tās pašas metodes, kas intensitātes līnijai, līdz brīdim, kad iegūtas līnijas un fona starpības absolūtās vērtības. Tad šīs līnijas apvieno summējot attiecīgos pikselus. Sliekšnis tiek meklēts un pielietots jau lietojot apvienoto līniju, kas gala rezultātā dos vienu intervāla slēgšanas parametru no fonam nepiederošu krāsu izpētes.

Intervālu slēgšanas nosacījumiem, kas iegūti ar fona līnijas uzkrāšanas paņēmieniem, piemīt trūkums. Ja fona bildes uzkrāšanā ieviesusies kļūda, piemēram, kā tukšs fons tika saglabāta automašīnas daļa, tad



Att. 2.3: Līnijas krāsu komponentu izmaiņas laikā

pēc mašīnas aizbraukšanas šis intervāls netiks uzreiz slēgs, jo šis slēgšanas nosacījums to neļaus. Kā jau iepriekš minēts, ar laiku fons atjaunosies līdz īstajai vērtībai, bet šis laiks izvēlēts pietiekami ilgs, lai uz detektējošās līnijas uz laiku apstājusies mašīna nepaspētu kļūt par fona sastāvdaļu. Šobrīd fona adaptācijas laiks piemērlēts eksperimentāli, bet perspektīvā to iespējams pielāgot, piemēram, luksofora sarkanās gaismas degšanas ilgumam (kas ITS gadījumā varētu būt mainīgs).

Lai viena kļūdaina parametra dēļ nebūtu jāgaida uz fona atjaunošanos intervāla slēgšanai, kas noved pie atklāšanas kļūdām, slēgšanas nosacījumiem, kuros izmantota fona uzkrāšana, tiek piešķirti svāra koeficienti. Piešķiršana ir atkarīga no tā, cik izteikti šis nosacījums norāda uz automašīnas esamību intervālā. Piemēram, ja sliekšņotā krāsu komponentu līnijā (CrCb) intervāla apgabalā vairāk kā trešdaļa no pikseliem ir pārsnieguši sliekšni, šim nosacījumam tiek piešķirts svārs 3. Ja mazāk kā divi pikseli pārsnieguši sliekšni – svārs ir 0, pārējos gadījumos – 2.

Šobrīd trīs nosacījumi algoritmā (robežas, krāsu komponentes un intensitāte) izmanto fona uzkrāšanu, tādēļ tiem tiek piešķirti svāra koeficienti. Gala nosacījums, kas nosaka, vai apskatītie parametri norāda uz mašīnas esamību intervālā, iegūstami saskaitot dažādo parametru svāra koeficientus. Ja summa pārsniedz 4, tas nozīmē, ka vismaz diviem parametriem ir vidējs svāra koeficients. Ja tikai vienam no trīs parametriem ir izteikts svāra koeficients, bet pārējiem tie ir 0, tas var liecināt par kļūdu šī parametra fona ieguvē. Šajā gadījumā svāru summa nepārsniedz 4 un gala nosacījums būs par intervāla slēgšanu.

2.2.3 Adaptīvu sliekšņu iegūšana

Gan intensitāšu, gan robežu, gan krāsu komponentu un blakus kadru intensitāšu starpību līniju apstrādē nepieciešams pielietot sliekšņus, kas noteiks, kuri pikseli ir ņemami vērā pie intervālu slēgšanas. Sliekšņa vērtībai jābūt tādai, lai attēlā, kurā redzama, piemēram, šī brīža intensitātes līnijas starpība ar fonu, pēc iespējas precīzāk tika izdalīti objekti (automašīnas, kam vajadzētu būt ar lielāku pikseļu intensitāti) no jaunā fona (kura pikseliem ideālā gadījumā būtu nulles vērtības). Jau iepriekšējos projektā veiktajos testos tika secināts, ka optimālās sliekšņu vērtības ir atkarīgas no kadrā redzamās scēnas apstākļiem – no apgaismojuma, no ceļa seguma stāvokļa u.c. Tādēļ sliekšņa atrašanai jābūt automatiskai un sliekšņa vērtībai būtu jāpielāgojas šī brīža apstākļiem.

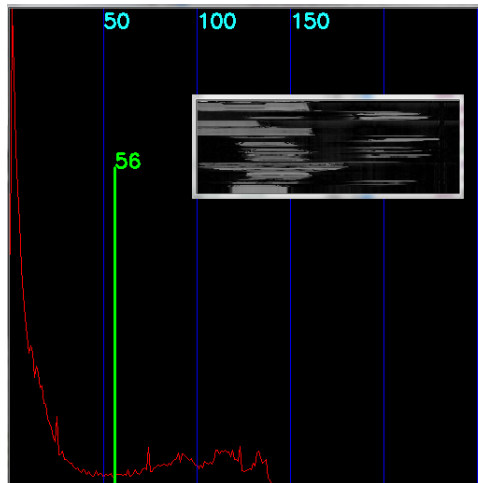
Automatiskai sliekšņu atrašanai tika apskatīta Otsu metode. Metode izmanto attēla histogrammu, lai sadalītu attēlu priekšplānā un fona pikseļos. Šim nolūkam histogramma tiek sadalīta divās daļās, iteratīvi mainot iespējamās sliekšņa vērtības. Gan priekšplānam, gan fonam tiek atrastas pikseļu intensitātes dispersijas σ_p^2 un σ_f^2 , kā arī svāri W_p un W_f , kas priekšplānam ir tajā ietilpstošo pikseļu skaita attiecība pret kopējo attēla pikseļu skaitu – līdzīgi iegūstams fona svārs. Katrai sliekšņa vērtībai tiek iegūta kopējā dispersija pēc formulas 2.2.

$$\sigma_k^2 = W_p \cdot \sigma_p^2 + W_f \cdot \sigma_f^2 \quad (2.2)$$

Tā iteratīvi piemeklētā sliekšņa vērtība, kurai kopējā dispersija ir vismazākā, arī ir ar Otsu metodi iegūtais rezultāts.

Tā kā mūsu algoritma gadījumā sliekšnis tiek pielietots vienai līnijai, kurā var gan būt, gan nebūt priekšplāna objekti, Otsu metodes pielietošana nepieciešams iegūt attēlu, kurā noteikti atrodas priekšplāna objekti. Šim nolūkam, tās līnijas, uz kurām ir tikuši detektēti intervāli, kas savukārt norāda, ka tajās visticamāk ir interesējošie objekti, tiek saglabātas vektorā. No buferī saglabātajām līnijām tiek izveidots attēls, kam pielietota Otsu metode. Attēlā 2.4.a redzams no vektorā saglabātajām intensitātes līnijām izveidotais attēls, kā arī šī attēla histogramma, kurā iezīmēta automatiski atrastā sliekšņa ar vērtību 56 atrašanās vieta.

Eksperimentos secināts, ka Otsu metodes pielietošana atrod labu sliekšni robežu pazīmju izdalīšanai, tomēr intensitātes un krāsu komponentu gadījumos atrastais sliekšnis ne vienmēr bija optimāls. Tādēļ



Att. 2.4: Otsu metodes pielietošana uzkrātam līniju attēlam, šī attēla histogramma un atrastais sliekšnis

šiem gadījumiem sliekšņa atrašana tika papildināta ar papildus nosacījumiem. Šeit Otsu metode tiek pielietota, kad attēla uzkrāšanas vektorā tiek ierakstīta jauna līnija, un, kad izpildās nosacījums, ka šobrīd uz līnijas esošu intervālu apgabalos ir pietiekams skaits sliekšni pārvērojušu pikseļu. Citos gadījumos sliekšnis tiek pielāgots iegūtajiem rezultātiem – ja ārpus intervāliem ir vairāk sliekšni pārvērojušu pikseļu nekā intervālu apgabalos vai ārpus intervāliem šādi pikseļi aizņem vairāk kā noteiktu daļu no ārpus intervālu kopējā apgabala, tad sliekšnis tiek katrā kadrā palielināts par 1. Savukārt, ja ārpus intervāliem ir maz sliekšni pārvērojušu pikseļu, kā arī intervālu apgabalos to ir maz, sliekšnis tiek samazināts par 1.

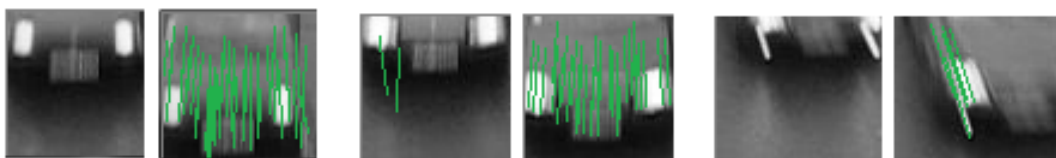
2.2.4 Ātruma noteikšana izmantojot optisko plūsmu

Iepriekš algoritmā jau tika paredzēta un īstenota iespēja noteikt automašīnas aptuveno ātrumu kadra plaknē, salīdzinot mašīnai atbilstošo līniju ar citām tuvām līnijām nākamajā kadrā. Tagad algoritms tika papildināts ar alternatīvu metodi ātruma noteikšanai, kas būtu prasīgāka pret skaitļošanas jaudām, tomēr varētu izrādīties arī precīzāka.

Līdzīgi kā iepriekšējā metodē, tiek pētīts neliels kadra apgabals. Konkrētais intervāls, kura detektētajai automašīnai tiek noteikts ātrums, ir šī pētāmā taisnstūrainā apgabala augšējā mala. Uz leju taisnstūris iekļauj vairākas kadra rindas, kuru daudzumu nosaka automašīnu ātrums, video kameras pozīcija pret ceļu, kā arī detektējošās līnijas novietojums kadrā. Jo ātrāk mašīna pārvietojas kadra plaknē, jo garāks apgabals jāapstrādā, lai būtu iespējams divos sekojošos kadrus šajā apgabalā atrast vienu un to pašu mašīnas pazīmi.

Jaunā metode, izmantojot OpenCV iestrādātu funkciju – *cv::goodFeaturesToTrack*, atrod attēlā zīmīgas pazīmes, konkrētajā gadījumā tiek noteikti stūri attēlā. Iespējams norādīt, cik ir maksimālais atrodamo stūru skaits, cik izteiktiem tiem ir jābūt. Tad tiek pielietota optiskās plūsmas aprēķina funkcija *calcOpticalFlowPyrLK*, kas atrod zīmīgo pazīmju pārvietojumu sekojošos kadrus.

Atšķirībā no iepriekšējās metodes, optiskās plūsmas analīze parāda ne tikai pārvietošanās ātrumu, bet arī virzienu. Gadījumos kad automašīnas kadrā pārvietojas uz leju, tas nav tik aktuāli, tomēr šādā gadījumā optiskā plūsma ļauj novērtēt kadrā redzamo perspektīvu.



Att. 2.5: Zīmīgu parametru izsekošana ar optiskās plūsmas aprēķinu

Attēlos 2.5 parādīti piemēri - viens otram sekojošu kadru apgabali, kuros tika meklēta optiskā plūsma. Ar zaļām līnijām atzīmētas izsekotās pazīmju trajektorijas. Par mašīnas ātrumu, jeb kadra

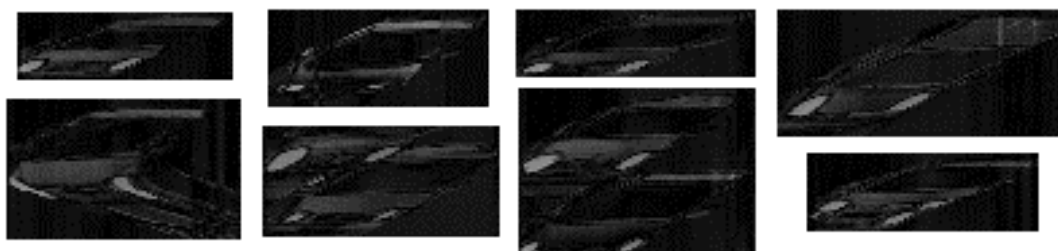
laikā nobraukto pikseļu skaitu var uzskatīt parādīto trajektoriju vidējo garumu – vairākas izsekotās pazīmes ļauj mazināt atsevišķo kļūdaini izsekoto pazīmju ietekmi.

2.2.5 Atklāto automašīnu attēlu saglabāšana

Domājot par automašīnu daļējas aizklāšanās problēmu, tika izveidota metode atklāto automašīnu attēlu saglabāšanai. Atklājot kādu pietiekami platu un pietiekami ilgi pastāvošu intervālu, tajā ietvertais intensitātes līnijas apgabals tiek saglabāts vektorā. Kad intervāls tiek slēgts, vektorā esošās līnijas tiek sakārtotas 2D matricā, kas kā attēls tiek saglabāta datora cietajā diskā. Šāds attēls saturēs vienu automašīnu, ja intervāls detektēji pareizi, savukārt gadījumos, kad divas vai vairākas mašīnas detektētas kā viens intervāls, attēlā būs redzamas vairākas mašīnas.

Šāda attēlu datubāze, kuras piemēri parādīti attēlā 2.6 varētu noderēt dažādo gadījumu izdalīšanai – perspektīvā šo attēlu atsevišķa apstrāde varētu ļaut saskaitīt arī aizklāšanās dēļ vienā intervālā saplūdušas automašīnas.

Modificējot programmu, varētu iegūt arī krāsainus mašīnu attēlus. Šādi attēli varētu perspektīvā ļaut klasificēt automašīnas un atklāt operatīvā transportlīdzekļa esamību kadrā.



Att. 2.6: Atklāto automašīnu attēli

2.3 Eksperimenti un testi

C++ kodā īstenotais algoritms testēts ar tiem pašiem video, kas tika izmantoti iepriekšējās programmas Simulink versijas testēšanai. Tā kā uzlabotajā versijā sliekšņi tiek atrasti automātiski, bija iespējams samazināt testu skaitu, jo nav vairs iteratīvi jāpiemeklē sliekšņi dažādajiem video. Vidēji programma noteica video braucošo automašīnu skaitu ar 90% procentu precizitāti. Darbinot programmu ar video failiem, kas uzņemti labos apstākļos vasaras laikā precizitāte bija virs 90%, ar ziemā uzņemtajiem video – zem.

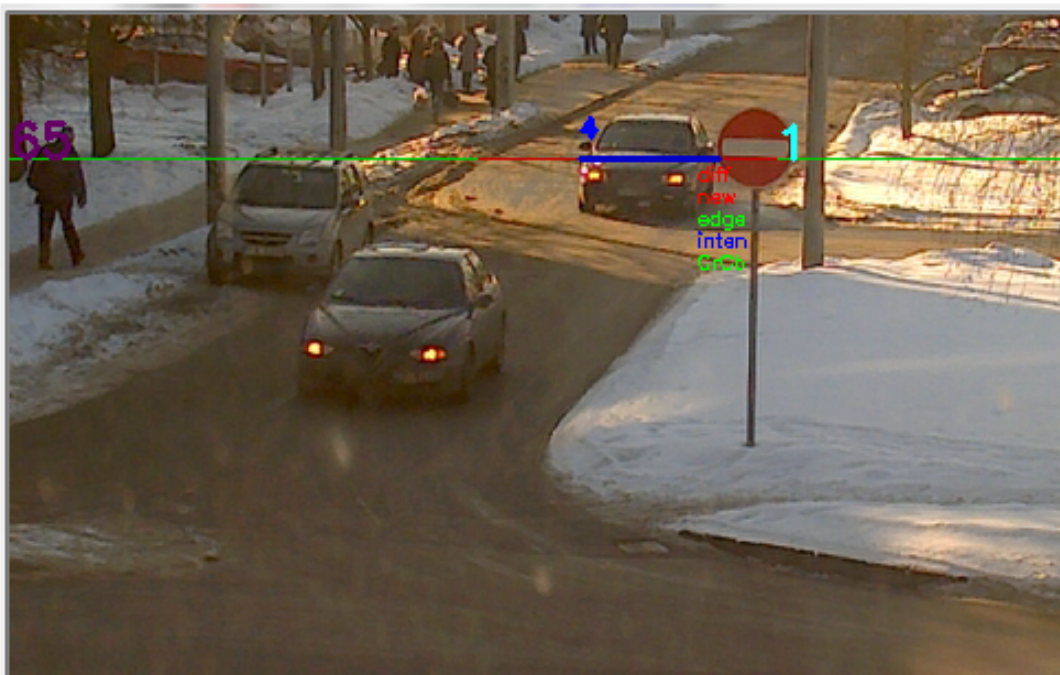
Viens no piedāvātā algoritma paredzētajiem plusiem ir samērā zemās prasības pret skaitļošanas jaudām – īstenošana C++ valodā ļauj šīs prasības novērtēt. Darbinot šī brīža izstrādes stadijā esošo programmu ar 640×480 izmēra video, apstrādājot 15 kadrus sekundē, programma izmantoja ap 30% no divu kodolu 1.3GHz procesora, un tai bija nepieciešami 12MB atmiņas. 720×576 pikseļu video ar 25 kadriem sekundē attiecīgie rādītāji bija ap 40% un 16MB.

Programma tika testēta arī darbinot to ar tiešā laikā pienākošiem video materiāliem. Šim nolūkam programmā iestrādāta iespēja saņemt datus no tīmekļa kameras AXIS P1344. Attēlā 2.7 parādīta algoritma darbība ar reālā laikā iegūtu video materiālu. Pirmie algoritma precizitātes testi ziemas laikā ar video kameru uzrādīja skaitīšanas precizitāti ap 80%.

2.4 Kopsavilkums un secinājumi

Jau iepriekšējā ceturksnā atskaitē tika izvēlēta konkrētā metode automašīnu atklāšanai, kurā ieguldīt visu turpmāko darbu. Tagad piedāvātais algoritms īstenots C++ kodā. Ar algoritmu tiek veikti eksperimenti un testi, pēc kuru rezultātiem tas tiek uzlabots, padarot to adaptīvāku mainīgajiem apstākļiem, kā arī samazinot kļūdas, ko algoritms pieļauj.

Šobrīd izveidotais objektorientētais kods ļauj ērti izmantot dažādus parametrus uz pētāmās līnijas mašīnu atpazīšanai, kā arī ļauj ātri pievienot jaunus parametrus. Tādā gadījumā ir tikai jāizdala uz līnijas attiecīgais parametrs (piemēram, robežu atrašana), jāizveido *Line* klases objekts, un pēc



Att. 2.7: Programmas rezultāta vizualizācija uz tīmekļa kameras video

vajadzības jāizsauc jau izveidotās, turpmākai apstrādei nepieciešamās metodes (filtrēšana, fona uzkrāšana un atņemšana, sliekšņu atrašana un pielietošana).

Šobrīd programma jau tiek darbināta ar reālā laikā iegūtiem video, kas ļauj novērtēt tās darbību nemitīgi mainīgos apstākļos, kuros iespējams atklāt labojamas algoritma nepilnības, kas neparādītos ierobežotā iepriekš ierakstītu video materiālu datu bāzē. Pēc pirmajiem testiem tiešsaistes režīmā secināms, ka piedāvātais algoritms darbojas reālā laikā.

2.5 Nākotnes perspektīvas

Turpmākais mērķis ir pilnveidot izstrādāto algoritmu. Pirmkārt jāpārbauda mašīnu atklāšanas kļūdu samazināšanu. Šim nolūkam var mēģināt meklēt vēl kādus zīmīgus parametrus uz pētāmās līnijas, kas noteiktu mašīnas esamību uz tās. Lielākais kļūdu skaits rodas mašīnām braucot tuvu vienai aiz otras. Šīs problēmas turpmākai risināšanai tika izveidota mašīnu attēlu saglabāšana. Tomēr vienkāršāks paņēmieni šajā situācijā iespējams būtu labāks kameras novietojums.

Piedāvātais algoritms apraksta vienas līnijas apstrādi automašīnu skaitīšanai. Nākamais solis ir izmantot vairākas šādas paralēlas līnijas vienlaicīgi. Tas ne tikai ļaus koriģēt vienas nepareizas līnijas rādījumu attiecībā pret pārējām līnijām, uzlabojot atklāšanas precizitāti, bet arī ļaus iegūt papildus parametrus katrai automašīnai. Piemēram, vairākas līnijas ļautu noteikt automašīnas gabarītus un iespējams arī kustības trajektoriju.

Nodaļa 3

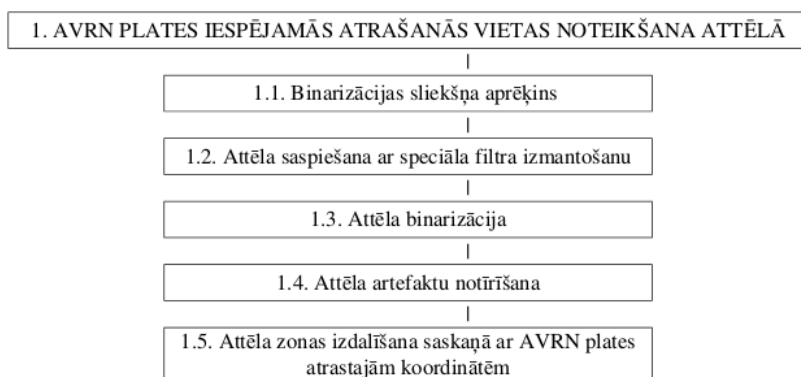
MITSTRegNo grupa

3.1 Ievads

Atskaides periodā tika veikti iepriekš piedāvātā automobiļu valsts reģistrācijas numuru (AVRN) noteikšanas algoritma efektivitātes pētījumi. IZanalizēta attēla histogrammas, numura izmēra un citu parametru ietekme uz numura zīmes stāvokļa noteikšanas precizitāti attēlā. Piedāvāti jauni numura zīmes attēla filtrācijas algoritmi, jauns simbolu atpazīšanas algoritms (OCR), izstrādāta jaunu algoritmu analīzes sistēma.

3.2 Automobiļu valsts numura zīmes noteikšanas tehnoloģijas apraksts

Šajā projekta etapā veiktie pētījumi apstiprināja izvēlētais (AVRN) noteikšanas tehnoloģijas, kas aprakstīta iepriekšējā atskaitē, pareizību. Atgādināsim, ka tās pamat princips ir visa procesa sadalīšana divos līmeņos: pirmais, (AVRN) plates atrašanās vietas noteikšana attēlā, otrais - (AVRN) noteikšana. Savukārt pirmā (attēls 3.1) un otrā (attēls 3.2) līmeņa uzdevumu risināšana, tiek sadalīta vairākos etapos. Pirmā līmeņa pirmajā etapā tiek noteikti vairāki attēla raksturlielumi, kā, piemēram, vidējā intensitāte un dispersija, kas tiek izmantota binarizācijas sliekšņa aprēķinam. Pirmā līmeņa otrajā etapā notiek attēla saspiešana ar filtrāciju. To izpilda, lai samazinātu turpmāko attēla analīzes laiku. Lai saglabātu mūs interesējošās attēla īpašības, tiek izmantots speciāls filtrs. Tad, izmantojot aprēķināto binarizācijas sliekšni, izpildām attēla binarizāciju. Rezultātā iegūstam melnbaltu attēlu (bez pelēkiem toņiem), kurā uz melna fona izvietotas viena vai parasti vairākas baltas zonas. Viena no tām atbilst numura platei, pārējās – artefakti. To nodzēšanai izmantojam speciālu tīrīšanas algoritmu, kas, izmantojot paredzamos (AVRN) plates izmērus, nodzēš visus pārāk maza izmēra baltos objektus.



Att. 3.1: Pirmais līmenis

Rezultātā, vairums gadījumu, paliek tikai viena balta zona, kas norāda uz (AVRN) plates atrašanās vietu attēlā. Izmantojot iegūtās zonas koordinātes, izdalām atbilstošu oriģinālā attēla daļu, kurā iespējams atrodas numurs. Tālāk jārisina otrā līmeņa uzdevums - (AVRN) noteikšana, izmantojot OCR (Optical Character Recognition). Otrā līmeņa uzdevumu risināšanas etapi parādīti attēlā 3.2.



Att. 3.2: Otrais līmenis

Otrā līmeņa uzdevumu risināšanas pirmajā etapā notiek atrastās zonas filtrācija, izmantojot attēla kontrastu paaugstinošas maskas. Šajā gadījumā filtrācija būs maskas un zonas konvolūcijas operācija, ko pielieto visiem zonas pikseļiem. Ja oriģinālā attēla apstrādājamo zonu (ar izmēru $M \times N$) apzīmē ar f , bet filtra masku (ar izmēru $m \times n$) ar w , tad, maskas un zonas konvolūcijas operāciju var izteikt sekojoši:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) * F(x + s, y + t)$$

kur

$g(x, y)$ – konvolūcijas operācijas rezultāts;

$a = (m-1)/2$;

$b = (n-1)/2$;

$x = 0, 1, 2, \dots, M-1$;

$y = 0, 1, 2, \dots, N-1$.

Otrajā etapā tiek veikta saistīto pikseļu kopumu meklēšana, kas varētu būt simbolu attēli. Ar saistīto pikseļu kopumu šeit saprotam pikseļu kopumu, kurā katrs no pikseļiem robežojas vismaz ar vienu citu šī kopuma pikseli. Šeit tiek izmantota īpašība, kas piemīt visiem cipariem no 0 līdz 9 un visiem latīņu burtiem, un kura nosaka, ka katrs no iepriekš uzskaitītajiem simboliem veido savu saistīto pikseļu kopumu. Pēc visu kopuma saistīto pikseļu atrašanas, tiek noteikts to kopējais skaits, apzīmēsim to ar R_c .

Ja $R_{min} < R_c \leq R_{max}$, tad izdarām secinājumu, ka tas ir simbols. Pretējā gadījumā pieņemam, ka tas ir artefakts un attiecīgais pikseļu kopums tiek pārkrāsots baltā krāsā (fona krāsā). Kā R_{min} tiek izmantots pikseļu skaits attēlā „1”, bet kā R_{max} – simbola „W” pikseļu skaits. Pēdējā etapā tiek startēta OCR apakšprogramma, kurai, kā ieejas dati, tiek secīgi padoti visi atrastie AVRŅ plāksnes apgabali, kas satur saistīto pikseļu kopumu. Tad tiek pārbaudīta atrasto simbolu atbilstība Latvijas AVRŅ šablonam (pirmie divi simboli - burti, četri pēdējie simboli - cipari). Ja kādas no pozīcijām neatbilst, tiek veiktas saistīto simbolu kopuma korekcijas un vēlreiz pārbaudīta OCR darbība.

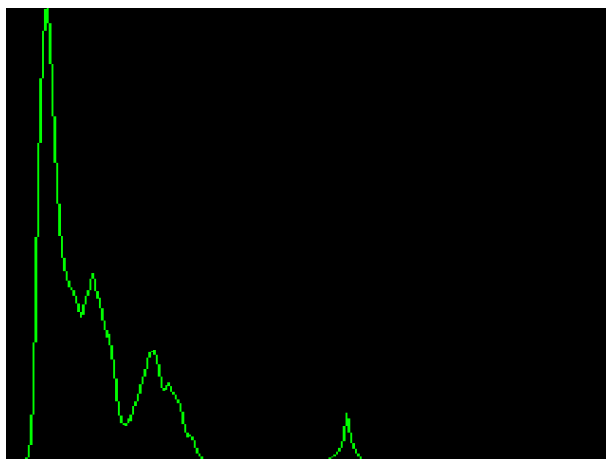
3.3 AVRŅ noteikšanas piedāvātās tehnoloģijas etapu darbības rezultātu apraksts

Piedāvātā tehnoloģija paredzēta attēlu analīzei, kas iegūti izmantojot infrasarkanās (IR) gaismas prožektora apgaismojumu. IR prožektora izmantošana nosaka to, ka visi no video kameras iegūtie attēli būs melnbalti. Izmantojot šo piedāvāto tehnoloģiju, būtiski atvieglojas attēlu tālākā apstrāde un paaugstinās rezultātu precizitāte. Kamera un prožektors jāuzstāda virs ceļa klājuma. Šādos apstākļos AVRŅ plāksnes izmērs visos attēlos ir aptuveni vienāds. Attēlā parādīts, kā izskatās automobiļa attēls, izmantojot IR prožektoru (attēls 3.3). Jāpiezīmē, ka attēlu tumšais fons, uz kura redzama AVRŅ plāksne, tāds ir vienmēr, neatkarīgi no diennakts laika.



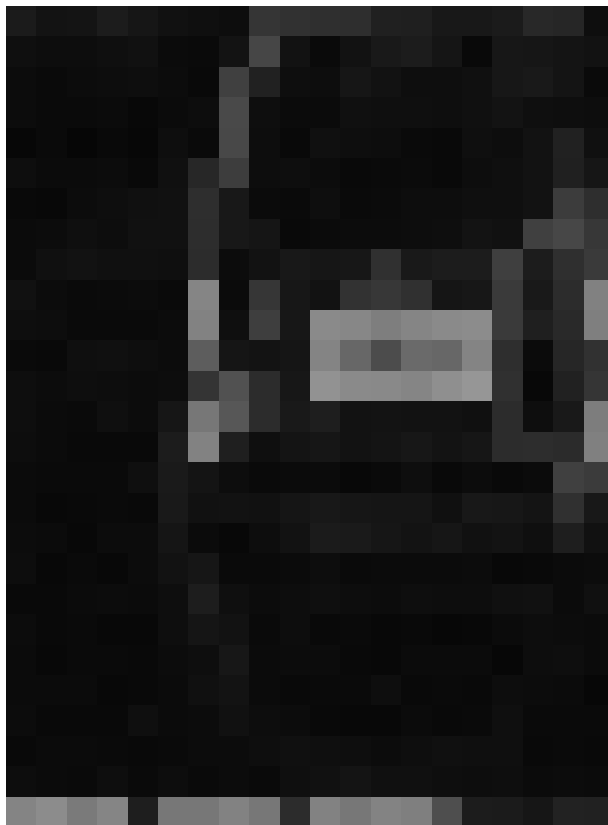
Att. 3.3: Automobiļa attēls, izmantojot IR prožektoru

Sākuma attēla histogramma parādīta attēlā 3.4. Lielākās pikseļu masas intensitāte atrodas tuvāk nullei. Atsevišķs, neliels izvirzījums histogrammas labajā pusē, norāda uz AVRŅ plāksnes gaišo pikseļu atrašanos attēlā. Vispārējā gadījumā, attēla intensitātes vidējā vērtība būs diapazonā no 20 līdz 60. No histogrammas redzams, ka pikseļu intensitātes dispersija ir samērā neliela. Intensitātes vidējā vērtība un dispersija tiek izmantota, lai noteiktu binarizācijas sliekšni.



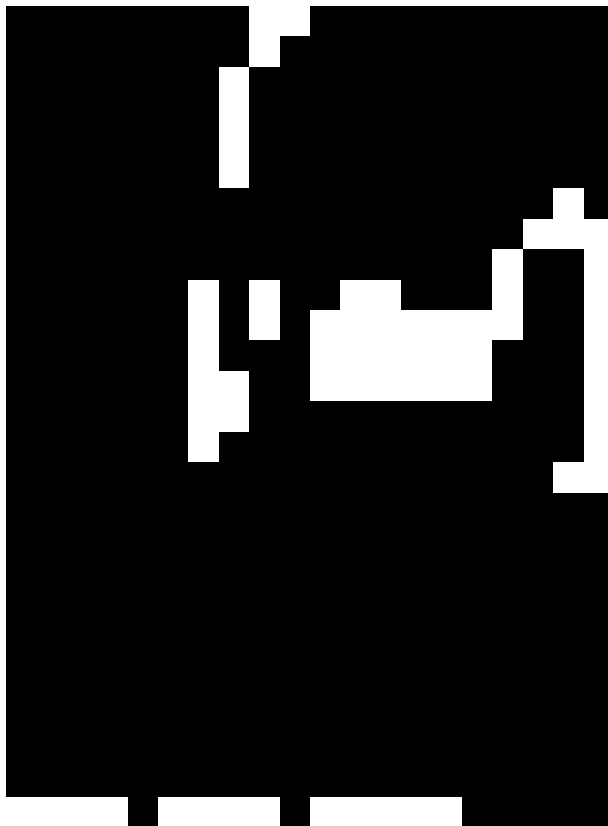
Att. 3.4: Attēla histogramma

Nākošajā attēlā 3.5 parādīts, kā izskatās attēls pēc saspiešanas un speciālā filtra pielietošanas (pirmā līmeņa otrais etaps). Pēc saspiešanas iegūtā attēla izmēri nav proporcionāli ar sākuma attēla izmēriem. Tas saistīts ar to, ka pielietotais saspiešanas koeficients pa horizontāli ir lielāks nekā saspiešanas koeficients pa vertikāli. Tas atspoguļo AVRN plāksnes platuma un augstuma attiecību (vairums standarta AVRN plāksnēm).



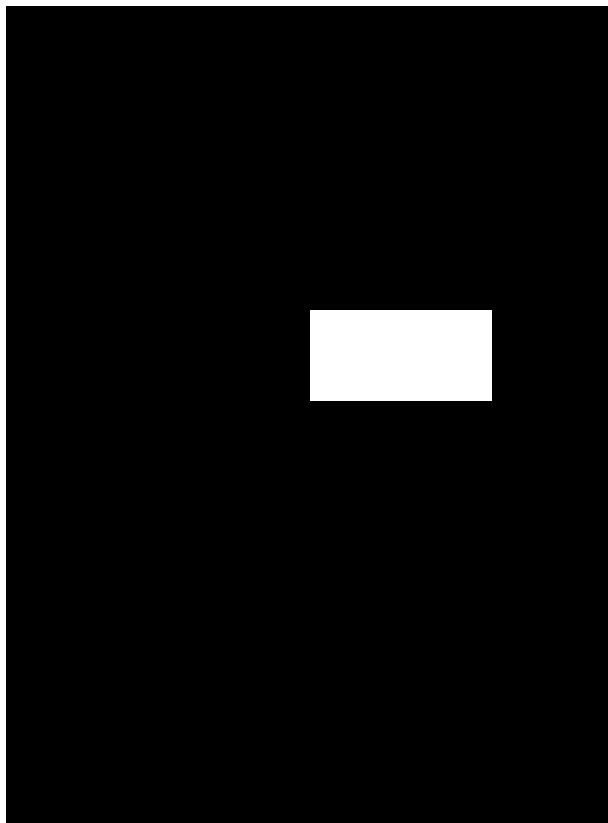
Att. 3.5: Attēls pēc saspiešanas un speciālā filtra pielietošanas

Attēls pēc binarizācijas parādīts attēlā 3.6. Šis un turpmākie attēli tiek mākslīgi izstiepti pa horizontāli, lai uzskatāmāk nodemonstrētu piedāvātās tehnoloģijas pielietojuma efektu turpmākos etapos. Patiesībā, visu saspiesto attēlu formāts ir tāds, kā parādīts attēlā 3.5. Jāatzīmē, ka līdzās ar AVRN plāksnes zonu, attēlā parasti ir arī citas gaišas zonas, kas nenovēršami rodas attēla pārveidojumu rezultātā. Šādas zonas ir analogas traucējumiem signālu pārveidošanā/pārraidē, un tiek sauktas par artefaktiem.



Att. 3.6: Attēls pēc binarizācijas

Saskaņā ar 1.attēla shēmu, nākošajā attēla apstrādes etapā realizēta artefaktu atrašana un nodzēšana (pārkrāsošana fona krāsā). Artefaktu identifikācijas kritērijs ir AVRN plāksnes orientējošie izmēri pikseļos. Vispirms, algoritms analizē visu attēlā atrasto gaišo laukumu vertikālos izmērus un nodzēš tos, kuriem vertikālais izmērs mazāks par AVRN plāksnes izmēru. Pēc tam analogiska procedūra tiek izpildīta atrasto zonu horizontālo izmēru analīzei. Attēla izskats, pēc artefaktu nodzēšanas, parādīts attēlā 3.7.



Att. 3.7: Attēls pēc artefaktu nodzēšanas

Iegūtā rāmīša koordinātu pārvešana uz oriģinālā attēla 3.8 atspoguļo pirmā līmeņa etapu darbības kvalitāti. Izvēlēti sekojoši AVRN plāksnes meklēšanas pozitīva rezultāta kritēriji: rezultāts tiek uzskatīts par pozitīvu, ja atrastā zona ietver AVRN plāksni, t.i. neviena no rāmīša malām, kas norobežo atrasto zonu, nešķērso nevienu AVRN simbolu. Cits AVRN plāksnes meklēšanas rezultāta kritērijs - AVRN plāksnes izmēru un atrastās zonas sakrīšanas precizitāte. Jo precīzāk tie sakrīt, jo precīzāki un efektīvāki būs otrā līmeņa etapu darbības rezultāti (attēls 3.2). Dziļāks pirmā līmeņa rezultātu novērtējuma kritēriju apraksts sniegts nākošajā nodaļā.

Kadri ar kustīga automobiļa attēliem, ko saņem no IP kameras, neizceļas ar augstu attēla asumu. Cilvēka acij tas var būt nav pamanāmi, bet atpazīšanas algoritmiem, kas analizē un salīdzina pikseļu kopumu intensitātes - samērā būtiski. Attēlu asuma trūkums var būt saistīts ar sekojošiem faktoriem: pirmkārt, ar to, ka automobiļi pārvietojas, un pie tam ar dažādiem ātrumiem; otrkārt, apkārtējās vides ietekme (apgaismojuma, lietus, sniega), kas, izmantojot IR prožektoru, lai arī mazinās, tomēr netiek novērsta pilnībā; treškārt, automobiļa krāsa un citi faktori. Sakarā ar to, otrā līmeņa pirmajā etapā notiek attēla apstrāde ar speciālas maskas palīdzību, kas ļauj paaugstināt attēla asumu (skat. 2.nodaļu). Nākošajā etapā realizēta saistītu pikseļu kopumu meklēšana, kas apmierina nosacījumu $R_{min} < R_c \leq R_{max}$ un artefaktu dzēšana. Pēdējā etapā, ar OCR palīdzību, tiek veikta simbolu atpazīšana. Šo etapu darba rezultāti parādīti attēlā 3.9, 3.10 un attēlā 3.11.



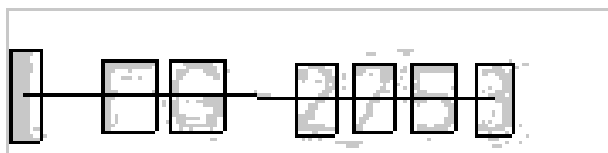
Att. 3.8: Iegūtā rāmīša koordinātu pārvešana uz oriģinālā attēla



Att. 3.9: AVRN



Att. 3.10: AVRN apstrāde



Att. 3.11: Apstrāde ar OCR

3.4 AVRN noteikšanas piedāvātās tehnoloģijas pielietojšanas statistiskie rezultāti

Pirmā eksperimenta rezultāti apkopoti tabulā 3.1 - 3.2. Šajā eksperimentā kamera ar IR prožektoru tika uzstādīta ceļa nomalē. Tas bez šaubām pasliktināja darbības rezultātu, jo AVRN plates noteikšanas piedāvātā tehnoloģija piemērota, ja kamera un prožektors uzstādīti tieši virs ceļa klājuma aptuveni četru metru augstumā. Tomēr tikai 3 gadījumos no 86, numura zīme netika detektēta (31, 42 un 77 rinda). Turpmāk apskatīsim šos gadījumus, ar ko tas saistīts.

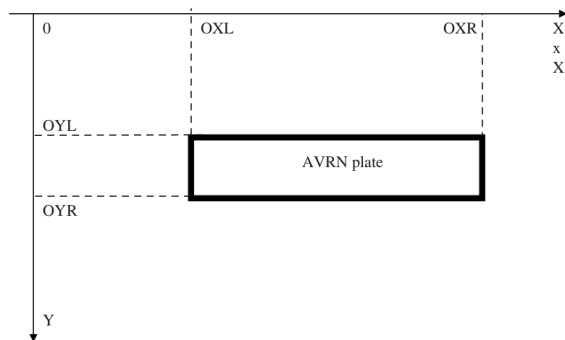
Kolonnā «TreshH» dotas aprēķinātās binarizācijas sliekšņa vērtības katram attēlam atsevišķi. Kolonnās OXL, OYL, OXR, OYR dotas atbilstošās AVRN plates koordinātes, kas ar roku nomērītas katrā attēlā. Kolonnās FXL, FYL, FXR, FYR dotas atbilstošās AVRN plates koordinātes, kuras noteikusi programma, izmantojot piedāvāto AVRN plates noteikšanas tehnoloģiju. Tā kā AVRN platei ir taisnstūra forma, tad tās atrašanās vietas noteikšanai ir nepieciešams zināt tikai plates diagonāles divu galējo punktu koordinātes, piemēram, kreiso augšējo un labo apakšējo (attēls 3.12).

Tabula 3.1: Rezultāti

Nr	TreshH	OXL	OYL	OXR	OYR	N _o	FXL	FYL	FXR	FYR	Di _X L	Di _Y L	Di _X R	Di _Y R
1	61	347	96	483	121	1	324	80	486	130	23	16	3	9
2	57	149	323	313	356	2	128	325	320	364	21	-2	7	8
3	58	236	215	392	245	1	217	204	403	252	19	11	11	7
4	45	422	109	560	136	1	405	110	567	140	17	-1	7	4
5	54	382	251	526	278	2	364	250	532	280	18	1	6	2
6	63	170	271	346	310	1	140	270	350	315	30	1	4	5
7	75	222	202	387	236	2	198	182	396	247	24	20	9	11
8	60	361	289	521	323	1	352	286	544	325	9	3	23	2
9	43	276	89	415	117	1	270	88	432	121	6	1	17	4
10	35	0	318	131	348	2	26	312	156	348	-26	6	25	0
11	57	97	229	249	258	1	90	231	270	264	7	-2	21	6
12	32	280	141	408	165	3	275	135	425	171	5	6	17	6
13	43	144	227	327	263	1	144	224	360	266	0	3	33	3
14	52	225	230	393	265	1	198	224	495	280	27	6	102	15
15	51	34	258	180	288	1	29	252	203	288	5	6	23	0
16	66	10	237	173	265	1	32	231	192	275	-22	6	19	10
17	46	0	120	117	144	1	23	117	138	153	-23	3	21	9
18	61	203	326	368	358	2	198	324	396	360	5	2	28	2
19	44	303	291	463	324	2	288	273	512	338	15	18	49	14
20	41	3	204	130	230	2	25	200	150	230	-22	4	20	0
21	59	333	279	476	305	1	308	280	504	310	25	-1	28	5
22	51	420	129	550	152	1	416	126	572	153	4	3	22	1
23	55	277	259	449	299	1	272	256	476	304	5	3	27	5
24	48	304	160	455	190	1	300	156	480	192	4	4	25	2
25	32	0	256	117	285	2	23	264	138	297	-23	-8	21	12
26	43	254	319	415	349	1	224	312	416	360	30	7	1	11
27	65	179	343	340	378	1	160	336	352	378	19	7	12	0
28	61	289	294	445	326	2	279	288	465	336	10	6	20	10
29	40	267	219	433	252	1	264	208	462	260	3	11	29	8
30	44	10	284	161	314	1	30	276	180	324	-20	8	19	10
31	44	485	200	602	224	0	0	0	0	0	0	0	0	0
32	52	291	221	456	254	1	264	221	462	260	27	0	6	6
33	43	343	210	507	243	1	320	208	512	247	23	2	5	4
34	35	16	212	144	237	1	25	210	150	240	-9	2	6	3
35	36	0	216	126	245	2	25	209	150	253	-25	7	24	8
36	52	310	268	460	300	1	300	264	480	300	10	4	20	0
37	46	360	165	508	194	1	348	165	522	198	12	0	14	4
38	44	476	224	602	250	1	475	220	575	260	1	4	-27	10
39	55	0	189	131	217	1	26	187	156	220	-26	2	25	3
40	37	0	339	152	374	1	30	336	150	378	-30	3	-2	4
41	70	369	163	516	191	2	348	165	522	198	21	-2	6	7
42	54	482	90	602	114	0	0	0	0	0	0	0	0	0
43	50	441	247	586	280	1	435	247	551	286	6	0	-35	6 height

Tabula 3.2: Rezultāti turp.

Nr	TreshH	OXL	OYL	OXR	OYR	N _o	FXL	FYL	FXR	FYR	Di _X L	Di _Y L	Di _X R	Di _Y R
44	71	259	231	426	263	1	231	228	363	264	28	3	-63	1
45	37	15	259	158	287	2	28	253	168	297	-13	6	10	10
46	47	463	199	597	225	1	442	200	572	230	21	-1	-25	5
47	49	337	224	501	259	1	320	224	512	266	17	0	11	7
48	65	312	252	412	286	2	300	247	420	299	12	5	8	13
49	31	231	259	382	292	1	210	260	390	299	21	-1	8	7
50	52	0	212	139	241	1	27	209	162	242	-27	3	23	1
51	39	346	290	507	322	2	320	288	512	324	26	2	5	2
52	58	347	249	494	279	1	348	252	522	288	-1	-3	28	9
53	51	248	210	405	240	1	248	204	434	252	0	6	29	12
54	45	8	241	141	267	1	26	240	156	270	-18	1	15	3
55	52	16	205	169	239	1	30	208	180	247	-14	-3	11	8
56	66	190	247	360	280	1	170	247	374	286	20	0	14	6
57	67	409	253	557	281	1	406	253	551	286	3	0	-6	5
58	71	274	295	434	328	1	256	312	448	338	18	-17	14	10
59	31	233	260	383	292	1	210	264	390	300	23	-4	7	8
60	67	272	184	428	216	1	248	180	434	228	24	4	6	12
61	72	302	237	452	267	1	300	216	510	276	2	21	58	9
62	53	222	302	382	335	1	192	299	384	338	30	3	2	3
63	72	211	213	382	249	1	204	210	408	252	7	3	26	3
64	41	125	175	269	203	2	112	176	280	209	13	-1	11	6
65	50	21	183	181	213	1	32	180	192	216	-11	3	11	3
66	56	7	186	181	218	1	34	180	204	228	-27	6	23	10
67	55	179	276	343	313	1	160	280	352	322	19	-4	9	9
68	35	0	184	119	209	2	23	180	115	210	-23	4	-4	1
69	35	438	208	574	237	1	432	209	567	242	6	-1	-7	5
70	50	291	305	459	340	2	264	294	462	350	27	11	3	10
71	39	299	311	463	345	1	288	299	480	351	11	12	17	6
72	62	365	261	512	290	1	348	264	522	297	17	-3	10	7
73	57	369	131	514	155	1	348	126	522	162	21	5	8	7
74	57	393	75	515	101	1	384	60	576	100	9	15	61	-1
75	72	245	229	406	261	1	224	228	416	264	21	1	10	3
76	64	439	54	569	78	1	416	63	572	81	23	-9	3	3
77	48	0	231	152	269	0	0	0	0	0	0	0	0	0
78	41	10	225	145	252	1	27	170	297	260	-17	55	152	8
79	56	273	278	427	308	2	270	276	450	312	3	2	23	4
80	73	120	246	297	282	1	105	238	315	280	15	8	18	-2
81	57	420	147	561	174	1	392	140	560	180	28	7	-1	6
82	58	229	238	403	275	1	204	238	408	280	25	0	5	5
83	59	343	154	491	181	1	319	150	493	190	24	4	2	9
84	73	255	215	420	249	1	231	208	429	247	24	7	9	-2
85	57	396	165	540	194	1	364	165	560	198	32	0	20	4
86	39	6	215	137	238	2	26	207	156	243	-20	8	19	5 height



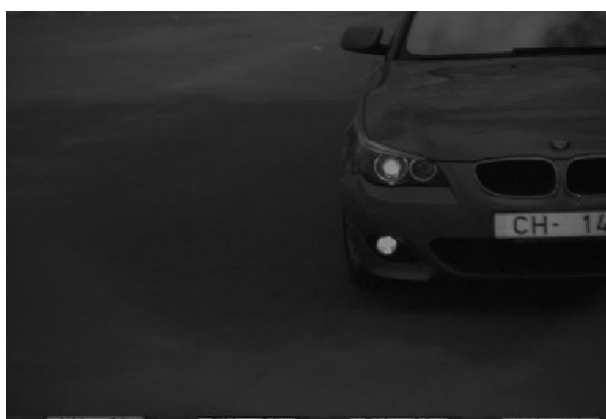
Att. 3.12: AVRN atrašanās vietas noteikšana

Kolonnā « N_o » norādīts programmas atrasto zonu skaits, kas atbilst AVRN plates kritērijiem. Divdesmit gadījumos no 86 tika atrastas vairāk nekā viena zona (vienā gadījumā 3 zonas un 19 - divas). Tas nav kritiski, jo turpmāk izmantojamais OCR algoritms, atlasīs tikai to zonu, kurā būs iespējams detektēt simbolus, bet pārējās nodzēsīs (bez simboliem). Kolonnas $Di_X L$, $Di_Y L$, $Di_X R$, $Di_Y R$ norāda uz atšķirību starp reālajām AVRN plates koordinātēm un programmas noteiktajām koordinātēm. Atbilstošo koordinātu pozitīvas atšķirības vērtības nozīmē, ka programmas atrastā koordināte "aptver" reālo, t.i. programmas noteiktā zona ir lielāka par reālās zonas platību. Tas ir pieņemams rezultāts. Sliktāk, ja šī atšķirība ir negatīva, kas norāda uz zonas un AVRN plates daļēju pārklāšanos. Tomēr iegūto rezultātu tālākā analīze parādīja, ka vairumā gadījumu AVRN plates pārklāšanās attiecās galvenokārt uz tās kreiso robežu. Ja aplūko automobiļu AVRN plates attēlus (attēls 3.3, attēls 3.13, 3.14, 3.15), redzams, ka AVRN plates kreisajā pusē (aptuveni 15% no plates platuma) nav nekādas informācijas, tādēļ šīs plates daļas „nogriešana” neietekmē OCR darbības kvalitāti. Nenozīmīga plates pārklāšanās no augšas, 1-2 pikseli (zonas samazinājums attiecībā pret plati) neietekmē OCR darbību, jo ar to netiek skarti AVRN simboli.

Attēli, kuros programma nav atradusi nevienu zonu, parādīti attēlā 3.13, 3.14, 3.15. AVRN plate 3.13 attēlā atrodas slīpi attiecībā pret horizontāli (šis trūkums saistīts ar kameras novietojumu ceļa nomalē), tāpēc programma nav spējusi korekti atrast numura zonu. Ne visa AVRN plate 3.14 attēlā nokļuvusi kadrā, un programma, analizējot šīs zonas izmērus, nodzēsusi zonu, jo tās izmēri ir mazāki nekā nepieciešams.



Att. 3.13: AVRN nav atrasts



Att. 3.14: AVRN nav atrasts



Att. 3.15: AVRN nav atrasts

Visbeidzot 3.15 attēls izrādījies pārāk gaišs, iespējams, sakarā ar kameras un IR prožektora sinhronizācijas defektu. Tas varētu būt galvenais iemesls, ka programma nav detektējusi AVRN plates zonu šajā attēlā. Otrā eksperimenta rezultāti, kad kamera un IR prožektors tika novietots tā, kā tas paredzēts izstrādātajā tehnoloģijā, izrādījās ievērojami labāki, nekā pirmajā eksperimentā, jo attēlu kvalitāte, no piedāvātās tehnoloģijas viedokļa, bija ievērojami labāka (attēls 3.16).



Att. 3.16: Attēls ar IR prožektoru

Otrā eksperimenta rezultātā AVRN plates atrašanās zona tika atrasta visos attēlos (100%), samazinājās gadījumi, kad tika atrastas vairāk, kā viena zona un ievērojami pieauga zonas izmēru noteikšanas precizitāte, t.i. zonas izmēru atbilstība AVRN plates izmēriem.

3.5 Secinājumi un turpmāko darbu plāns

Tālākie pētījumi tiks veikti, lai pilnveidotu piedāvāto tehnoloģiju un noteiktu optimālus algoritma parametrus, veicot eksperimentus.