IEGULDĪJUMS TAVĀ NĀKOTNĒ

TRANSPORTA LĪDZEKĻU ATPAZĪŠANA, REĢISTRĒŠANA UN INTELIĢENTAS SATIKSMES VADĪBAS LOĢIKA

Metodes apraksts

Eiropas reģionālās attīstības fonds Darbības programma: “Uzņēmējdarbība un inovācijas” ;

Prioritāte: 2.1.“Zinātne un inovācijas”

Pasākums: 2.1.1. “Zinātne, pētniecība un attīstība”

Aktivitāte: 2.1.1.1.aktivitāte "Atbalsts zinātnei un pētniecībai"

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.

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

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

Projekta zinātniskais vadītājs: Leo Seļāvo

ELEKTRONIKAS UN DATORZINĀTŅU INSTITŪTS
Rīga, Latvija

2013. gads

Saturs

Ievads **3**
MITS . . . 3

1 Inteliģentas satiksmes vadības sistēmas arhitektūra **4**
1.1 Ievads . . . 4
1.2 Arhitektūra . . . 4
1.2.1 Moduļi . . . 5
1.2.2 Sistēmas kodols . . . 6
1.2.3 Sistēmas demonstratora prototips un lietotāja saskarne . . . 10

2 Inteliģentas satiksmes vadības loģikas modulis un algoritms **12**
2.1 Ievads . . . 12
2.2 Lēmumu pieņemšanas algoritms . . . 12

3 Multimodālu sensoru izmantošana transporta līdzekļu reģistrācijai un klasifikācijai **14**
3.1 Motivācija . . . 14
3.2 Ievads . . . 14
3.3 Magnētiskā sensora mote . . . 14
3.3.1 Magnētisko sensoru izmantošana transporta līdzekļu detektēšanā un klasifikācijā. . . 14
3.3.2 Tehniskais risinājums un darbības princips. . . . 15
3.3.3 Signālu apstrādes algoritms, realizācija uz motes . . . 17
3.3.4 Signāla apstrādes algoritms uz mini skaitļotāja . . . 19
3.4 Siltuma sensora mote . . . 21
3.5 Bezvadu datu komunikācija starp sensoru mezgliem . . . 23
3.6 Lāzera sensora mote . . . 24
3.7 Ceļa stāvokļa un laika apstākļu noteikšanas sensori . . . 27
3.8 Mikrofonu masīvs . . . 29
3.8.1 Automašīnu detektēšana izmantojot mikrofonu masīvu . . . 29
3.8.2 Tehniskie parametri un darbības princips . . . 30
3.8.3 Apstrādes algoritms un tā realizācija ARM Cortex M4F procesorā . . . 32
3.8.4 Operatīva transporta līdzekļa detektēšana . . . 33

4 Transporta līdzekļu atpazīšana un algoritmi **34**
4.1 Ievads . . . 34
4.2 Saistītie risinājumi . . . 34
4.3 Automašīnu atklāšana . . . 35
4.3.1 Metode - intervāli uz atklāšanas līnijas . . . 35
4.3.2 Atklāšanas algoritma uzlabojumi . . . 38
4.3.3 Mašīnu atklāšana naktī . . . 46
4.3.4 Testi ar uzlaboto algoritmu mašīnu atklāšanai . . . 47
4.4 Mašīnas izsekošana, ātruma un izmēru noteikšana . . . 48
4.4.1 Atklāšanas līniju izvietošanas iespēju paplašināšana . . . 48
4.4.2 Vairāku līniju rādījuma apvienošana . . . 51
4.4.3 Automašīnu parametru noteikšana . . . 53
4.4.4 Automašīnu klasificēšana . . . 56
4.4.5 Testi ar vairāku atklāšanas līniju algoritmu . . . 57
4.5 Avārijas situāciju un pārkāpumu atklāšana . . . 58

4.5.1 Avāriju atklāšana pēc mašīnu apstāšanās 59
4.5.2 Avāriju atklāšana pēc mašīnu trajektoriju analīzes 59
4.5.3 Luksofora signāla neievērošanas atklāšana 62
4.6 Kopsavilkums un secinājumi 63

5 Transporta līdzekļu reģistrācija numuru noteikšana un algoritmi 65
5.1 Numuru atpazīšanas algoritma apraksts 65
5.2 Kameras novietojuma apraksts 74
5.3 Datu sūtīšana izmantojot GSM tīklu 74

Ievads

Anotācija

Metode transporta līdzekļu atpazīšanai, reģistrēšanai un inteliģentas satiksmes vadības loģikai ir izstrādāta ERAF atbalstītā projekta "Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija" (MITS) ietvaros. MITS projekta mērķis bija modulāra sensoru informācijas savācēja, 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. Tehnoloģija paredzēta lietošanai ārpus apdzīvotām vietām uz ceļiem ar vienu braukšanas joslu katrā virzienā. Šajā dokumentā ir aprakstīta minētā metode, sadalīta sekojošās loģiskās sadaļās: sistēmas arhitektūra un satiksmes vadības loģika, transporta līdzekļu atpazīšana un reģistrācija ar video un attēlu apstrādes paņēmieniem, transporta reģistrācijas numura zīmju atpazīšana un sensoru pielietojumi un to datu apstrāde. Metodes aprakstā ievietoti attēli no izveidotās sistēmas demonstratora/prototipa, kā arī izveidotie algoritmi un shēmtehniskie risinājumi.

"Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija" (MITS)

MITS grupas galvenais pētniecības virziens bija rast inovatīvu risinājumu satiksmes drošības un efektivitātes uzlabošanai. Viens no rezultātiem ir šajā dokumentā aprakstītā metode un izveidotais prototips.

Projekta laikā tika izstrādāts modulārs un multifunkcionāls punkts, kas saņem dažādu sensoru datus, veic informācijas sapludināšanu un analīzi, kā arī pieņem lēmumus par potenciālas avārijas situācijas un dažādu konstatēto notikumu nodošanu vadības punktam. Izveidotā metode nodrošina plašu funkcionalitāti: transporta līdzekļu reģistrēšanu uz brauktuves, transporta līdzekļu klasifikāciju, nosaka transporta līdzekļu pārvietošanās ātrumu, atrod un atpazīst transporta līdzekļa reģistrācijas numura zīmes, nosaka iespējamo satiksmes negadījuma esamību, kontrolē krustojuma šķērsošanu pie neatļautā signāla, atpazīst operatīvo transporta līdzekli, nodrošina datu sūtīšanu, izmantojot GSM tehnoloģijas, nodrošina datu saglabāšanu, nosaka laika apstākļus un veic ceļa stāvokļa analīzi.

MITS metodes aprakstā ir sekojošas apakšgrupas:

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

Turpmākajās nodaļās ir aprakstītas izstrādātās metode un tās algoritmi minētajos virzienos, kā ari aprakstīta izveidotā prototipa funkcionalitāte un darbība.

Nodaļa 1

Inteliģentas satiksmes vadības sistēmas arhitektūra

1.1 Ievads

Lai inteliģenta satiksmes vadības sistēma varētu veiksmīgi darboties tai ir jāspēj nodrošināt datu savākšana un analīze, situācijas analīze un lēmumu pieņemšana satiksmes regulēšanai. Visas šīs funkcijas ir jānodrošina bez jūtamas laika aiztures, jo situācija satiksmē ir dinamiska, kā arī droši, jo no notikumiem uz ceļa ir tiešā veidā atkarīga satiksmē iesaistīto cilvēku dzīvība un labsajūta.

Drošums inteliģentas satiksmes vadības sistēmas kontekstā nozīmē, ka:

- Satiksmes vadība notiek atbilstoši satiksmes noteikumiem un drošības normām
- Sistēma pieņem lēmumus balstoties uz vairākām informācijas plūsmām samazinot kļūdainu datu ietekmi
- Pat ja atsevišķas sistēmas sastāvdaļas pārstāj korekti funkcionēt (iesaistīta iekārta tiek izslēgta vai programmatūras modulis pārstāj darboties), sistēma kopumā ir spējīga turpināt darboties ar samazinātu kapaciāti, bet droši.

Lai inteliģentā satiksmes vadības sistēma spētu nodrošināt visas iepriekš minētās prasības ir izstrādāta modulāra un izkliedēta reāla laika sistēmas arhitektūra. Sistēmas modulārums ļauj atsevišķas sistēmas izveides funkcijas, kā piemēram auto numuru atpazīšanu, uzticēt konkrētās jomas speciālistiem, neliekot tiem iepazīties ar visas sistēmas kopējo uzbūvi. Sistēmas izkliedētība nozīmē, ka tā var paralēli darboties uz vairākām fiziskām iekārtām sniedzot gan papildus jaudu datu apstrādei, gan papildus drošību gadījumiem, ja kāda no iekārtām pārstāj funkcionēt. Vēl svarīgs ieguvums no sistēmas izkliedētības ir, ka datus var savākt atsevišķas iekārtas no attālākām ceļa/krustojuma vietām kurās nav pilnvērtīga vadu infrastruktūra un pārsūtīt tos apstrādei, uzticot lielāko jaudu prasošos aprēķinus veikt labāk situētām iekārtām.

Šajā nodaļā tiks aprakstīta šī sistēmas arhitektūra kopumā un turpmākajās nodaļās būs redzami jau specifiski sistēmas moduļi, kas atbildīgi par tādām sistēmas funkcijām kā datu savākšanu un lēmumu pieņemšanu.

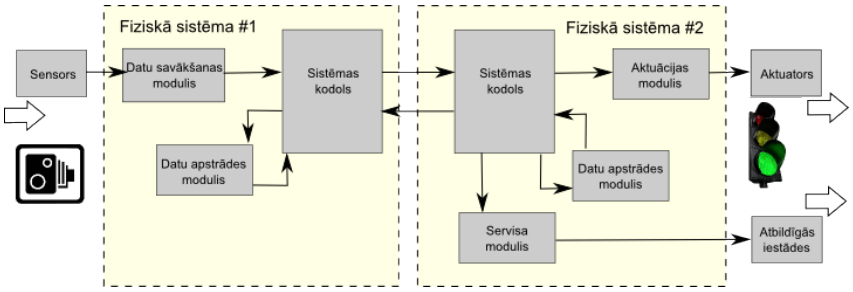
1.2 Arhitektūra

Lai arhitektūru varētu uzskatīt par izkliedētu tai ir jāspēj veikt savus uzdevumus paralēli darbojoties uz vairākām iekārtām, kuras var atrasties tālu viena no otras, un starp kurām komunikācijas kvalitāte ne vienmēr ir garantēta.

Tā kā no satiksmes vadības sistēmas kļūdainas darbības var rasties apdraudējums cilvēku dzīvībai, veselībai un mantai, ir svarīgi lai sistēma veiktu savus uzdevumus drošā veidā pat gadījumos, kad kāda atsevišķa sistēmas daļa tiek bojāta vai atvienota no kopējās sistēmas.

Zemāk aprakstītā sistēmas arhitektūra ir veiksmīgi publicēta šādā rakstā:

- Nesenbergs, K.; Selavo, L., "A distributed data processing architecture for real time intelligent transport systems," Informatics and Applications (ICIA),2013 Second International Conference on , vol., no., pp.155,160, 23-25 Sept. 2013



Att. 1.1: Datu plūsmas shēma starp sistēmas loģiskajām daļām, kā arī to novietojums fiziskajās sistēmās

Lai sistēma varētu nodrošināt dažādu sensoru datu ieguvi, drošu glabāšanu, aprēķinu veikšanu, un lēmumu izpildi izkliedētā vidē noturīgā un drošā veidā sistēmas arhitektūra tika izveidota modulāra, sastāvoša no divām daļām:

- Sistēmas kodols, kas nodrošina sistēmas pamata funkcionalitāti, tādu kā datu glabāšana, rezerves kopiju izveide (replicēšana), sistēmas darba uzmanīšana un noslodzes balansēšana.
- Datu pico ‘apstrādes moduļi, kurus vajadzības gadījumā var izstrādāt trešās puses. Šie moduļi var būtu gan datu avoti, kas nodrošina sensoru mērījumu datus, gan loģiski moduļi, kas izveido jaunus datus no pieejamajiem datiem, vai veic citas darbības.

Lai šāda modulāra sistēma būtu iespējama ir izstrādāts komunikācijas standards kas ļauj moduļiem sarunāties ar sistēmas kodoliem veidā, kas nav atkarīgs no sistēmas fiziskā izvietojuma un konfigurācijas.

Uz katras no izkliedētās sistēmas fiziskajām iekārtām darbojas sistēmas kodols un nenoteikts skaits sistēmas moduļu (Attēls 1.1).

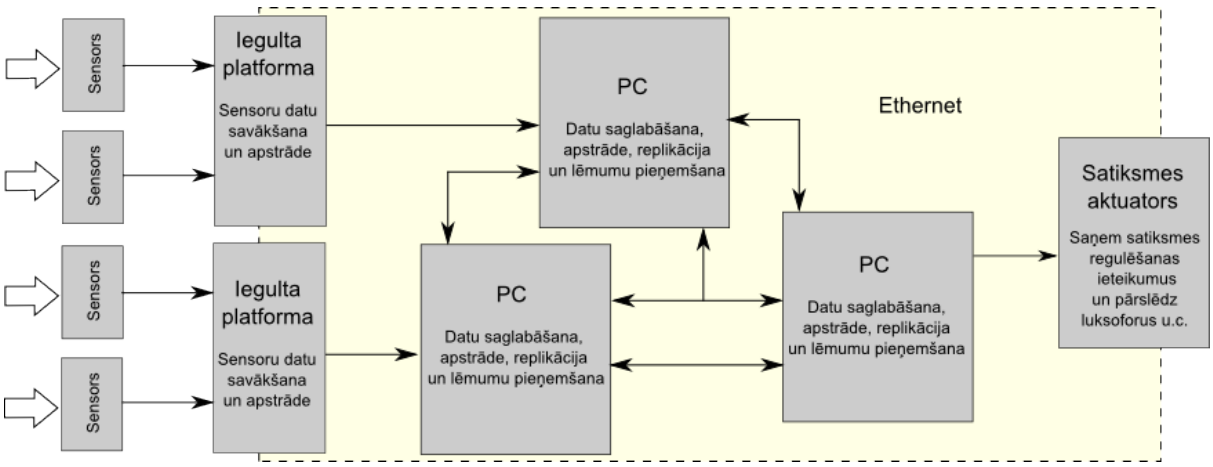
Lai radītu skaidrāku priekštatu par sistēmas kodola nepieciešamo funkcionalitāti vispirms nepieciešams aprakstīt moduļu funkcionalitāti.

1.2.1 Moduļi

Sistēmas moduļi pēc būtības ir funkcija, kas no sistēmas saņem ievaddatus, apstrādā tos un rada jaunus datus, ko nodod sistēmai. Visi moduļi izmanto kopīgu komunikācijas saskarni ar sistēmas kodolu lai saņemtu un nodotu datus. Papildus tam moduļiem var būt pieejami arī citi sistēmas resursi, kā neapstrādāti fiziskā sensora dati vai GSM modēms īsziņu nosūtīšanai.

Daži sistēmas moduļu piemēri, kas varētu būt nepieciešami inteliģentā transporta sistēmā:

- Moduļi, kas savāc datus no datoriem - šādi moduļi var saņemt datus no datora seriālā porta vai kāda cita savienojuma, pa kuru sensori var nodrošināt datu plūsmu. Piemēram, IP kameras datu plūsmu modulis saņem caur datora Ethernet savienojumu. Šādu moduļu uzdevums ir savākt datus un priekšapstradāt tos tādā formātā, kāds nepieciešams sistēmai, pēcāk pārsūtot tos sistēmas kodolam glabāšanai un sistēmas darbības nodrošināšanai.
- Moduļi, kas darbina ārējus aktuatorus - šie moduļi lieto citu moduļu savāktos un apstrādātos datus un kontrolē ārējas ierīces, piemēram, luksoforus vai GSM modēmu, kas sūta īsziņas atbildīgajām iestādēm.
- Datu apstrādes moduļi veic galveno sistēmas loģikas darbu un datu apstrādi. Datu apstrādes moduļi ņem jau sistēmā esošos datus un apstrādājot tos rada jaunus datus. Piemēram, numuru atpazīšanas algoritms ņem kameras uzņemtos kadrus un atrod tajos numura zīmes, kurās tālāk cits modulis var atrast numura zīmju tekstu. Līdzīgi šāds modulis varētu ņemt datus no dažādu sensoru rādījumiem konkrētā laika posmā un radīt datu kopu, kas apzīmē konkrētu detektētu automašīnu noteiktā vietā un noteiktā laikā.
- Servisa moduļi, piemēram tādi, kas saglabā sistēmas stāvokļa informāciju vai sniedz komandrindas pieeju sistēmai.



Att. 1.2: Datu plūsmas shēma starp sistēmas loģiskajām daļām, kā arī to novietojums fiziskajās sistēmās

Visa informācija par izkliedēto satiksmes regulēšanas sistēmu ir paslēpta no moduļiem lai atvieglotu to izstrādi. Šī iemesla dēļ kad modulis tiek iedarbināts tas caur sistēmas saskarni sazinās ar sistēmas kodolu uz vietējās fiziskās iekārtas, un tālāk saņem instrukcijas kā saņemt vai saglabāt nepieciešamos datus. Šāda sākotnējā komunikācija var saturēt tādu informāciju, kā:

- Sistēmas uzstādījumi un statuss, kas nepieciešams lai modulis varētu korekti darboties.
- Informācija, kā sazināties ar kodolu, kas nodrošinās modulim nepieciešamos ieejas datus.
- Informācija, kā sazināties ar kodolu kurā glabāt moduļa izejas datus. Šī informācija laika gaitā var mainīties, ja sistēmas kodoli noslodzes sadalīšanas nolūkos nolemj, ka modulim jāsazinās ar citu fizisko iekārtu/kodolu.

Pēc tam, kad sākotnējā komunikācija ir pabeigta, modulis var lasīt no ienākošās datu plūsmas vai pieprasīt papildus datus no sistēmas, kā arī sūtīt savus darba rezultātus atpakaļ sistēmai. Papildus tam, modulis var saņemt arī kontroles komandas no sistēmas kodola, kas potenciāli var nomainīt saziņas uzstādījumus darbības laikā, lai veidotu vienmērīgu noslodzes sadalījumu sistēmā.

Sīkāka informācija par individuāliem moduļiem atrodama turpmākajās nodaļās.

1.2.2 Sistēmas kodols

Sistēmas kodolam nepieciešams nodrošināt datu plūsmu starp sistēmas loģiskajām daļām, kā tas redzams attēlā 1.2.

Lai paveiktu šo darbu un nodrošinātu sistēmas moduļus ar to darbībai nepieciešamo, sistēmas kodolam ir jāsatur šādas svarīgākās sastāvdaļas:

- Konfigurācijas apakšsistēma
- Datu glabāšanas apakšsistēma
- Starpkodolu tīkla apakšsistēma
- Datu rezerves kopiju apakšsistēma
- Datu pieprasīšanas apakšsistēma

Tālākajās apakšnodaļās katra no šīm apakšsistēmām aplūkota sīkāk.

Konfigurācijas apakšsistēma

Tā kā sistēmai ir nepieciešams spēt atjaunot savu darbību gadījumos, kad kāda no iekārtām vai programmatūras daļām ir pārstājusi darboties, sistēmā vienmēr ir jābūt pieejamai vissvaigākajai konfigurācijai, lai sistēmas darbībai atjaunojoties nerastos darbības problēmas. Konfigurācija, kas jānodrošina kodolam ir iedalāma divās daļās:

- Statiski konfigurācijas dati, kā piemēram, katra kodola un moduļa unikālais identifikācijas numurs, lokālās datubāzes piekļuves informācija, kā arī fiziskais pasaules modelis - uz kuras no brauktuvēm atrodas kādi sensori.
- Dinamiskie dati, kas rodas darbības laikā, piemēram, moduļu un kodolu savienojumu informācija, datu apstrādes starprezultāti u.t.t.

Dinamiskie konfigurācijas dati vienmēr tiek saglabāti tā, lai strāvas zuduma vai citu problēmu gadījumā sistēmas procesi varētu atsākt darbu tik tuvu iepriekšējam stāvoklim cik vien iespējams.

Sistēmas statiskā konfigurācija darba laikā var arī tikt mainīta no ārpuses, ko sistēmai ir jāievēro un jāatjauno savu darbību.

Tā kā datu replikācijas mehānisms, kas aprakstīts zemākā apakšnodaļā izplata visu informāciju pa izkliedēto sistēmu, tad konfigurācijas dati vienmēr tiek izplatīti pa vairākiem kodoliem un glabājas vairākās vietās.

Datu glabāšanas apakšsistēma

Lai sistēma būtu spējīga veikt lēmumus balstītus satiksmes datos, tai ir jāspēj šos datus savākt, apstrādāt un saglabāt, lai sistēmas darba pārtraukumu gadījumā šo darbu varētu turpināt. Katrai no fiziskajām datu apstrādes sistēmām ir lokāla datu bāze, kurā tiek glabāti nepieciešamie dati konkrētajai sistēmai, kā arī replicētie dati no citām sistēmām.

Saglabātie dati var būt jebkāda datu tipa un tiek saglabāti kopā ar meta datiem, kā laika zīmogu un datu avota identifikācijas numuru, kas ir nepieciešams datu iegūšanas apakšsistēmai, kā arī daru replikācijas apakšsistēmai.

Kad datus sistēmas kodolā iesūta kāds no moduļiem, tos uzreiz nesaglabā konkrētais kodols, bet nodod tos datu replikācijas sistēmai apstrādāšanai. Pēc tam šī apakšsistēma nosūta datus saglabāšanai uz tām fiziskajām sistēmām, kuras ir vispiemērotākās konkrēto datu glabāšanai balstoties uz noslodzes izlīdzināšanas un datu apstrādes vajadzībām.

Datu paketes, kas ceļo sistēmā ir iedalāmas divās grupās:

- Datu paketes DATA, kas satur šādus laukus:
  - $SourceID$ - unikāls identifikators modulim, kurš radīja šo datu paketi (Atšķirīgs no sūtītāja - datu paketēm sūtītājs nav svarīgs, jo replikācijas procesā, sūtītājs var būt jebkurš modulis);
  - ID - priekš konkrētā datu avota unikāls identifikators datu paketei (kopā ar $SourceID$ veido unikālu atslēgu pēc kā atpazīt paketi);
  - $Time$ - paketes rašanās laiks (milisekundēs);
  - $Data\_Type$ - identifikators, kas apzīmē datu tipu (skaitlis, attēls, u.t.t.);
  - $Data\_Size$ - skaitlis, kas apzīmē cik baitus aizņem dati;
  - $Data$ - paketes saturs $Data\_Size$ baitu izmērā;
  - $Replication$ - skaitlis (vismaz 4 baiti), kura binārā forma parāda kuros kodolos šī datu pakete ir dublēta. Katram kodolam ir secīgs vesela skaitļa identifikators. Piemēram, šis lauks ir "... 00000110" ja 2. un 3. kodolā šie dati ir saglabāti. (Attiecīgi no mazāk vērtīgā gala skaitot katrs 0 - nav replicēts šajā kodolā, un 1 - ir replicēts šajā kodolā);
  - $ReplicationClock$ - skaitlis - replikācijas pulkstenis - cik kodoliem šī pakete vēl drīkst iet cauri replikācijas gaitā.
- Sistēmas paketes SYS, kas tiek izmantotas komunikācijas nodibināšanai un uzturēšanai un satur šādus laukus:
  - $SenderID$ - unikāls identifikators sistēmas daļai, kas radīja šo paketi;

- *AddressID* - unikāls identifikators sistēmas daļai, kam būtu jāsaņem šī pakete. $AddressID = 0$ nozīmē ka pakete adresēta visiem (broadcast);
- *Type* - skaitlis, kas apzīmē konkrētās sistēmas paketes tipu;
- *Data* - skaitlis, kas satur papildus datus, kas sistēmas paketei tas ir nepieciešams;

No kopējās komunikācijas un konfigurācijas datubāzē tiek saglabāti dati šādā struktūrā:

- Unikāls datu ID;
- Laika zīmogs;
- Brauktuves ID;
- Avota ID (process, sensors u.tml.);
- Datu tips (numurs, auto noteikšana u.c.);
- Paši dati.

Starpkodolu tīkla apakšsistēma

Lai izkliedētā inteliģentā satiksmes vadības sistēma varētu darboties, tai ir nepieciešams spēt novadīt ziņojumus starp dažādām sistēmas daļām. Tas nozīmē, ka ir nepieciešams savākt un saglabāt informāciju par tīklu, kas veidojas starp sistēmas kodoliem uz dažādajām fiziskajām iekārtām.

Kad sistēmas kodols tiek iedarbināts pirmo reizi, tas sūta SYS paketi uz visiem datoriem (broadcast - gan Ethernet līmenī, gan arī SYS paketes adresē, kas vienāda ar 0). Šai paketei tips ir "1", kas nozīmē, ka kodols ir ieslēgts un gaida sūtījumus. Pretī visi pārējie kodoli, kas šo sūtījumu saņem sūta jau precīzi adresētu paketi ar tipu "2", kas nozīmē - informācija saņemta, lūk mana adrese, ko pievienot kodola kontaktu sarakstam. Tāpat katrs kodols pievieno jaunieslēgto kodolu savam kontaktu sarakstam, kurā atrodami visi aktīvie sistēmas kodoli, ieskaitot to adreses identifikatoru, IP adresi, kā arī pēdējo laiku, kad šis kodols sūtījis informāciju (lai varētu noskaidrot, vai kodols nav pārtraucis darboties). Šādā veidā kodols ieslēdzoties iegūst tabulu ar datiem par visiem aktīvajiem kodoliem tīklā.

Papildus tam, lai atvieglotu datu replikāciju un sistēmas noslodzes balansēšanu, kodoli regulāri viens otram nosūta datus par lokālās fiziskās sistēmas noslodzi.

Ja modulim ir nepieciešama piekļuve kādam citam sistēmas kodolam, tā lokālās fiziskās sistēmas kodols var sniegt informāciju par to, kādā adresē atrodams vēlamais kodols, un modulis šo informāciju var izmantot tiešas saziņas nodibināšanai.

Lai moduļi varētu pieslēgties šim sistēmas starpkodolu tīklam un uzzināt nepieciešamo saziņas informāciju tie arī seko noteiktam saziņas protokolam ieslēdzoties:

- Vispirms modulis pieslēdzas Localhost:port (kur ports ir konkrētajā sistēmā izvēlēts brīvs ports, kas ierakstīts konfigurācijas failā) - šajā adresē modulis atrod vietējo fiziskās sistēmas kodolu;
- Caur šo adresi tiek veikts pieprasījums *Data_destination_request*, ja modulis plāno sūtīt savus datus:
  - Šis pieprasījums jautā uz kurieni sūtīt datus ar noteiktu tipu;
  - Kodols vadoties no informācijas par sistēmas noslodzi un fizisko attālumu starp moduļa un mērķa kodola fiziskajām sistēmām izvēlas vienu no sistēmas kodoliem, kas ir atbilstošākais šo datu savākšanai. Atbildē tiek nosūtīta mērķa kodola ārējā adrese, un šim nolūkam specifiski atvērts ports (paralēli starp kodoliem notiek saziņa, ka šāds ports ir jāatver);
  - Pēc tam modulis prasītā tipa datu paketes sūta uz doto adresi.
  - Papildus sistēmā ir paredzēta iespēja, ka Kāds no kodoliem izlemj, ka jāpaziņo datus sūtošajam modulim, ka turpmāk šie dati jāsūta uz citu kodolu (vai nu slodzes izmaiņu dēļ, vai tādēļ, ka konkrētais adresāts pārstājis darboties).
- vai pieprasījums *Data_Request_request*, ja modulis vēlas saņemt datus no sistēmas:
  - Šis pieprasījums jautā, lai tam atsūta konkrētus datus. Šos datus identificē pēc datu avota ID un datu tipa;
  - Pretī modulis sagaida visjaunākos sistēmas datus ar šādu veidu;

– Kodols saņemot šo pieprasījumu noskaidro vai tādi dati sistēmā ir un atbild ar ”OK” (viss kārtībā) vai ”ERR” (kļūda) SYS paketi attiecīgi;
– Pēc tam tas kodols, kam šie dati ir, pa tiešo pieslēdzas modulim uz norādīto adresi/portu (kas ietverti pieprasījuma paketē) un nosūta datus;
– Ir iespējams arī datu pieprasījums straumēšanas formātā - tādā gadījumā vienmēr ienākot jauniem šāda tipa datiem tos arī sūtīs pieprasītājam.

Datu rezerves kopiju apakšsistēma

Šī kopā ar datu pieprasīšanas apakšsistēmu ir sarežģītākās un svarīgākās izkliedētās inteliģentās transporta sistēmas arhitektūras daļas. Dati, kas tiek glabāti sistēmā ir jāglabā vairākās fiziskās vietās vienlaicīgi un procesiem, kas vēlas tiem piekļūt ir jāspēj pieslēgties tai sistēmas daļai kurā atrodami visaktuālākie specifiskie dati.

Datu pakete, šim uzdevumam jau ir īsumā aprakstīta datu glabāšanas apakšsistēmas apakšnodaļā.

Kad datu pakete pirmo reizi tiek izveidota, tā tiek piepildīta ar atbilstošo informāciju un replikācijas datu lauks tiek atstāts tukšs. The replikācijas pulksteņa lauks parasti tiek uzstādīts kā skaitlis $N \in [2; M]$, kur M ir kopējais kodolu skaits sistēmā.

Kā iepriekš minēts, katrs no sistēmas kodoliem regulāri aprēķina savu sistēmas noslodzi (ko mēra %) un izplata šo informāciju visiem kodoliem.

Kad datu pakete ierodas pie kodola, tas veic šādus soļus:

1. Replikācijas datu laukā tiek pārbaudīts cik kodolos CR šie konkrētie dati jau ir replicēti. Šis skaitlis tiek salīdzināts ar globālajā konfigurācijā norādīto skaitli CR_MIN, kas norāda cik kodolos kā minimums nepieciešams noglabāt katras datu paketes datus;

2. Ja $CR \geq CR_MIN$ pakete tiek izmesta un netiek tālāk apstrādāta. Pretējā gadījumā tiek pāriets uz nākamo soli;

3. Tiek izvērtēta iespēja nesaglabāt šo paketi lokālajā fiziskajā sistēmā. Tā ir atkarīga no replikācijas pulksteņa skaitļa CLK paketē. Ja $CLK \leq CR_MIN - CR$ šī pakete ir obligāti jāsaglabā lokālajā fiziskajā sistēmā, lai nodrošinātu, ka tā tiek saglabāta vismaz CR_MIN reizes. Pretējā gadījumā sistēmas kodols var izlemt saglabāt šo paketi lokālajā sistēmā, ja tā sistēmas noslodze ir pietiekami zema (konkrēts līmenis ir norādīts sistēmas iestatījumos) un tam ir pietiekoši daudz brīvas vietas datu glabāšanai. Papildus tam, kodolam ir lielāka varbūtība noglabāt savā lokālajā sistēmā datus no tāda avota un ar tādu datu tipu, kādi jau glabājas šā kodola sistēmā - tas samazina biežumu, ar kādu datus patērējošajiem moduļiem būtu jāpārslēdzas no viena datu piegādātāja uz citu, ja tam interesē konkrēts datu veids, kā vairumā gadījumu;

4. Ja tiek lemts datus saglabāt lokālajā fiziskajā sistēmā, replikācijas datu laukā paketei tiek atzīmēts konkrētā kodola bits ar ”1” un tad šī pakete tiek saglabāta konkrētās fiziskās sistēmas kodola datu saglabāšanas apakšsistēmā;

5. Pēc tam replikācijas pulksteņa lauka skaitlis tiek samazināts par 1.

6. Ja jaunais $CLK > 0$, un jaunais $CR < CR_MIN$ pakete tiek nosūtīta uz nākamo kodolu. Pretējā gadījumā pakete netiek pārsūtīta tālāk. Nākamais paketes galamērķis tiek izvēlēts starp tiem kodoliem, kuri vēl nav atzīmēti replikācijas datu laukā, sākot ar to, kuram ir vismazākā sistēmas noslodze.

Tas nozīmē, ka katra pakete ar garantiju tiks replicēta CR_MIN sistēmas kodolos. Papildus tam katram no sistēmas kodoliem ir neliela brīvība pieņemt paketes vai nosūtīt tās tālāk citiem kodoliem, šādā veidā veicot slodzes sadalīšanu. Replikācijas pulksteņa lauks garantē, ka neviena pakete neceļo sistēmā pārāk ilgi.

Papildus iepriekšējiem soļiem katrā brīdī, kad spriežot pēc citu kodolu pēdējā aktivitātes laikā izskatās, ka kāds no kodoliem ir pārstājis darboties, citi kodoli par to paziņo viens otram ar SYS paketēm, un šīs informācijas saņēmēji ņem tās paketes, no savas datubāzes, kuras bija arī izslēgtajā kodolā un izvēlas vai konkrēto paketi nav nepieciešams replicēt vēlreiz. Tā kā izslēdzoties konkrētam kodolam katru tā paketi nepieciešams replicēt tikai vienu papildus reizi, tad tas aktīvais kodols, no visiem kodoliem pie kā replicēta šī pakete, kuram ir vismazākais ID numurs no visiem iesaistītajiem pārsūta šo paketi vēl vienai replicēšanai (iepriekš izņemot no replikācijas datu lauka izslēgto kodolu). Šādā veidā nevienu paketi paralēli nemēģina replicēt vairāki sistēmas kodoli.

Datu pieprasīšanas apakšsistēma

Kad dati ir veiksmīgi replicēti un saglabāti vairāku kodolu datu glabāšanas apakšsistēmā, ir svarīgi nodrošināt vissvaigākos un visātrāk pieejamos konkrēto datu paraugus tiem moduļiem, kuri tos vēlas izmantot. Tas ir datu pieprasīšanas apakšsistēmas uzdevums.

Ja modulim ir nepieciešami konkrēti dati, tas vispirms vēršas ar datu pieprasījumu pie lokālā sistēmas kodola (*Data_Request_request*, kas aprakstīts sistēmas stapkodolu tīkla apakšsistēmā). Lokālais sistēmas kodols tālāk pārraida šo pieprasījuma informāciju visiem pārējiem kodoliem un visi kodoli, kas ir pie sevis saglabājuši šāda datu tipa un datu avota paketes atbild uz šo pieprasījumu ar šāda veida jaunākās saglabātās paketes identifikatoru. Balstoties uz šīm atbildēm lokālais sistēmas kodols atbild modulim ar adresi kodolam, kurš glabā visjaunāko šāda tipa un avota paketi tīklā, un tajā pašā laikā mērķa kodolam tiek dota ziņa, ka tam ir jāatver ports un jāgaida pieprasījums pēc konkrētajiem datiem.

Precizējot moduļiem ir pieejamas divas datu pieprasīšanas metodes, ko tie var izmantot ērtākai piekļuvei datiem:

1. *Current_Timestamp*() - šī metode atgriež pašreizējo sistēmas laika zīmogu, par ko sistēmas kodoli kopīgi vienojas sinhronizējot laiku.
2. *Request_Data*() - metode konkrētu datu pieprasīšanai ar šādiem parametriem:
   - T1 - laika zīmogu intervāla sākums;
   - T2 - laika zīmogu intervāla beigas (atbilde ir tikai par datiem intervālā [T1;T2];
   - Datu tips - tas pats, kas datu bāzes lauks - atgriež tikai šo datu tipu;
   - Default - būla vērtība, ja True un laika intervālā T1;T2 nebija prasīto datu, atgriež svaigākos datus kas ir, ja nav ko atgriezt vai šis lauks ir False tad atgriež NULL (datu nav);
   - Brauktuve - brauktuves ID par kuru interesē dati;
   - Avots - Avota ID, ja NULL tad jebkurš avots no šīs brauktuves;

   Šī metode cenšas atgriezt svaigākos datus konkrētajā laika intervālā par konkrētas brauktuves konkrētiem avotiem.

Modulis ar šīm metodēm spēj pieprasīt datus un saņemt vissvaigāko datu avotu tīklā. Atkarībā no moduļa vajadzībā, tas vai nu pieprasa tikai vienu paketi vai datu straumēšanu. Ja tiek pieprasīta tikai viena pakete, tad kodols, kas atbildīgs par pieprasītajiem datiem nosūta konkrēto datu paketi un komunikācija tiek pārtraukta. Ja tiek pieprasīta datu straumēšana, tad kodols atver datu straumi un informē modeli par straumes atrašanās vietu. Tad modulis brīvi piekļūst datu straumei un kodols ir atbildīgs par to lai datu straumē atrastos dati. Ja pienāk brīdis, kad kodolam vairs nav pieejami svaigākie šī veida dati, šis kodols pats nokārto jauna datu avota atrašanu un nosūta informāciju kodolam, ka tam ir jāmaina datu avota informācija lai piekļūtu jaunākajiem datiem no cita sistēmas kodola.

Tas nozīmē, ka visās izkliedētās sistēmas vietās, kurās ir nepieciešami svaigi dati, datu ieguves sistēma nodrošinās vis svaigākās datu paketes tā, lai moduļu veidotājiem nebūtu par to jāuztraucas.

1.2.3 Sistēmas demonstratora prototips un lietotāja saskarne

Sistēmas iespēju novērtēšanai tika izveidots demonstratora prototips, kā arī lietotāja saskarne.

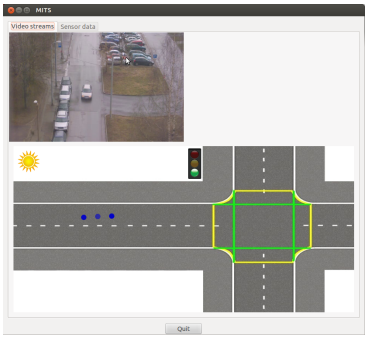
Sistēmas demonstratora prototips sastāv no šādām fiziskām daļām (Attēls 1.3):

- trīs mini-ITX formas skaitļošanas moduļiem aprīkotiem ar 3.4 GHz Intel core i5 procesoru, 8 GB RAM, kā arī vadu un bezvadu tīkla savienojumiem;
- astoņiem Raspberry PI B iegultajiem skaitļošanas moduļiem aprīkotiem ar 700 MHz ARM 11 procesoru un 512 MB RAM, kā arī ar bezvadu tīkla savienojumu;
- Bezvadu tīkla maršrutētāja, kas atbalsta standartus 802.11g/n;
- Divām tīkla kamerām AXIS P1344, kas izmantotas video analīzei un numura zīmju atpazīšanai;
- Rezonanses kontūra ceļu stāvokļa noteikšanai;
- Sensora laika apstākļu noteikšanai (temperatūra, mitrums, vēja virziens un stiprums, nokrišņu daudzums);

Att. 1.3: Sistēmas prototipa fiziskās sastāvdaļas

- Akustiskais sensors;
- Lāzera attāluma mērīšanas sensors;
- Siltuma sensori (PIR);
- Magnētiskie sensori;

Sistēmas demonstratora prototipa datoros ir Linux operētājsistēma, kas ne tikai samazina prototipa izmaksas, bet arī uzlabo sistēmas drošību. Uz katra no datoriem, ieskaitot iegultos datorus darbojas sistēmas kodols. Raspberry PI datori tiek lietoti sensoru datu attālinātai savākšanai no tālākām krustojuma vietām, un tālāk pa bezvadu savienojumu nogādā šos datus jaudīgākajiem datoriem sistēmā apstrādei.



Att. 1.4: Sistēmas lietotāja saskarne

Lai demonstratora testēšana nebūtu atkarīga no atļauju iegūšanas pieslēgšanai pie īsta luksofora reālā krustojumā, kas ļoti apgrūtinātu sistēmas izstrādi, tika izveidota sistēmas lietotāja saskarne, kurā virtuāli tiek attēloti sistēmas pieņemtie lēmumi un luksofora signāli (Attēls 1.4).

Šajā saskarnē ir pieejami divi skati:

- Neapstrādāto datu skats un
- Sistēmas pieņemto lēmumu skats

Neapstrādāto datu skatā ir iespējams sistēmas darbības laikā aplūkot datus no dažādiem sistēmas sensoriem, kas ļauj novērtēt vai sistēma darbojas korekti. Sistēmas pieņemto lēmumu skatā redzams, kā sistēma ir novērtējusi kopējos ceļa apstākļus, atpazinusi automašīnas un pieņēmusi lēmumu par satiksmes regulēšanu.

Nodaļa 2

Inteliģentas satiksmes vadības loģikas modulis un algoritms

2.1 Ievads

Šajā nodaļā aprakstīts sistēmas modulis, kas balstoties uz datiem, ko sieguvuši sistēmas sensori un datu apstrādes moduļi pieņem lēmumus satiksmes regulēšanā konkrētā krustojumā.

2.2 Lēmumu pieņemšanas algoritms

Lai noteiktu kā regulēt satiksmi noteiktā krustojumā sistēmai ir nepieciešams zināt krustojuma sākotnējo konfigurāciju, tas nozīmē, ka sistēmas konfigurācijā tiek norādīts kura no brauktuvēm atbilstoši satiksmes noteikumiem ir galvenais ceļš. Tā pat, kā jau minēts, visiem sensoru datiem sistēmā ir informācija par to, no kuras brauktuves tie ir iegūti.

Tā kā ne vienmēr apstākļi ir pietiekami lai automātiskā sistēma droši pieņemtu lēmumus par satiksmes vadību, tad gadījumos, kad pieņemt drošu lēmumu nav iespējams sistēma pārslēdz luksoforus noklusētajā laika sadalījuma režīmā, kādā tie darbojas bez inteliģentās vadības sistēmas iejaukšanās.

Šī sistēmas moduļa ievaddati ir visu pieejamo sensoru dati, kā arī dati par to, kādi šobrīd ir luksoforos degošie signāli katrā no braukšanas virzieniem krustojumā.

Sistēmas moduļa izvaddati ir vēlamais luksoforu stāvoklis - dubultai drošībai šo vēlamo stāvokli pārbauda luksoforos iebūvētie drošības kontroles mehānismi un korektuma gadījumā (nav konfliktējoši signāli, signālu pārslēgšana nav pārāk strauja) veic signālu atbilstošu pārslēgšanu. Pat gadījumos, ja sistēma pārtrauc darboties luksoforu kontrolieri paši spēj nodrošināt sadalītā laika regulēšanas mehānismu tā kā tas notiek jau tagad krustojumos bez inteliģenta satiksmes vadības mezgla.

Satiksmes regulēšanas algoritms divu ceļu krustojumā ir šāds:

1. No meteoroloģiskajiem sensoriem nosaka vai laika apstākļi ir piemēroti satiksmes regulēšanai balstoties uz video analīzi - tas nozīmē:
   (a) Ir diennakts gaišais laiks
   (b) Nav miglas, spēcīga lietus vai citu redzamību apgrūtinošu apstākļu
2. Gadījumā, ja video analīze nesniegs vēlamos rezultātus sistēma pāriet analīzes režīmā - savāc datus no pieejamajiem sensoriem vēlākai analīzei un brīdina par bīstamām situācijām, tomēr tiešu satiksmes regulēšanu neveic dēļ nepietiekamas informācijas drošai satiksmes regulēšanai - tiek ieslēgts noklusētais laika sadalījuma režīms.
3. Pretējā gadījumā - ja video analīze ir iespējama, sistēma balstoties uz kameru video analīzi nosaka automašīnu skaitu katrā no brauktuvēm (video analīzes algoritms aprakstīts turpmākā nodaļā)
4. Pēc noklusējuma zaļā gaisma vienmēr ir galvenā ceļa virzienā
5. Ja uz mazākas nozīmes šķērsojošās brauktuves tiek noteikts auto, atkarībā no situācijas tiek izvēlēta viena no šīm darbībām:

(a) Ja galvenā brauktuve ir brīva, mazākās nozīmes brauktuvei tiek dota zaļā gaisma nekavējoties palaižot piebraukušo transporta līdekli

(b) Gadījumā, ka galvenā brauktuve ir aizņemta tiek gaidīts līdz izpildās viens no trijiem kritērijiem, lai ieslēgtu zaļo gaismu mazākas nozīmes brauktuvei apturot satiksmi uz galvenā ceļa:

- Kopš pirmā auto noteikšanas uz mazākās nozīmes ceļa ir pagājušas 3 minūtes (Laiks uzstādāms atkarībā no konkrētā krustojuma specifikas)
- Uz mazākās nozīmes ceļa stāv jau vismaz 3 (skaits maināms atkarībā no konkrētā krustojuma specifikas) transporta līdzekļi
- Galvenais ceļš tiek atbrīvots

6. Kad automašīna no mazākās nozīmes ceļa ir izbraukusi sistēma atjauno zaļo gaismu galvenā ceļa braukšanas virzienā.

7. Gadījumos, kad no mazākās nozīmes ceļa ir nepārtraukta transporta līdzekļu plūsma, šo plūsmu aptur pēc 1 minūtes, vai cita laika posma atkarībā no konkrētā krustojuma specifikas.

8. Sistēma atgriežas sākotnējā stāvoklī un algoritms atsākas.

9. Izņēmuma gadījums algoritmā - ja kādā brīdī uz kādas no brauktuvēm tiek noteikta operatīvā transporta tuvošanās (izmantojot skaņas analīzi no mikrofonu matricas, par ko aprakstīts tālākā nodaļā), abos krustojuma virzienos tiek ieslēgta sarkanā gaisma apturot satiksmi krustojumā un ļaujot operatīvajam transportam krustojumu šķērsot bez starpgadījumiem (operatīvais transports drīkst šķērsot arī luksofora sarkano signālu).

Nodaļa 3

Multimodālu sensoru izmantošana transporta līdzekļu reģistrācijai un klasifikācijai

3.1 Motivācija

Vispilnīgāko informāciju par satiksmes plūsmu iespējams iegūt, izmantojot video kameru un datu video apstrādi. Izmantojot šo risinājumu iespējams iegūt gan informāciju par transporta līdzekļu skaitu uz brauktuves, gan to tipu, ātrumu un citus interesējošos parametrus. Taču video apstrāde labus un teicamus rezultātus dod tikai diennakts gaišajā laikā – labas redzamības apstākļos. Piemēram tumsā, izmantojot video kameras datus, iespējams veikt tikai transporta līdzekļu skaitīšanu un aptuvenu ātruma noteikšanu. Intensīva sniega, miglas vai lietu gadījumā datu iegūšana par satiksmes intensitāti un veidu jau var būt stipri apgrūtināta, bet gadījumos, ja video kameras objektīvs nosarmo, datu iegūšana kļūst neiespējama. Lai nodrošinātu sistēmas darbību jebkuros laika apstākļos, kā arī jebkurā diennakts laikā, jāizmanto citi sensoru veidi. Labs risinājums ir multimodālu sensoru izmantošana, t.i. viena mērķa sasniegšanai izmanto vairākus sensoru veidus (video kameras, magnētiskie sensori, siltuma sensori, lāzera sensori (radari), akustiskie sensori). Šāds risinājums ļauj veikt uzdevumu auto detektēšanu un klasificēšanu, praktiski jebkuros laika apstākļos.

3.2 Ievads

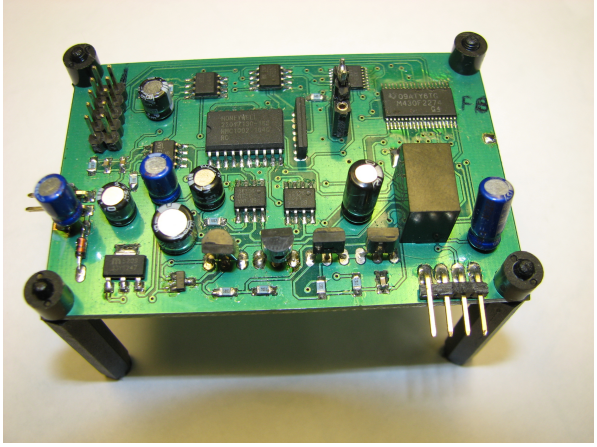
Par galveno projekta mērķi tika uzstādīts uzdevums veikt transporta līdzekļu reģistrēšanu un klasifikāciju izmantojot dažādus pret ceļa segumu neinvazīvus sensorus, kā arī, lai izveidotais risinājums būtu mobils un tā apkalpošana neprasītu satiksmes plūsmas ierobežošanu. Klasiskā variantā transporta līdzekļu detektēšanai izmanto induktīvās cilpas, kuras tiek iestrādātas ceļa segumā. Šāds risinājums dod labu rezultātu transporta līdzekļu detektēšanā, taču ir invazīvs pret ceļa segumu. Induktīvās cilpas dzīves laiks ir īss. Dēļ ceļa seguma deformācijām vadītājs bieži tiek pārrauts un pārstāj veikt savu funkciju. Jaunas induktīvās cilpas iestrādāšanai atkārtoti ir jābojā asfalta segumus, kā arī jāaptur satiksmes plūsma. Modernāks risinājums ir ceļa segumā iestrādājama magnētisko sensoru izmantošana. Arī ši risinājums ir invazīvs pret ceļa segumu.

3.3 Magnētiskā sensora mote

3.3.1 Magnētisko sensoru izmantošana transporta līdzekļu detektēšanā un klasifikācijā.

Transporta līdzekļa radītais magnētiskais lauks ir ļoti neliels un to tikpat kā nav iespējams fiksēt ar piedāvāto metodiku. Transporta līdzeklī magnētisko lauku rada dažādas elektroiekārtas un elektromotori. Transporta līdzeklis uz brauktuves tiek noteikts ar netiešo atklāšanas metodi – mērot Zemes magnētiskā lauka intensitāti. Metālisks objekts, kāds arī ir jebkurš transporta līdzeklis, Zemes magnētiskajā laukā rada lokālus kropļojumus, kurus iespējams izmērīt. Šos lokālos kropļojumus iespējams konstatēt

izmantojot dažāda tipa magnētiskā lauka mērītājus. Projektā tika pieņemts lēmums izmantot anizotropiskos magnetorezistīvos (AMR) magnētiskos sensorus. Šī tipa sensoru priekšrocība ir mazi izmēri un augsta magnētiskā lauka izmaiņu jūtība. Transporta līdzekļa radīto Zemes magnētiskā lauka kropļojumu literatūrā pieņemts saukt par transporta līdzekļa magnētisko parakstu. Pēc vairāku eksperimentu veikšanas tika pieņemts lēmumus, ka transporta līdzekļu atklāšanai jāizmanto magnētiskie sensori, kuri novietoti brauktuves malā. Šāds risinājums ir mobils, kā arī neinvazīvs pret ceļa segumu. Vairumā gadījumu magnētiskie sensori tiek izmantoti kā invazīvs risinājums – iestrādāti ceļa segumā. Izveidotais risinājums ļauj magnētiskos sensorus novietot brauktuves malā 1,5m attālumā no ceļa seguma un aptuveni 2 – 2,5 attālumā no braucošiem transporta līdzekļiem. Izveidotās sensora motes demonstrators redzams 3.1 attēlā. Uzskatāmības dēļ attēlā nav iekļauta bezvadu datu pārraides daļa. Mote ietver gan analogo gan ciparu elektronikas daļu. Daļa no izveidotās sistēmas aprēķiniem tiek realizēta uz mini datora Raspberry pi, kurš savāc datus no viena sensoru mezgla un veic to analīzi, kā arī veic transporta līdzekļu klasifikāciju. Dati par



Att. 3.1: Magnētiskā sensora mote

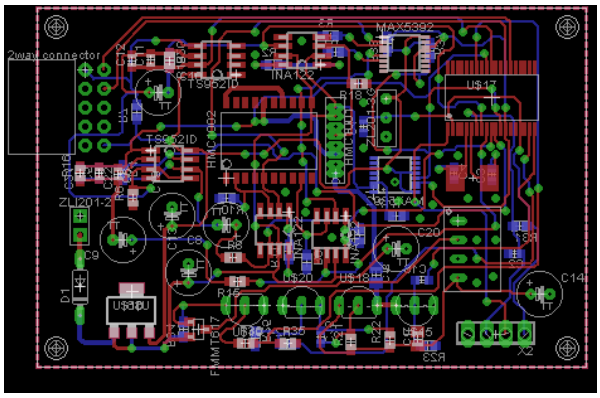
reģistrētajiem transporta līdzekļiem reālā laikā tiek nosūtīti uz MITS punktu, kurš jau pieņemt tālākos lēmumus attiecībā uz satiksmes regulēšanu un transporta līdzekļu skaitīšanu.

3.3.2 Tehniskais risinājums un darbības princips.

Izveidotā magnētiskā sensora mote ietver vairākas daļas (komponentes): - analogie magnētiskā lauka sensori HMC 1001 un HMC 1002

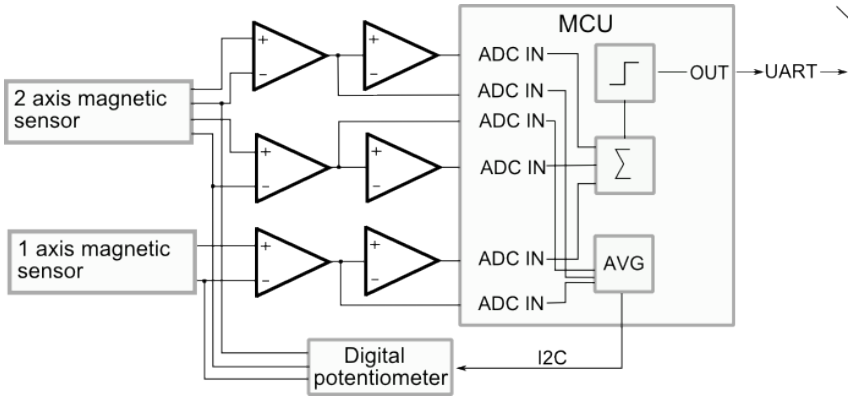
- divu līmeņu pastiprinājuma kaskāde
- atmagnetizācijas kaskāde
- signāla līdzkomponentes uzturēšanas kaskāde
- mikrokontrolieris msp430f2274
- datu bezvadu raidītāja kaskāde

Izveidotās magnētiskā sensora motes PCB redzams 3.2 attēlā. Katra no daļām veic konkrētu uzdevumu, kas nodrošina magnētiskās motes darbību. Abiem analogajiem magnētiskajiem sensoriem HMC1001/1002 ir diferenciālās izejas. HMC1001 ir vienas ass magnētiskais sensors, HMC1002 divu asu magnētiskais sensors. Izmantojot visas trīs asis, tiek noklāts viss telpas apgabals magnētiskā lauka mērīšanai, līdz ar to sensora orientācija attiecībā pret kustīgu objektu nav svarīga. Motes hardware darbības principa shematisks attēlojums redzams 3.3 attēlā. Izejas signāla pastiprināšanai tiek izmantoti instrumentālie pastiprinātāji INA 122U. Katrai izejas kaskādei tiek izmantots savs instrumentālais pastiprinātājs.Pēc pirmās pastiprinājuma kaskādes signālam ir jāuzrauga sprieguma vērtība. Lai pastiprinājuma kaskādes darbotos korekti, sprieguma vērtībai pēc instrumentālā pastiprinātāja jābūt robežās no 0,8 – 1,4 V. Katra pastiprinājuma kaskāde tiek uzraudzīta atsevišķi, neatkarīgi no citas. Sprieguma pastiprinājums pēc pirmās kaskāds ir aptuveni 400 reizes. Ideālā gadījumā magnētiskā sensora izejas signāli (diferenciālā izeja) ir vienādi pēc sprieguma amplitūdas. Reālās iekārtās izejas signāli ir ar nelielu sprieguma atšķirību, kā rezultātā pēc diferenciālā pastiprinājuma veikšanas signāls var “izbraukt” no darba diapazona. Lai šādu situāciju novērstu, motē ir iekļauta signāla līdzkomponentes uzturēšanas kaskāde. Kaskāde ietver ciparu potenciometru MAX5392LAUE+, kurš tiek vadīts ar



Att. 3.2: Magnētiskā sensora motes PCB

mikrokontroliera palīdzību, izmantojot I2C saskarni. Pēc pirmās pastiprinājuma kaskādes signāls tiek kontrolēts izmantojot mikrokontrolieri (MCU). MCU ik pēc noteikta laika intervāla kontrolē sprieguma līmeni. Ja sprieguma līmenis attiecīgā kanālā vairākos kontroles laika momentos ir lielāks vai mazāks par darba diapazonu, ciparu potenciometram MAX5392LAUE+ tiek dota komanda attiecīgi palielināt vai samazināt rezistora lielumu. Izmantotais algoritms ļauj uzturēt sprieguma vidējo vērtību neņemot vērā transporta līdzekļu radītās svārstības signālā (nodrošina izveidotais algoritms), kā arī atmagnetizācijas impulsu radītos izsvārstījumus. Otrā pastipinājuma kaskāde ietver invertējošu pastiprinātāja slēgumu (operacionālais pastiprinātājs TS952). Kopējais signāla pastiprinājums ir 11950 reizes. Otrajā pastiprinājuma kaskādē signāls tiek filtrēts ar augsto un zemo frekvenču filtru. Signāla filtrēšana nodrošina šauras joslas, kurā koncentrēta derīgā signāla enerģija, izdalīšanu no visa frekvenču spektra. Filtrācijas galvenais uzdevums ir nodrošināt minimālu trokšņu klātbūtni tālākā signāla apstrādē. Pēc otrās pastiprinājuma kaskādes izejas signāls tiek tiek ciparots ar 208 Hz frekvenci. Signāla ciparošana tiek veikta izmantojot mikrokontrolieri, kur arī notiek tālāka apstrāde. Minētā ciparošanas frekvence tiek nodrošināta katram kanālam secīgi. Signāla ciparošanas frekvence ir izvēlēta ņemot vērā transporta līdzekļa kustības ātrumu, kas tiek pieņemts 25 m/s. Ja transporta līdzekļa



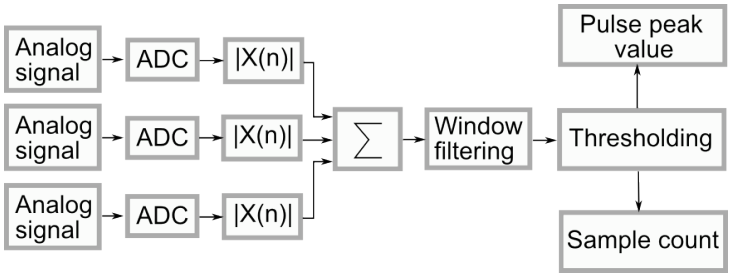
Att. 3.3: Magnētiskā sensora motes darbības principiālā shēma

garums ir 5 m, tiešā sensora tuvumā pavadītais laiks sastāda 0,2 sekundes, kas pie augstāk minētās ciparošanas frekvences ir aptuveni 41 signāla nolase. Realitātē nolašu skaits ir lielāks, jo lokālā Zemes magnētiskā lauka izmaiņa transporta līdzekļa kustības rezultātā notiek jau 2 metrus pirms (atkarīgs no transporta līdzekļa lieluma, kā arī attāluma no magnētiskā sensora motes) transporta līdzeklis nokļuvis ģeometriskā motes tuvumā, kā arī magnētiskā lauka izmaiņa konstatējama arī īsu laiku pēc transporta līdzekļa izbraukšanas no sensora ģeometriskā redzes lauka. Šo faktoru rezultātā reālais transporta līdzekļa impulsu raksturojošais nolašu skaits ir aptuveni 100 nolases, gadījumā, ja transporta līdzekļa garums ir 5 metri un tas kustas ar ātrumu 25 m/s. Šāds nolašu skaits ir pilnībā pietiekams, lai veiktu signāla apstrādi un detektētu transporta līdzekli, kā arī veiktu to klasifikāciju. Pilnvērtīgai analogo magnētisko sensoru HMC1001/HMC1002 darbības nodrošināšanai nepieciešams

veikt atmagnetizāciju. Šī darbība nepieciešama, jo magnētiskie sensori laika gaitā magnētisku objektu vai magnētiskā lauka iedarbībā magnetizējas, iegūstot konkrētu domēnu magnetizācijas virzienu, kas savukārt rada magnētiskā sensora jūtības samazināšanos. Atmagnetizāciju iespējams nodrošināt izmantojot palielināta sprieguma impulsus. Impulsa laiks ir 3 us, bet strāva līdz 3A, bet spriegums 15V. Šādu impulsu formēšanai tiek izmantots mikrokontrolieris, uz tranzistoriem balstīta ķēde un sprieguma paaugstinātājs. Atmagnetizācijas impulss uz magnētisko sensoru tiek padots tūlīt pēc motes ieslēgšanas (pirms sprieguma vidējās vērtības pēc pirmās pastiprinājuma kaskādes kalibrācijas veikšanas) un ik pēc 400 sekundēm. Mikrokontrolieris pilnībā nodrošina magnētiskā sensora motes darbību – veic signālu apstādi, nosakot transporta līdzekļu magnētiskā paraksta parametrus, veic signālu ciparošanu, nodrošina atmagetizācijas kaskādes darbību, kā arī ciparu potenciometra vadību un signāla vidējās vērtības analīzi datu. Mikrokontrolieris nodrošina arī datu pārsūtīšanu uz bezvadu datu pārraides moduli nordic NRF9E5.

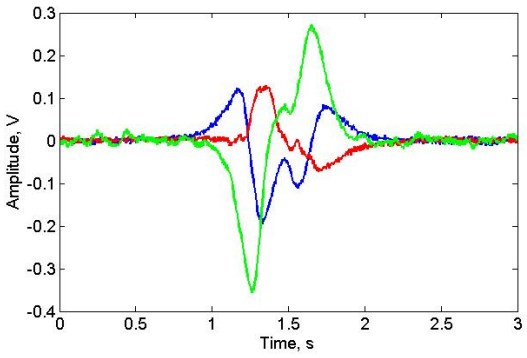
3.3.3 Signālu apstrādes algoritms, realizācija uz motes

Veicot eksperimentālo izstrādi, tika konstatēts, ka efektīvākais veids transporta līdzekļu reģistrācijai ir signāla laika – frekvences analīze un jaudas spektrālā blīvuma noteikšana. Ņemot vērā, ka galvenais uzdevums bija algoritma realizācija zema energopatēriņa iekārtā (mikrokontrolierī), minētais risinājums izrādījās neoptimāls, tapēc bija nepieciešamība izveidot jaunu, resursus maztērējošu algoritmu. Lai veiktu transporta līdzekļu reģistrāciju un klasifikāciju izmantojot ceļa malā novietojamos magnētiskos sensorus, algoritms ir sadalīts divās daļās: viena daļa realizēta mikrokontrolierī, otra daļa mini datorā Raspbery Pi. Mikrokontrolierī realizētā algoritma shematiskas attēlojums redzams 3.4 attēlā. Pēc analogo



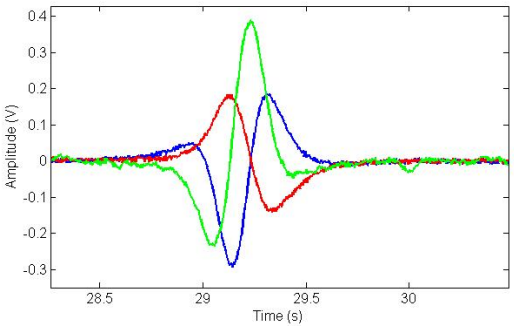
Att. 3.4: Algoritma shematiskas attēlojums

signālu ciparošanas, tiek veikta moduļa operācija, pirms kuras tiek atņemta signāla līdzkomponentes vērtība. 3.5 un 3.6 attēlos redzams transporta līdzekļa magnētiskais paraksts visās trīs asīs pēc signāla

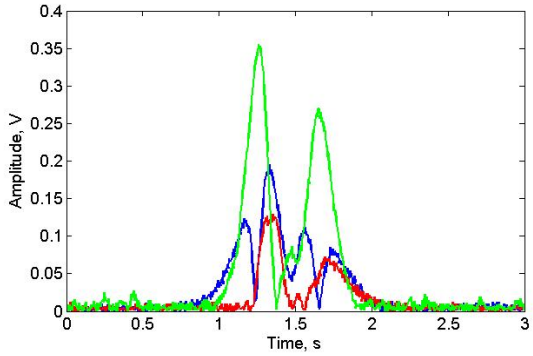


Att. 3.5: Vieglā transporta līdzekļa ar piekabi magnētikskais paraksts trijās asīs

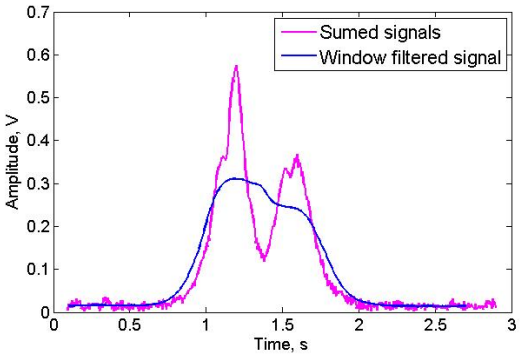
filtrēšanas. **??** attēlā redzami transporta līdzekļa magnētiskais paraksts trijās asīs pēc moduļa operācijas veikšanas. Nākošais solis ir visu trīs kanālu summēšana. Šis etaps izslēdz iespēju vienas no magnētisko asu nobīdes gadījumā transporta līdzekļa reģistrēšanas izslēgšanu. Otrs, iemesls signālu summēšanas operācijai ir magnētiskā paraksta svārstīgums smagā transporta līdzekļa gadījumā vai vieglā transporta līdzekļa ar piekabi gadījumā 3.5. Veicot signāla summēšanu (3.8 un 3.9 attēli), daļa no



Att. 3.6: Vieglā transporta līdzekļa magnētiskais paraksts trijās asīs

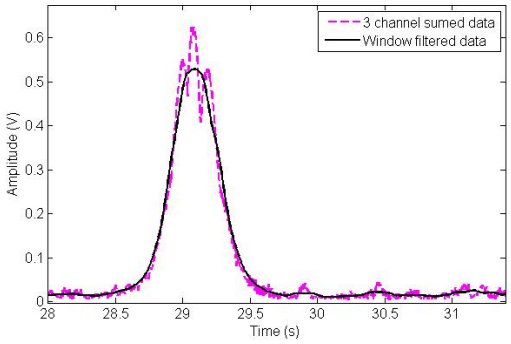


Att. 3.7: Vieglā transporta līdzekļa ar piekabi magnētiskais paraksts pēc moduļa operācijas veikšanas



Att. 3.8: Vieglā transporta līdzekļa ar piekabi magnētiskais paraksts pēc summēšanas trijās asīs un filtrācijas veikšanas

svārstībām tiek nolīdzinātas. Lai vēl vairāk samazinātu trokšņu un transporta līdzekļa magnētiskā paraksta svārstīgumu dažādu transporta līdzekļu (smagie transporta līdzekļi, to piekabes) un to vedamās kravas gadījumā, tiek izmantota loga filtrācija. Loga filtra garums ir 40 nolases. Loga filtrācijas negatīvā puse ir signāla amplitūdas samazināšanās, taču tas ir visiem signāliem proporcionāli vienāds lielums un parametru lielumu neietekmē. Tālākā signāla analīze tiek veikta tai nolasēm, kuras ir virs noteikta sliekšņa. Visas nolase, kuras ir zem sliekšņa, netiek apstrādātas. Ja signāla vērtība pārsniedz sliekšņa vērtību, tiek skaitīts nolašu skaits, kā arī noteikta maksimālā vērtība. Šīs divas vērtības (nolašu skaits virs sliekšņa un maksimālā vērtība) pilnībā raksturo transporta līdzekli, kurš atrodas uz brauktuves. Signāla apstrādē slieksnis ir nepieciešams, jo Zemes magnētiskā lauka nelielas lokālās izmaiņas ir nepārtraukti, kuras deļ lielā signāla pastiprinājuma var tikt detektētas kā transporta līdzekļi. Noteiktās transporta līdzekļa paraksta parametru vērtības (impulsa maksimālā vērtība un nolašu skaits virš sliekšņa) tiek sūtīti

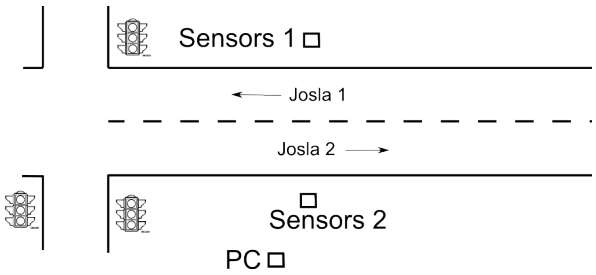


Att. 3.9: Vieglā transporta līdzekļa paraksts pēc summēšanas trijās asīs un filtrācijas veikšanas

uz ceļa malā novietoto mini skaitļotāju. Datu sūtīšana tiek nodrošināta izmantojot bezvadu sakarus 433 MHz frekvencē.

3.3.4 Signāla apstrādes algoritms uz mini skaitļotāja

Magnētisko sensoru motes novietošana transporta līdzekļu klasifikācijas un reģistrēšanas realizēšanai redzama 3.10 attēlā. Iepriekš tika jau minēts, ka izveidotais risinājums strādā pa motes pāriem – 2 sensoru motes uz vienu punktu. Sensora motes tiek novietotas tieši viena otrai pretī, veidojot vienu pāri. Mini skaitļotājs



Att. 3.10: Magnētiskā sensora motes izvietojuma shēma

(3.10 attēlā apzīmēts ar PC) saņem datus no abām motēm un, veicot analīzi, nosaka kurā joslā un kāda tipa transporta līdzeklis atrodas. Dati uz mini skaitļotāju (PC) no katras no motēm tiek sūtīti ar 400 ms laika nobīdi. Uztvērējs, saņemot informāciju par transporta līdzekļa magnētiskā paraksta parametrim 0,4s gaida, vai no otra sensora arī netiks sūtīta informācija par transporta līdzekli. Ja dati netiek saņemti, informācija par reģistrēto transporta līdzekli tiek pārsūtīta galvenajam datoram. Transporta līdzekļu klasifikācija tiek veikta klasificējot vieglos un smagos transporta līdzekļus. Pie smagajiem transporta līdzekļiem tiek pieskaitīti arī mikroautobusi. Klasifikācija tiek veikta balstoties uz transporta līdzekļa paraksta maksimālās vērtības un nolašu skaita virs līmeņa analīzi. Veicot testus reālos ceļa apstākļos tika noskaidrots, ka smagais transporta līdzeklis atbilist signāla amplitūdai pēc otrās pastiprinājuma kaskādes lielākai par 0,8V un nolašu skaitam virs līmeņa lielākam par 100. Ja parametri ir mazāki par minētajiem, transporta līdzeklis atbilst vieglajam transporta līdzeklim. Izveidotajam risinājumam ir vairāki iespējamie varianti transporta līdzekļu detektēšanā:

1. Abās joslās vienlaicīgi viens otram pretī brauc smagie transporta līdzekļi. Šajā gadījumā abi transporta līdzekļi tiks detektēti un klasificējot tie tiks noteikti kā smagie transporta līdzekļi.
2. Abās joslās viens otram pretī brauc vieglie transporta līdzekļi. Šajā gadījumā abi transporta līdzekļi tiks detektēti un klasificēti kā vieglie transporta līdzekļi.
3. Vienā no joslām atrodas smagais transporta līdzeklis, otrā vieglais transporta līdzeklis. Šajā gadījumā vieglais transporta līdzeklis netiks detektēts, bet smagais transporta līdzeklis tiks detektēts un klasificēts kā smagais transporta līdzeklis. Smagā transporta līdzekļa magnētiskais paraksts (lokālais Zemes magnētiskā lauka kropļojums) ir ievērojami lielāks par vieglā transporta līdzekļa magnētisko

parakstu, respektīvi, smagais transporta līdzeklis tiks detektēts abās magnētiskā sensora motēs, tuvāk esošajā ar lielāku vērtību, tālāk esošajā ar mazāku, kas atbilst vieglajam transporta līdzeklim. Ja abās joslās vienlaicīgi atrodas smagais un vieglais transporta līdzekļi, vieglais transporta līdzeklis netiek detektēts un klasificēts.

4. Abās joslās viens otram pretī vienlaicīgi brauc vieglie transporta līdzekļi. Šajā gadijumā abi transporta līdzekļi tiek detektēti un klasificēti kā vieglie transporta līdzekļi.

Izveidotais risinājums tika testēts reālas satiksmes apstākļos, ietverot 1595 transporta līdzekļus abos kustības virzienos.

Lane 1		
	Heavy vehicle and minibus count	**Percents**
Ground truth	388	100%
False positive	16	4,1%
False negative	20	5,1%
	Light vehicle count	**Percents**
Ground truth	560	100%
False positive	35	6,2%
False negative	91	16,2

Att. 3.11: Magnētiskās sensora motes reģistrēto un klasificēto transporta līdzekļu skaits 1. joslā

Lane 2		
	Heavy vehicle and minibus count	**Percents**
Ground truth	283	100%
False positive	17	6 %
False negative	7	2,47
	Light vehicle count	**Percents**
Ground truth	364	100%
False positive	27	7,4%
False negative	24	6,6 %

Att. 3.12: Magnētiskās sensora motes reģistrēto un klasificēto transporta līdzekļu skaits 2. joslā

Lane 2		
	Vehicle count	**Percents**
Ground truth	1595	100%
False positive	8	0,5%
False negative	83	5,2%

Att. 3.13: Magnētiskā sensora motes reģistrēto transporta līdzekļu skaits uz brauktuves

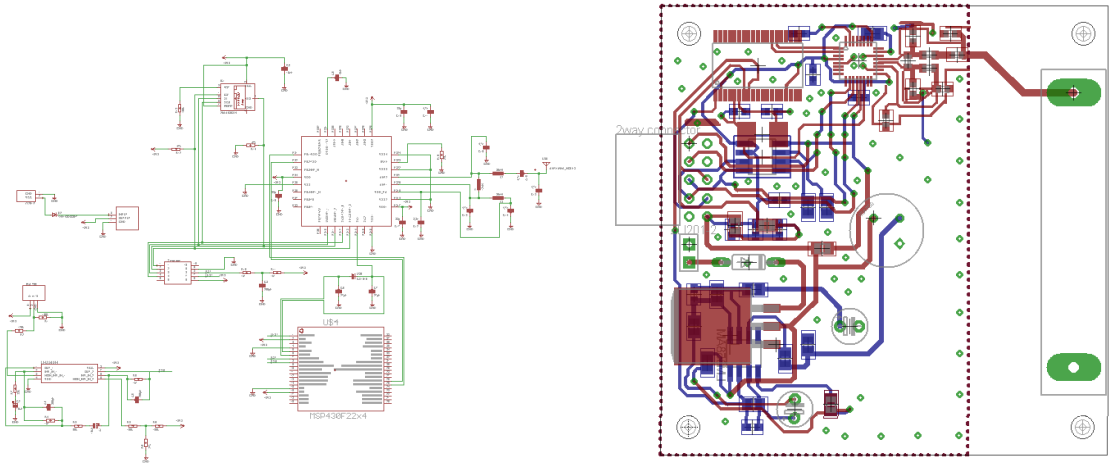
3.11 attēlā redzami rezultāti transporta līdzekļu skaitīšanai un reģistrēšanai 1.joslā. Attēlā 3.12 redzama transporta līdzekļu klasifikācija un skaits 2. joslā. Attēlā 3.13 redzams kopā uz ceļa esošo transporta līdzekļu skaits.

Transporta līdzekļi tika skaitīti uz klasificēti manuāli, bet rezultāti vēlāk salīdzināti ar sensoru doto rezultātu. Apzīmējums "Ground truth" nozīmē transporta līdzekļus, kas fiziski atradās uz brauktuves un tika saskaitīti manuāli. "False positive" norāda notikumus, kad transporta līdzeklis tika pieskaitīts, kaut

gan reāli tas uz brauktuves neatradās. ”False negative” norāda transporta līdzekļu skaitu, kuri reāli uz ceļa atradās, taču dažādu iemeslu dēļ magnētiskā sensora mote tos nepieskaitīja un nekvalificēja. 1. joslā satiksme bija intensīvākā, kā arī sadalīta pa ”porcijām” dēļ netālu esošā regulējamā krustojuma. Veicot transporta līdzekļu klasifikāciju, kļūdas procents ir lielāks kā veicot tikai detektēšanu. tas skaidrojams ar to, ka izveidotais risinājums nosaka transporta līdzekļa tipu balstoties uz pieņēmumu, ka transporta līdzekļi brauc aptuveni vienādā attālumā no ceļa malas. Izmantojot magnētiskos sensorus piedāvātajā risinājumā, nav iespējams noteikt, cik tālu no sensora objekts atrodas. var būt gadījums, kad vieglais transporta līdzeklis brauks tuvu ceļa malai, līdz arī to arī tuvu motei. Šajā gadījumā vieglais transporta līdzeklis var tikt klasificēts kā smagais transporta līdzeklis, jo nav zināms attālums līdz objektam. Testu un mērījumu laikā tika novērots, ka vairāki secīgi viens aiz otra braucoši smagie transporta līdzekļi var tikt apvienoti vienā transporta līdzeklī, kas ievieš ”False negative” kūdu. Tāpat nav iespējams izšķirt vieglo transporta līdzekli, kas brauc tuvāk kā 10 metrus aiz smagā transporta līdzekļa (katrs gadījums ir unikāls, jo transporta līdzekļa magnētisko parakstu ietekmē arī krava, ko tas ved).

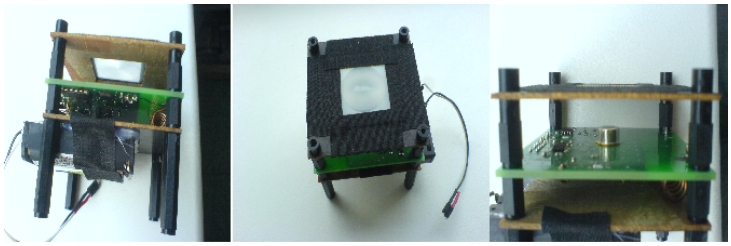
3.4 Siltuma sensora mote

Jebkurš transporta līdzeklis darbojoties (pārvietojoties) izdala siltumu, kura galvenais avots ir dzinējs (gada aukstajos mēnešos arī salons). Uztverot šo izdalīto siltumu, netieši tiek noteikta transporta līdzekļa esamība uz brauktuves. Siltuma starojuma uztveršanai tiek izmantoti pasīvie infrasarkanā starojuma sensori (PIR). PIR sensori ir spējīgi uztvert objektus kuri atrodas kustībā. Sāvošu transporta līdzekli PIR sensors nespēj reģistrēt. Izveidotās PIR sensora motes shematiskais zīmējums un PCB redzami 3.14 attēlā. PIR sensora mote



Att. 3.14: PIR sensora motes shēmtehniskais risinājums

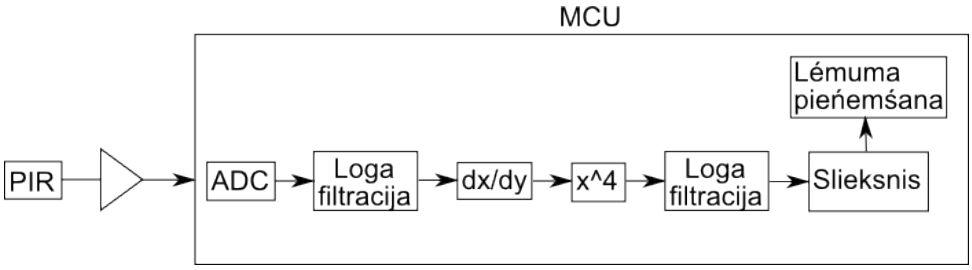
sastāv no trīs daļām: analogās daļas, ciparu daļas un bezvadu datu sakaru nodrošināšanas daļas. Analogā daļa ietver pašu PIR sensoru (Murata IRA-E7000), kā ari divu līmeņu pastiprinājuma kaskādi, iekļaujot analogos pirmās kārtas filtrus. Signāla filtrēšana



Att. 3.15: PIR sensora mote

nepieciešama, lai izdalītu derīgo signālu, kuru rada kustībā esošs transporta līdzeklis, no apkārtējiem

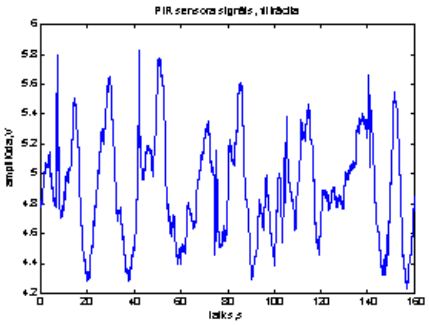
trokšņiem, kas vairumā gadijumu ir gaisa plūsmas. Ciparu daļa ietver mikrokontrolieri, kura uzdevums ir signāla ciparošana, apstrāde un lēmuma pieņemšana par transporta līdzekļa esamību uz brauktuves. Pieņemtais lēmums par reģistrēto transporta līdzekli tiek nosūtīts MITS serverim izmantojot bezvadu datu pārraidi 433 MHz frekvencē. Izveidotā PIR sensora mote redzama 3.15 attēlā. Galveno uzdevumu transporta līdzekļa detektēšanā veic mikrokontrolieris, kurā ietverta signāla apstrāde. Signāls pēc analogā pastiprinātāja tiek ciparots ar 208 Hz frekvenci. Principiālā PIR motes darbības shēma redzama 3.16 attēlā. Frekvences lielums ir izvēlēts ņemot vērā atļauto braukšanas ātrumu, pamatojums dots sadaļā "Magnētisko sensoru mote". Pēc ciparošanas signāls tiek filtrēts izmantojot loga filtru ar garumu 10 nolases. Pēc signāla filtrēšanas tiek veikta atvasināšanas operācija, kuras rezultātā iegūtais signāls tiek kāpināts 4. pakāpē. Šādā veidā tiek mazināta trokšņa ietekme. Nākamais solis ir



Att. 3.16: PIR sensora motes algortima shēma

atkārtota loga filtrācijas izmantošana. Šajā gadījumā loga filtra garums ir 30 nolases. Pēc atkārtotas vidējošanas laikā tiek veikta signāla sliekšņošana un tālāk analizēti tikai tie notikumi, kuri ir virs sliekšņa. Sliekšņa lielumu tiešā veidā ietekmē izvēlēta Freneļa lēca, tādēļ, veicot lēcas nomaiņu uz citu, nepieciešams veikt atkārtotu sensora kalibrāciju, piemērojot jaunu slieksni. Lēmums par transporta līdzekļa detektēšanu tiek pieņemts, jāizpildās sekojoši nosacījumi: signāls virs sliekšņa ir vairāk kā 50 nolases, nokrītot signāla līmenim zem sliekšņa ir reģistrētas 60 nolases. Šāds algoritms nepieciešams dēļ dažādām siltuma kustībām, kuras rodas transporta līdzekļa kustības rezultātā un kuru dēļ, neizmantojot augstāk minēto novērtējumu, viens transporta līdzeklis var tikt detektēts kā vairāki transporta līdzekļi.

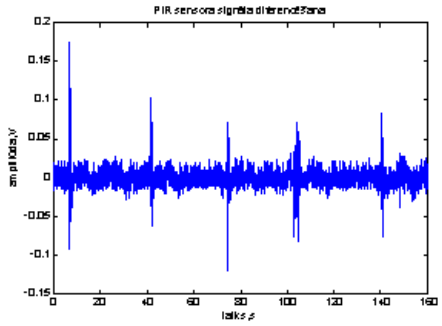
No 3.17 līdz 3.20 attēliem redzams signāls dažādos apstrādes momentos.



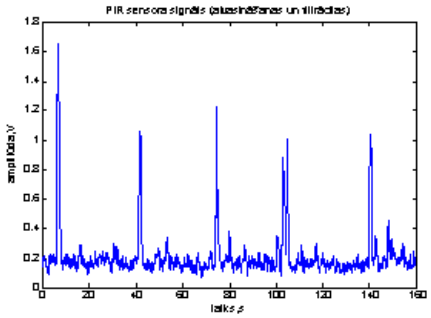
Att. 3.17: PIR sensora signāls pēc analogā filtra pielietošanas un pastiprināšanas divās kaskādēs

Attēlā 3.17 redzams signāls pēc divu pakāpju pastiprinājuma veikšanas. Ir pielietota analogā signāla filtrācija, ierobežojot signālu joslā. Šāds signāls nav tipisks transporta līdzekļu detektēšanā, jo reti kad ir silto gaisa masu svārstības kā šajā gadījumā. Attēlā 3.18 redzams signāla atvasinājums. Ar atvasināšanas palīdzību ir noņemtas zemās frekvenču svārstības, kuras iepriekš traucēja izšķirt transporta līdzekli. Atvasināšanas operāciju ir vienkārši realizēt mikrokontrolierī, kā arī tā neaizņem daudz resursu. Pēc atvasinājuma signālam tiek ņemts modulis un piemērota loga filtrācija (3.19). Pēdējais solis ir celšana ceturtajā pakāpē, kad ievērojami samazinās trokšņu lielums (3.20). Pēc visu šo operāciju veikšanas iespējam viegli detektēt transporta līdzekli, kas 3.20 attēlā atbilst pīķiem.

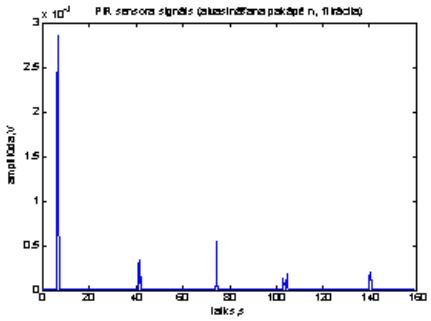
Veicot PIR sensora motes testus reāla ceļa apstākļos tika sasniegta transporta līdzekļu detektēšanas ticamība 96%. Kļūdas transporta līdzekļu detektēšanā galvenokārt parādās dēļ distances neievērošanas starp transporta līdzekļiem. Ja distance starp diviem viens aiz otra braucošiem transporta līdzekļiem ir mazāka kā 5 metri, liela iespēja, ka abi transprta līdzekļi tiks detektēti kā viens objekts. PIR sensori



Att. 3.18: PIR sensora signāls pēc atvasināšanas operācijas



Att. 3.19: PIR sensora signāls moduļa un loga filtracijas operācijām



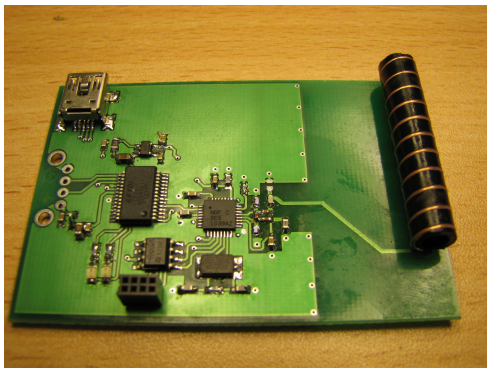
Att. 3.20: PIR sensora signāls pēc kāpināšanas ceturtajā pakāpē

paredzēti novietošanai virs brauktuves, katrā joslā pa vienam sensoram. Par cik MITS sistēma paredzēta ekspluatācijai ārpus apdzīvotām vietām un un divu joslu ceļiem (katrā virzienā ne vairāk kā viena josla), sensorus iespējams arī novietot ceļa malā uz mastiem, kuru augstums ir ne mazāk par 6m. Maksimālais novietošanas augstums izveidotajam sensoram sastāda 20 metrus.

3.5 Bezvadu datu komunikācija starp sensoru mezgliem

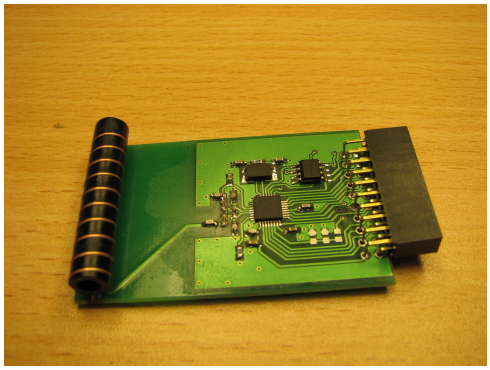
Projekta gaitā tika veikti eksperimenti ar dažādiem sensoru risinājumiem automašīnu detektēšanai. Eksperimentu gaitā tika secināts, ka mērījumu pārraidei nepieciešams izmantot bezvadu sakarus. Vadu sakaru izmantošana nav optimāla, jo nepieciešams vadus iestrādāt ceļa segumā vai novietot tos uz ceļa, līdz ar to bezvadu sakaru izmantošana ir efektīvāks un lētāks risinājums. Lai nodrošinātu datu pārsūtīšanu uz MITS punktu ir izveidoti divi prototipi: raidītājs, kurš pieslēgts pie sensora motes un uztvērējs, kurš caur USB pieslēgvietu pieslēgts pie MITS punkta. Uztvērēja modulis redzams 3.21 attēlā, tas sastāv no USB-UART FT232 pārveidotāja nRF9e5 raiduztvērēja , kurš darbojas 433MHz frekvenču

joslā, izejas salāgošanas kontūra un antena.



Att. 3.21: Uztvērējmodulis ar USB pieslēgvietu

Uztvērēj modulis paredzēts izmantošanai priekš magnētisko un siltuma sensoru datu saņemšanas un pārsūtīšanas uz MITS punktu. Katrai sensoru grupai tiek izmantots savs uztvērējs, magnētiskiem sensoriem 434.7MHz frekvenču kanāls, savukārt siltuma sensora motei bezvadu datu pārraide tiek nodrošināta 433.1MHz frekvenču kanālā. Siltuma sensora motes bezvadu raiduztvērējs nRF9E5 izstrādāts kopā uz vienas iespiedplates. Sensoram tiek izdalīts identifikācijas numurs, pēc kura uztvērējs to atpazīst. Siltuma sensora mote noraida divu baitu garu datu paketi, kura ietver informāciju par identifikācijas numuru un automašīnas esamību. Raidītājs darbojas Master režīmā un sūta datus, kad ir noticis notikums(detektēta automašīna), savukārt uztvērējs atrodas slave režīmā un gaida datus no sensora. Pēc datu saņemšana uztvērējs tos pārsūta uz MITS punktu.



Att. 3.22: Raidītāj modulis priekš magnētiskā sensora motes

Raidītājs priekš magnētiskā sensora datu pārsūtīšanas ilustrēts 3.22 attēlā. Radio modulis sastāv no nRF9E5 raiduztvērēja, radio izejas salāgošanas kontūra un Helical Wire antenas. Radio modulis tiek savienots ar magnētiskā sensora moti izmantojot 2.54mm savienotājus. Radio modulis darbojas Master režīmā. Magnētiskā sensora motes vadības mikrokontrolieris pa UART saskarni pārsūta datus uz radio moduli, kurš savukārt datus pārsūta pa bezvadu kanālu uz uztvērēju, kurš pieslēgts pie MITS punkta. Katram radio modulim ir piešķirts unikāls identifikācijas numurs, kurš ir ietverts datu paketē, lai uztvērējs varētu atpazīt sūtītāju. Sensoru sistēma darbojas pa pāriem, sensori novietoti pretī viens otram abās ceļa pusēs. Lai nerastos situācijas, kad dati pa bezvadu kanālu tiek nosūtītu vienlaicīgi, viens no sensoriem datus sūta ar aizturi 0.4s.

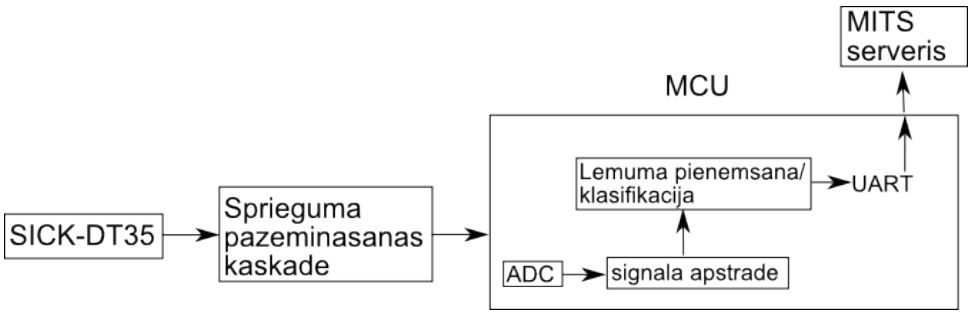
3.6 Lāzera sensora mote

Lāzera sensora mote darbojas uz radara principa. Tiek raidīts gaismas impulss, kuram atstarojoties no virsmas un atnākot atpakaļ uztvērējā, tiek noteikts attālums līdz objektam. Projektā tiek izmantoti SICK DT35 lāzera sensors, kurš darbojas infrasarkanajā diapazonā. Lāzera darbības attālums ir līdz 12 metriem, atstarojošas virsmas var tikt detektētas līdz 35 metru attālumam. Transporta līdzeklis

nevar tikt uzskatīts kā atstarojoša virsma, atšķiras transporta līdzekļu krāsas, līdz arto arī atstarošanas koeficienti. Izmantotais lāzera sensors redzams 3.23 attēlā. Dati no sensora tiek saņemti izmantota analogā signāla izeja ar mainīgu spriegumu, kurš mainās no nulle līdz barošanas spriegumam (12V). Datu apstrāde tiek veikta izmantojot mikrokontrolieri. Lai sensora izejas spriegums būtu apstrādājams ar mikrokontrolieri, nepieciešams veikt sprieguma pazemināšanu proporcionāli visā diapazonā. Maksimālais mikrokontroliera darba spriegums ir 3,6V, taču, lai analogais signāls tiktu korekti ciparots, ieteicams izmantot atbalsta spriegumu (tiek ģenerēts mikrokontrolierī) ar maksimālo vērtību 2,5V.

Att. 3.23: Lāzera sensors transporta līdzekļu reģistrēšanai un klasifikācijai

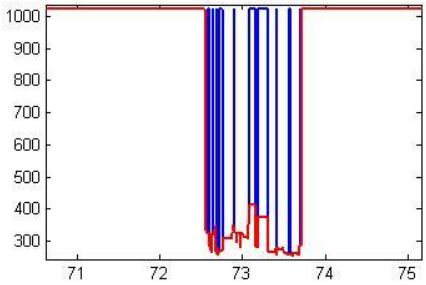
Mikrokontrolierī signāla ciparošanai tiek izmantots 10 bitu ADC (iespējami 1024 signāla līmeņi). Samazinot sensora izejas sprieguma diapazonu vairāk kā 4 reizes, samazinās arī distances izšķirtspēja. Konkrētā sensora izšķirtspēja ir 0,05 līdz 5 mm, kas ir salīdzinoši augsts rādītājs. Lai veiktu transporta līdzekļu reģistrēšanu un klasifikāciju, pietiek ar izšķirtspēju 0,05 - 0,1 m. Līdz ar to samazinot izejas sprieguma vērtību, transporta līdzekļu detektēšana netiek traucēta. Principiālā lāzera sensora motes blokshēma redzama 3.24 attēlā. Pēc signāla ciparošanas ADC, lai noteiktu transporta līdzekļa esamību uz brauktuves, nepieciešams veikt signāla apstrādi. Ne no visām virsmām signāls tiek atstarots (nav iespējams nomērīt attālumu), piemēram, transporta līdzekļa vējstikls, virsmas, kas nav perpendikulāras lāzera motes signālam. Ja virsma ir melna un matēta, lāzera stars arī netiek atstarots. Šāds gadījums iespējams, ja vieglajam transporta līdzeklim ir uzstādīts jumta bagāžnieks, vai arī tiek vesta kāda krava (piemēram piekabē).



Att. 3.24: Lāzera sensors transporta motes blokshēma

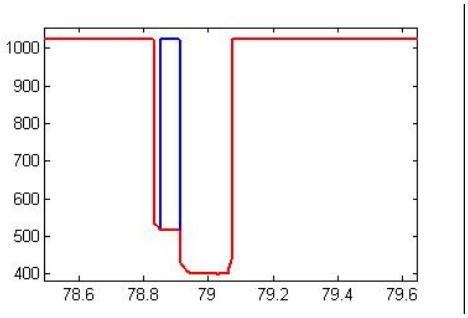
Ja izpildās minētie apstākļi viens transporta līdzeklis var tikt detektēti kā vairāki. Tipisks piemērs, kad motes uztvertais signāls ir ļoti saraustīts, redzams 3.25 attēlā. Ar zilu krāsu atzīmēts smagā

transporta līdzekļa (baļķvedējs ar pilnu kravu) paraksts. Redzams, ka signāls ir ļoti saraustīts un ir daudz gadījumu, kad atstarošanās nenotiek un signāls pieņem maksimālo vērtību (t.i. transporta līdzeklis nav motes redzamības zonā). Izveidotās motes darbības princips ir balstīts uz īpašību, ka asfalta segums absorbē lielu daļu no krītošā lāzera stara, līdz ar to laika momentos, kad transporta līdzekļi nepārvietojas motes darbības zonā, attālums līdz ceļa virsmai ir maksimālais, t.i. 10 metri. Kā iepriekš minēts, šāda signāla "lēkāšana" iespējama dēļ transporta līdzekļa (vai tā kravas) virsmas lāzera stara absorbcijas vai atstarošanas īpašībām. Lai novērstu signāla "lēkāšanu" ir izstrādāts algoritms, kurš seko līdzi transporta līdzekļa lāzera sensora parakstam, kā rezultāta tiek iegūts signāls kurš attēlā 3.25 attēlots ar sarkano krāsu. Veicot signāla izsekošanu laikā, tiek pieņemts, ka signāla absorbcija, gadījumā, ja transporta līdzeklis ir motes redzamības zonā, nevar būt ilgāk kā 60 nolases. Ja absorbcija ir ilgāku laiku, tiek pieņemts lēmums par viena transporta līdzekļa reģistrēšanu.



Att. 3.25: Lāzera sensora motes signāls reģstrējot smago transporta līdzekli

3.26 attēlā redzams vieglā transporta līdzekļa lāzera sensora paraksts. Arī šajā grafikā ar sarkanu līniju ir atzīmēts izsekotais paraksta signāls, bet ar zilu - no transporta līdzekļa atstarotais signāls. Par cik vieglajam transporta līdzeklim ir daudz mazāk pārejas virsmu, kā smagajam transporta līdzeklim (baļķvedējam ar kravu), signāl ir ievērojami mazāk saraustīts kā 3.25 attēlā redzamajā gadījumā.

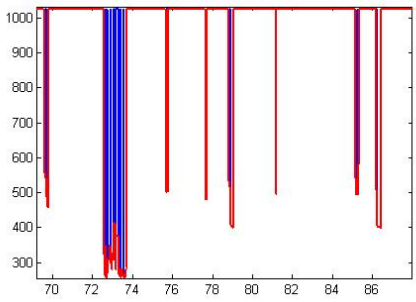


Att. 3.26: Lāzera sensora paraksts vieglajam transporta līdzeklim

Vairāku transporta līdzekļu plūsma redzama 3.27 attēlā. Šeit var redzēt transporta līdzekļu atšķirību pēc augstuma, kas arī tiek ņemts vērā, veicot klasifikāciju. Transporta līdzekļi tiek klasificēti sekojošās klasēs:

- vieglais transporta līdzeklis
- paaugstinātas caurejamības transporta līdzeklis (SUV)
- mikroautobuss
- smagais transporta līdzeklis
- liels smagais transporta līdzeklis

Klasifikācija tiek veikta mērot attālumu no sensora motes līdz transporta līdzeklim. Vieglā transporta gadījumā attālums būs vislielākais, bet smagā transporta līdzekļa gadījumā - vismazākais. Dati par transporta līdzekļu klasificēšanu tiek nosūtīti MITS punktam. Lai veiktu transporta plūsmas klasifikāciju, lāzera sensora motei jāveic kalibrācija attiecībā pret transporta līdzekļiem. Ja mote tiek novietota citā augstumā, kā veicot kalibrāciju, klasificējot transporta līdzekļus, rezultāts būs kļūdains. Vajadzības gadījumā kalibrāciju iespējam veikt īsā laikā. Lāzera sensora mote paredzēta novietot virs ceļa

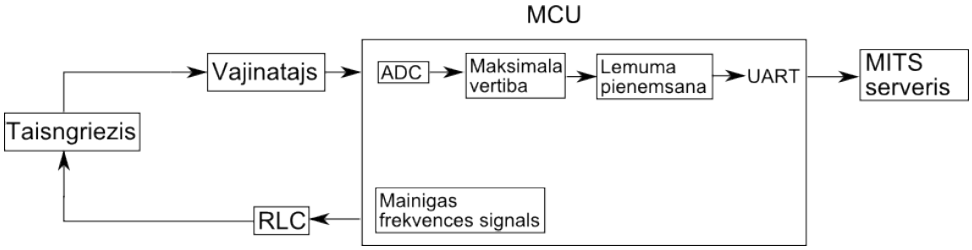


Att. 3.27: Lāzera sensora motes signāls dažādiem transporta līdzekļiem

braucamās daļas, katrā joslā pa vienam sensoram. Motes darbības precizitāte sasniedz 98% transporta līdzekļu detektēšanai un 95% transporta līdzekļu klasifikācijai. Veicot detektēšanu, kļūdas galvenokārt izraisa transporta līdzekļu tumšā krāsa (matēta). Veicot klasifikāciju, nekorekts lēmums var tikt pieņemts gadījumā, ja smagais transporta līdzeklis pārvietojas ar piekabi bez kravas vai arī gadījumā ja smagais transporta līdzeklis velk piekabi bez kravas.

3.7 Ceļa stāvokļa un laika apstākļu noteikšanas sensori

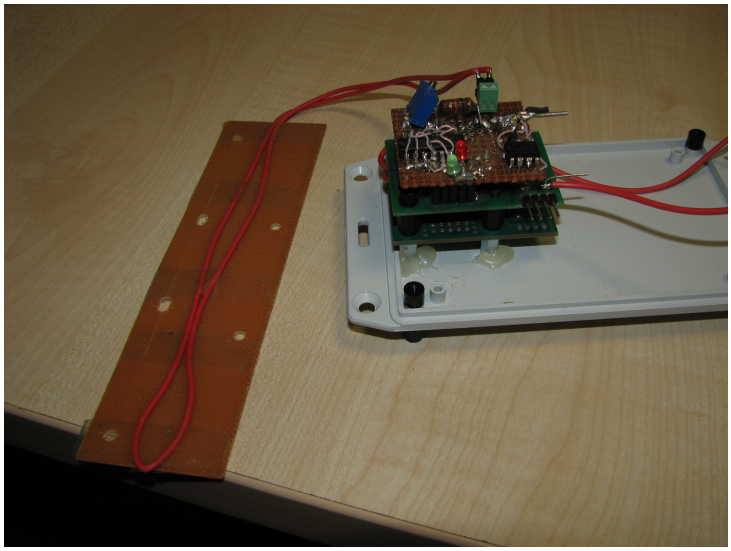
Ceļa stāvokļa noteikšana tiek balstīta uz tiešo mērījumu principa. Tiešā veidā tiek noteikts, vai ceļa virsma ir slapja, mitra, vai sausa. Izmantojot netiešo metodi, tiek noteikts, vai ceļa virsma ir apledojusi un vai uz tās ir pret-sasaluma reaģents pietiekamā daudzumā. Brauktuves virsmas stāvokļa noteikšanai ir izveidota ceļa stāvokļa mote, kura iekļauta kopējā iekārtā laika apstākļu noteikšanai. Ceļa stāvoklis tiek noteikts veicot rezonanses frekvences mērījumus. Motes darbības pamatā ir RLC svārstību kontūrs, kurā tiek padots mainīgas frekvences signāls. Izveidotais risinājums ir invazīvs pret ceļa virsmu. Pati mote tiek novietota ceļa malā, taču, lai noteiktu kāda ir brauktuves virsma, brauktuves virsmā nepieciešamas iestrādāt izolētu vadītāja cilpu (garums līdz 0,2 m). Izveidotās motes darbības princips redzams 3.28 attēlā.



Att. 3.28: Ceļa stāvokļa noteikšanas diagramma tiešā mērījuma gadījumā

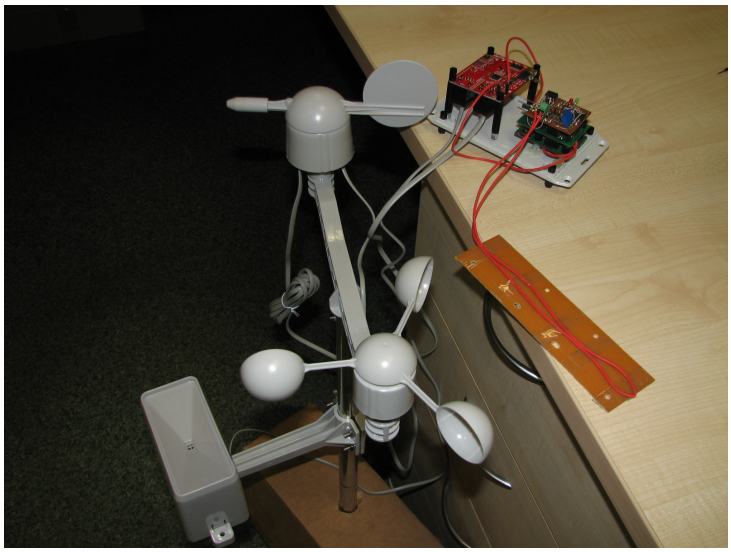
Mikrokontrolieris ģenerē mainīgas frekvences signālu, kurš tiek novadīts uz virknes RLC svārstību kontūru. Kontūrā mainīgie parametri ir induktivitāte L un kapacitāte C, kuri mainās atkarībā no brauktuves stāvokļa. Ar taisngriezi signāla amplitūda tiek vidējota, iegūstot līdzspriegumu ar mainīgi amplitūdu. Līdzsprieguma vērtība mainās atkarība no mikrokontroliera ģenerētās frekvences un vides parametriem. Rezonanses gadījumā signāla vidējā vērtība var pārsniegt vairākas reizes barošanas spriegumu un sabojāt mikrokontroliera ADC ieeju. Lai šādu gadījumu izslēgtu, mitē ir ievietots signāla vājinātājs. Pēc vājinātajā signāla vērtība rezonanse gadījumā nepārsniedz 2,5V. Mikrokontrolierī signāls pēc vājinātāja tiek ciparots, kā arī tiek noteikta maksimālā vērtība, kas atbilst rezonansei. Par cik rezonanses frekvence mainās atkarībā no virsmas stāvokļa - sauss, slapjš vai mitrs segums, iespējams, veicot ieejošā signāla un ģenerētā signāla analīzi, iespējams noteikt ceļa stāvokli. Izveidotā mote testēta reālos apstākļos. testu laikā veikta motes kalibrācija. Uzstādot moti uz tiltu konstrukcijām, var rasties vajadzība veikt virsmas stāvoklim atbilstošās rezonanses frekvences atkārtotu kalibrāciju, jo vides parametri (induktivitāte un kapacitāte) var atšķirties no parametriem novietojot moti tieši uz zemes.

Izveidotais ceļa stāvokļa sensors/mote redzams 3.29 attēlā.



Att. 3.29: Ceļa stāvokļa noteikšanas mote

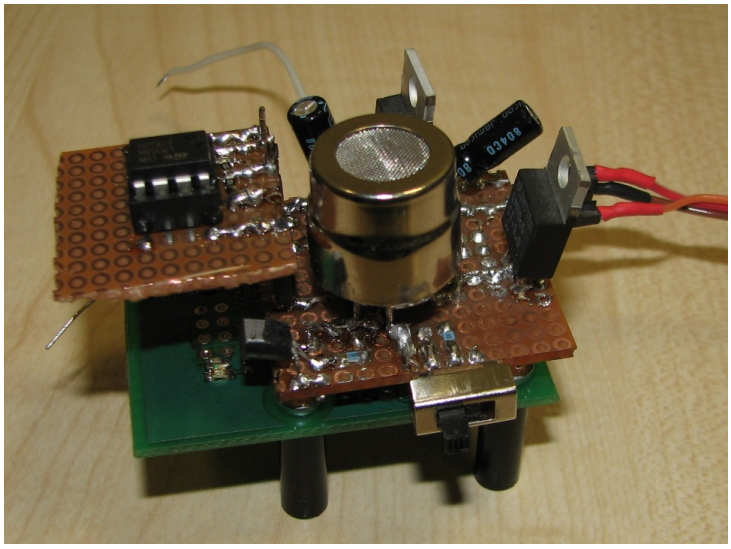
Ceļa stāvokļa motes ir pievienota laika apstākļu noteikšanas sensoram. Izveidotas sensoru kopējais modulis ilustrēts 3.30 attēlā.Datu sūtīšana no laika apstākļu sensora un ceļa stāvokļa motes uz MITS punktu tiek vadīta izmantojot ceļa stāvokļa motē iebūvēto mikrokontrolieri. Par laika apstākļa noteikšanas sensoru izmantots Weather Board uz kura atrodas gaismas, mitruma, temperatūras, atmosfēriskā spiediena,vēja virziena un stipruma sensori.



Att. 3.30: Laika apstākļu sensors kopā ar ceļa stāvokļa moti

Sensoru kopums nodrošina pilnvērtīgu informāciju par lokālo laika apstākļu situāciju. Laika apstākļa ir savienots ar ceļa stāvokļa moti izmantojot UART interfeisu. Koordinators ir ceļa stāvokļa mote, kur nodrošina laika apstākļu sensora datu nolasīšanu un sapludināšanu kopā ar ceļa stāvokļa motes datiem. Pēc sapludināšanas uz MITS punktu tiek nosūtīta viena datu paka, kura satur lokālā mērījuma informāciju no visiem laika apstākļu sensoriem. Datu pakas garums ir 61 baits. Laika apstākļu sensors veic mērījumus ar 6 sekunžu intervālu. Vispirms tiek veikta laika apstākļu sensoru datu savākšana, pēc kuras tiek datu paketei pie vienots ceļa stāvokļa motes mērījums. Pie laika apstākļu sensora tiek pievienots CO gāzu sensors, kurš attēlots 3.31. Gaisa kvalitātes noteikšanai izveidotais prototips sastāv no CO jūtīgā elementa HS134, vadības mikrokontroliera, barošanas ķēdes un izejas signāla buferis ar

dalītāju. Sensora jūtīgais elements maina pretestību atkarībā no gāzu vides, kurā tas atrodas.



Att. 3.31: CO gāzu sensors

Sensora jūtīgais elements tiek periodiski sildīts 90 sekundes ar 4V lielu spriegumu un 60 sekundes ar 1,4V lielu spriegumu, paša sensora barošanai nepieciešami 5V liels spriegums. Sildīšanas laika intervālu pārslēgšanu nodrošina mikrokontrolieris. Pie katras sildīšanas cikla ar 4V spriegumu tiek nomērīts izejas spriegums, pēc kura lieluma nosaka gāzu koncentrāciju vidē, kurā atrodas sensora jūtīgais elements. Izejas spriegums no gāzu sensora ir robežās līdz 5V, lai to būtu iespējams saciparot izmantojot MSP 430F2234 mikrokontroliei ir izveidots izejas sprieguma dalītājs izmantojot operāciju pastiprinātāju kā atkārtotāju. Tiek izmantots MSP430F2234 iebūvētais analogais ciparu pārveidotājs. 4V sildīšanas periodā izmantojot mikrokontrolieri tiek veikta analogā signāla saciparošana un maksimālās vērtības atrašana. Maksimālā vērtība norāda kādā gāzu vidē atrodas sensors. Attiecīgi indikatīvi ar dažādu krāsu gaismas diožu palīdzību tiek izvadīta informācija par vidi, kā arī informācija tiek nosūtīta uz MITS punktu.

3.8 Mikrofonu masīvs

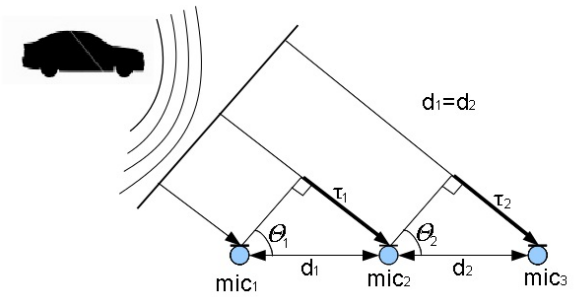
3.8.1 Automašīnu detektēšana izmantojot mikrofonu masīvu

Projekta laikā tika izstrādāta automašīnu skaitīšanas un kustības virziena noteikšanas prototips balstoties uz automašīnas kustības rezultātā radītā trokšņa signāla analīzi izmantojot mikrofonu masīvu. Prototips realizēts kā autonoma ierīce reālā laika sistēmā, kuras darbībai nav nepieciešams personālais dators. Algoritmi automašīnas skaitīšanai un virziena noteikšanai realizēti uz ARM CortexM4F procesora C valodā. Izstrādātais prototips balstās uz neinvazīvu sensoru risinājumu automašīnu detektēšanai. Izveidotā risinājuma prototips redzams attēlā 3.32, kur parādīts skaitļošanas modulis un izstrādātās iekārtas kopskats ar mehāniskās konstrukcijas izpildījumu.

Automašīnai atrodoties kustībā, rodas troksnis, kura avots ir riepas, dzinējs, aerodinamika. Stiprākās spektra komponentes atrodas joslā no 500Hz līdz 3500Hz. Izmantojot šo troksni iespējams veikta automašīnu detektēšanu pielietojot aiztures starp mikrofoniem noteikšanas algoritmus, kuri balstās uz cross-spectrum metodi. Aprēķini balstās uz pieņēmumu, ka pienākošie skaņas viļņi ir plakani viļņi, kas raksturojas ar skaņas avota 'tālu' atrašanos no mikrofonu matricas. Pieņēmums ļauj pielietot ģeometriskus aprēķinus, lai izskaitļotu automašīnas novietojumu attiecībā pret mikrofonu matricu izejot no mikrofonu savstarpējā attāluma un skaņas viļņu pienākšanas aiztures vērtības starp mikrofoniem. Darbības blokshēma attēlota attēlā 3.33.Lai nodrošinātu automašīnu skaitīšanu nepieciešami minimums divi mikrofoni, ar kuriem tiks veikta skaņas signālu fiksēšana un ierakstīšana.

Izstrādātais akustiskais sensors paredzēts novietošanai brauktuves malā horizontālā plaknē apmēram 3m no brauktuves un 1,25m augstumā no zemes. Šāds novietojums ļauj iekārtu viegli pārvietot un uzstādīt. Mikrofonu matricas mikrofoniem jāatrodas perpendikulāri zemei un paralēli brauktuvei.

Att. 3.32: Iekārtas aprēķinu moduļa un iekārtas kopskats



Att. 3.33: Skaņas pienākšanas leņķa aprēķina modelis

Šāds uzstādījums nepieciešams, lai izpildītos ģeometriskais nosacījums pie skaņas pienākšanas aiztures noteikšanas. Ja netiek ievērots uzstādīšanas nosacījums iekārtai samazinās automašīnu detektēšanas precizitāte. Automašīnai tuvojoties un attālinoties no mikrofonu masīva skaņa katrā mikrofonā nenonāk vienlaicīgi, bet ar kaut kādu aiztures lielumu 'x'. Pēc šīs aiztures lieluma var noteikt automašīnas kustības virzienu attiecībā pret mikrofonu masīvu. Aiztures lielums mainās atkarībā no automašīnas atrašanās pozīcijas, līdz ar to pastiprinājums pirms ACP izvēlēts pietiekami mazs, lai netiktu fiksēts tālu(virs 30m) esošu automašīnu troksnis.

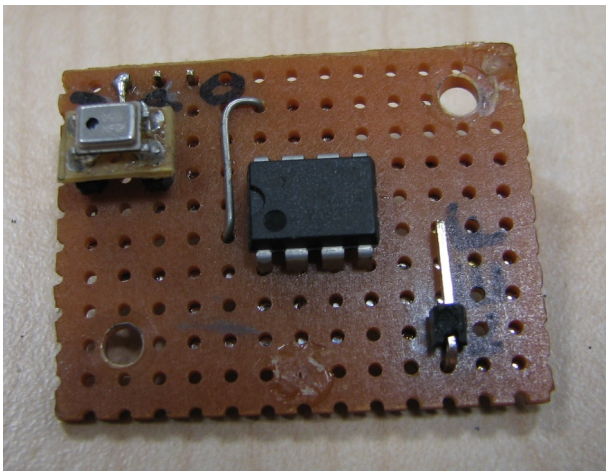
3.8.2 Tehniskie parametri un darbības princips

No 3 mikrofoniem tiek izmantoti 2 MEMS mikrofoni, kuru savstarpējais attālums sastāda 20cm. Attālums atkarīgs no uztveramā skaņas viļņa garuma. Analogais skaņas signāls no mikrofoniem pirms ACP ieejas tiek pastiprināts izmantojot divu kaskāžu operāciju pastiprinātājus.

- 3 MEMS mikrofoni WM7110;
- Ieejas pastiprinātājs ar 2 kaskādēm;
- Audio ACP AD1974 ar diferenciālo ieejas pārveidotāju;
- MSP430F2013 ACP vadībai;
- ARM Cortex M4F procesors(STM32F4 Discovery).
- UART USB pārveidotājs(FT232RL) datu pārsūtīšanai uz MITS punktu;

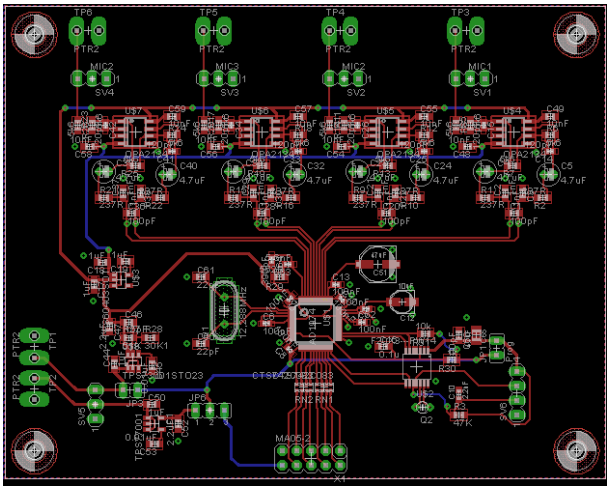
Tiek izmantots ST952ID operāciju pastiprinātājs. Attēlā 3.34 parādīta izveidotā plate uz kuras izvietots MEMS WM7110 omni-directional mikrofons kopā ar operāciju pastiprinātāju. Otrās kaskādes

slēgumā realizēts pirmās kārtas Hi-pass filtrs, lai nofiltrētu komponentes, kuras atrodas joslā zem 400Hz. Šajā joslā atrodas vēja radītais troksnis, kurš ir nevēlams signālu apstrādē.



Att. 3.34: MEMS mikrofons kopā ar divu kaskāžu operāciju pastiprinātāju

Pēc pastiprinātāja signāls tiek pārveidots uz diferenciālo slēgumu, kurš nepieciešams AD1974 analogam ciparu pārveidotājam, kas tiek izmantots priekš analogā skaņas signāla pārveidošanas diskrētā formā. Par atbalsta sprieguma avotu tiek lietots ārējs barošanas elements(AA tipa baterija) , lai nodrošinātu ka signāls ACP ieejā tiktu saglabāts darbības diapazonā. Izveidota iespiedplate 3.35 ar AD1974 analogo ciparu pārveidotāju kopā ar ieejas diferenciālo pārveidotāja analogo risinājumu un ACP vadības mikrokontrolieri MSP F2013.

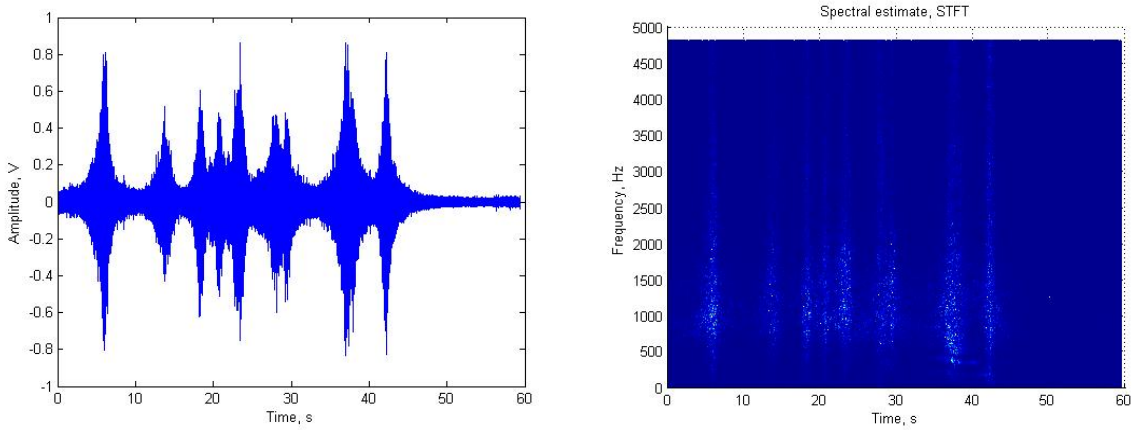


Att. 3.35: Iespiedplate AD1974 ar ieejas ķēdi un vadības mikrokontrolieri

Mikrokontrolieris nodrošina analogā ciparu pārveidotāja iekšējo reģistru vērtību iestatīšanu un darbības režīma kontroli(start,reset funkcijas). AD1974 plate tiek savienota ar STM32F4 Discovery plati izmantojot I2S saskarni. I2S interfeiss nodrošina datu nosūtīšanu no AD1974 iespiedplates uz STM32F4 Discovery. Saskarnes uzstādījumi un sūtīšanas biežumu nosaka 'Master' iekārta, kas šajā gadījumā ir AD1974 analogais ciparu pārveidotājs. STM32F4 Discovery iekārta darbojas 'slave' režīmā un saņem datus nepārtraukti balstoties uz iepriekš definētiem nosacījumiem. Dati pēc to saņemšanas tiek saglabāti ieejas buferos,kur pēc tam tiks apstrādāti. Sūtīšanas biežumu nosaka diskretizācijas frekvence ar kādu tiek veikta analogā signāla pārveidošana. Veicot eksperimentus, balsoties uz testa ierakstiem un veicot šo ierakstu spektrālo analīzi tika noskaidrots, ka optimālā diskretizācijas frekvence sastāda 24kHz. Pēc apstrādes dati pa UART USB pārveidotāju tiek nosūtīti uz MITS punktu. Dati satur informāciju par automašīnas esamību un kustības virzienu.

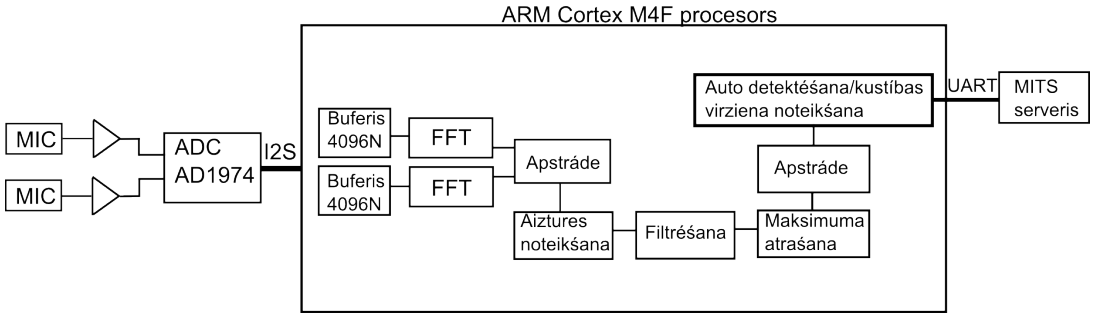
3.8.3 Apstrādes algoritms un tā realizācija ARM Cortex M4F procesorā

Izstrāde tika balstīta sākotnēji izmantojot ierakstītus signālus no reālas automašīnu plūsmas uz ceļa un analizējot tos ar MatLab. Eksperimentālās izstrādes etapā tika konstatēts, ka optimālais analizējamā signāla garums, lai veiktu automašīnu detektēšanu izmantojot divus mikrofonus sastāda 170ms pie diskretizācijas frekvences 24kHz. Par cik tiek analizēti skaņas signāli, kuru avots atrodas kustībā, loga garuma izvēle ir svarīga algoritma precizitātē. Attēlā 3.36 parādīts 1 minūtes gars skaņas signāls, kurš ierakstīts reālas satiksmes apstākļos un šī signāla spektrs. Algoritms iedalās divos etapos: aiztures lieluma atrašana starp diviem mikrofonu signāliem un automašīnas detektēšana un kustības virziena noteikšana. Abas daļas ir realizētas uz Cortex M4F procesora un rezultātā MITS punktam tiek nosūtīta informācija par automašīnas esamību un kustības virzienu. Procesorā realizētā algoritma blokshēma parādīta 3.37 attēlā.



Att. 3.36: Automašīnas kustības radītais skaņas signāla ieraksts un šī signāla spektrs

Pēc signālu nolasīšanas no AD1974 abu mikrofonu ciparu nolases tiek saglabātas divos atsevišķos ieejas buferos. Buferu garums ir 4096 nolases. Tiek izvēlēts references signāls pret kuru tiks aprēķināta aizture. Abiem ieejas signāliem tiek veikta FFT, pēc šis operācijas pirmā kanāla spektrs tiek sareizināts ar otra kanāla spektra kompleksi saistīto komponenti. Rezultātu izdala ar maksimālo absolūto vērtību (veic normēšanu). Sekojošais etaps ietver reālās reizinājuma komponentes inverso FFT. Pēc tam tiek veikta iegūta rezultāta matricas kvadrantu apmaiņa(fftshift operācija). Šī operācija ļauj iegūto rezultātu pārveidot tā , lai rezultāts būtu uztverams un no tā būtu iespējams atrast maksimumu un no tā arī aizturi. Maksimuma atrašanās vieta norāda uz divu signālu savstarpējo nobīdi. Lai ierobežotu nejaušu gadījuma rakstura maksimumu atrašanu tiek veikta filtrācija pēc kvadrantu apmaiņas darbības. Maksimuma meklēšana notiek ierobežotā loga robežās, kur loga platumu nosaka maksimālā iespējamā aizture.



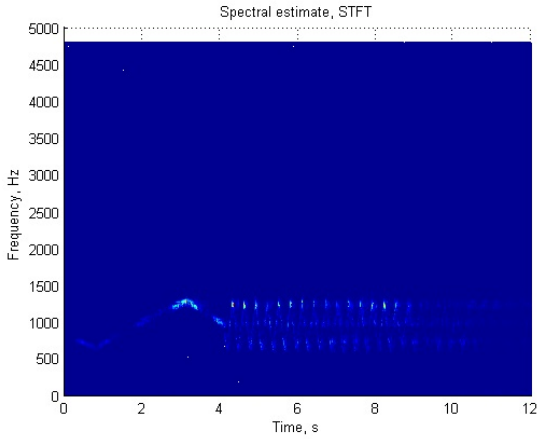
Att. 3.37: Procesorā realizētā algoritma blokshēma

Lai veiktu automašīnu skaitīšanu tika izveidots bīdes reģistrs, kurā tiek glabātas tekošās piecas aprēķinātas leņķa vērtības. Tālākajos aprēķinos izmanto absolūtās leņķa vērības, kur tiek meklēts gadījums, kad reģistra vidus adresē atrodas maksimālā vērtība. Sekojošais lēmēj mehānisma etaps sevī ietver tuvāko kaimiņu vērtību salīdzināšanu ar atrasto maksimumu. Tiek veikta atņemšana un salīdzināta

starpība starp maksimumu un abām kaimiņu vērtībām, ja starpība pārsniedz 15 grādus, tad tiek nolemts, ka garām ir pabraukusi automašīna. Eksperimentāli tika atrasts šis slieksnis pie kura tiek nolemts par automašīnas pieskaitīšanu. Identisks bīdes reģistrs tiek izveidots arī priekš aprēķinātajām aizturēm, kuras reģistrā tiek ierakstītas attiecīgajām leņķa vērtībām. Šis reģistrs nepieciešams automašīnu kustības virziena noteikšanai. Tiek analizētas visas 5 aiztures vērtības un savstarpēji salīdzinātas. Attiecīgi algoritms lemj par automašīnas esamību un tās kustības virzienu vienlaicīgi. Kopējais laiks, kas nepieciešams, lai algoritms nolemtu par automašīnas esamību un tās kustības virzienu ir 68ms. Kustības virziens tiek noteikts salīdzinot attiecīgās aiztures vērtībās, kuras glabājas bīdes reģistrā. Transporta līdzeklim tuvojoties aizture samazinās, attālinoties palielinās, līdz ar to analizējot aiztures lielumu un tās zīmi tiek noteiks skaņas pienākšanas virziens. Automašīnas detektēšanas precizitāte sastāda 92 procentus. Automašīnas netiek detektētas situācijās, kad to distance atrodoties kustībā ir mazāka par 8 metriem. Vieglā automašīna netiks detektēta situācijās, kad tā brauc uzreiz aiz smagās, jo smagās automašīnas radītais troksnis ir lielāks. Tik detektēta tikai viena automašīna situācijās, kad divas pretējos braukšanas virzienos braucošas automašīnas pabrauc garām mikrofonu masīvam.

3.8.4 Operatīva transporta līdzekļa detektēšana

Operatīvā transporta līdzekļa detektēšana balstās uz jaudas spektrālā blīvuma analīzi. Sirēnas radītais troksnis atrodas frekvenču joslā 600-1500kHz. Sirēnas signāla STFT attēlots 3.38. Sirēnas troksnis raksturojas ar lineāri mainīgu, kāpjošu un krītošu frekvenci (chirp signāls). Izmantojot Matlab realizēts joslas filtrs sirēnas radītās skaņas atdalīšanai. Operatīva transporta detektēšanas algoritms balstās uz jaudas spektrālā blīvuma aprēķinu noteiktam signāla loga garumam, kas atbilst vienam pilnam sirēnas skaņas signāla periodam un sastāda 4,4 sekundes.



Att. 3.38: Operatīvā transporta līdzekļa sirēnas STFT spektrs

Tiek uzdots noteikts slieksnis, pie kura pārsniegšanas tiek aprēķināta vidējā frekvence, ja vidējā frekvence pārsniedz 1000Hz, tad algoritms 'nolemj', ka uz ceļa atrodas operatīvais transporta līdzeklis. Ar šo metodi iespējams iegūt informāciju tikai par operatīvā transporta esamību, bet neļauj noteikt kustības virzienu. Tāpēc tika pielietota laika aiztures aprēķinu metode starp diviem mikrofoniem(aprakstīta iepriekšējā nodaļā), lai noteiktu operatīvas transporta līdzekļa kustības virzienu. Operatīvā transporta līdzekļa detektēšanas sistēma paredzēta ārpus apdzīvotām vietām.

Nodaļa 4

Transporta līdzekļu atpazīšana un algoritmi

Anotācija

Nodaļa apraksta attēlu apstrādes grupas izveidoto algoritmu, kas lieto virs ceļa novietotu kameru (neinvazīvs sensors), lai iegūtu video plūsmu. Video tiek apstrādāts, atklājot pa ceļu braucošās automašīnas, gan dienas, gan nakts apstākļos. Atklātajām mašīnām tiek noteikti parametri - ātrums, garums, platums, augstums, braukšanas virziens. Pēc atklātajiem parametriem mašīnas tiek klasificētas vieglajās un smagajās. Algoritms spēj iegūt arī mašīnu braukšanas trajektorijas un noteikt, vai mašīnas ir apstājušās. Šī informācija tiek izmantota, lai noteiktu negadījumus uz ceļa, kā arī, lai noteiktu pār sarkano luksafora signālu braucošas automašīnas. Lai algoritms darbotos reālā laikā, tas tika veidots efektīvs pret skaitļošanas jaudām, izmantojot inovatīvu metodi - intervāli uz virtuālām atklāšanas līnijām.

4.1 Ievads

Pirmais solis satiksmes novērtēšanā ir automašīnu skaita noteikšana uz konkrētā ceļa, noteiktā virzienā. Šāda informācija ļauj, piemēram, pārslēgt luksofora signālus atbilstoši esošajai situācijai krustojumā, nevis fiksētos laika sprīžos, kā tas visbiežāk notiek šobrīd. Automašīnu saskaitīšanai lietoto dažādus sensorus - induktivās cilpas, pneimatiskos un pjezoelektriskos sensorus, sonārus, mikroviļņu diapazonā darbojošos sensorus, kā arī kameras. Pēdējo priekšrocības ir lēta uzstādīšana un apkalpošana, kā arī paplašinātas iespējas dažādu parametru noteikšanai. Video apstrāde ļauj noteikt gan braucošas gan stāvošas automašīnas, iespējams novērtēt to ātrumu, aptuvenos izmērus, kā arī iespējams iegūt aptverošāku informāciju par situāciju uz ceļa. Papildus informācija ļauj uzkrāt statistiku, tādējādi nodrošinot ar nozīmīgiem datiem transporta plūsmas planotājus. Ceļa situācijas nepārtraukta analīze ļauj uzlabot cilvēku drošību.

Video apstrādes risinājuma trūkums ir attēlu apstrādes augstās prasības pret skaitļošanas jaudu. Tā kā mūsdienās šī jauda kļūst arvien lielāka un lētāka, video apstrāde satiksmes novērtēšanai kļūst arvien populārāka. Algoritmi, kas tiek izmantoti video apstrādē, ir daudzveidīgi, tomēr pētniecība šajā jomā arvien turpinās - tiek meklētas jaunas metodes, kas atbilst vienmēr mainīgajām, pretrunīgajām prasībām pret darbības precizitāti un pieejamajiem skaitļošanas resursiem. Attēlu grupas mērķis ir izveidot uzticamu video apstrādes sistēmu transporta plūsmas novērtēšanai. Tā kā kameras nav uzticams sensors stipra lietus, sniega vai miglas apstākļos, tad kopējā MITS sistēmā video apstrāde un citi sensori efektīvi papildina viens otru.

4.2 Saistītie risinājumi

Literatūras izpētes rezultātā tika apkopotas dažādas metodes, algoritmi automašīnu detektēšanai un izsekošanai. Pilnvērtīgs dažādu metožu pārskats piedāvāts publikācijā [**?**]. Secināms, ka metožu dažādība ir plaša, tādēļ izvēlēties vienu konkrētu pieeju uzdevuma risināšanai vadoties tikai no literatūras bez eksperimentāliem testiem nebūtu optimāli.

Pirmais uzdevums, ko veiks sistēma, ir mašīnu detektēšana. Tika apskatītas vairākas publikācijas un patenti, kuros piedāvāti algoritmi konkrētā uzdevuma risināšanai, kā arī grāmatas par ciparu attēlu apstrādi, kuros šie algoritmi aprakstīti sīkāk.

Avotos [**?**] un [**?**] piedavāts samērā vienkāršs paņēmiens kustīgu objektu nodalīšanai no ceļa - izmantojot blakus kadru starpību, kas atstāj apstrādātajā kadrā tikai izmainījušos pikseļu vērtības. Tā kā metode ļauj detektēt tikai kustībā esošas automašīnas, tad lielāka uzmanība tiek vērsta uz nevis blakus kadru starpības izmantošanu, bet gan uz kadra un fona bildes starpības iegūšanu. Lai algoritms efektīvi darbotos mainīgajos āra apstākļos, fona attēlam jabūt adaptīvam. Publikācijas [**?**] un [**?**] piedāva vairākas metodes adaptīva fona attēla iegūšanai.

Fonam nepiederošs objekts var arī nebūt automašīna, tādēļ algoritmam izdalītajā objektā jameklē arī kādi transporta līdzekli raksturojoši parametri (piemēram, izmērs), kas ļaus to klasificēt kā mašīnu. Rakstos [**?**], [**?**] par šādu parametru tiek piedāvāts automašīnas kontūrs, kā arī blakus pikseļu intensitāšu vērtību veidotas robežas automašīnai atbilstošajā kadra apgabalā. Robežas ir parametrs, kas saglabājas maz mainīgs pie dažāda apgaismojuma (neskaitot nakts stundas) un dažādos laikapstākļos, tādēļ, veidojot mūsu risinājumu, šim parametram tika pievērsta uzmanība.

4.3 Automašīnu atklāšana

4.3.1 Metode - intervāli uz atklāšanas līnijas

Populārās literatūrāš īstenotās metodes tika īstenotas MATLAB Simulink vidē un testētas dažādas to kombinācijas. Tika secināts, ka efektīvai reālā laika darbībai ir jāveido jaunu metodi. Tādēļ gala iekartā attēlu apstrādes grupa piedāvā citu pieeju automašīnu atklāšanai, pētot nevis visu kadru, cenšoties nodalīt kustīgus objektus no tukša ceļa, bet gan pētot tikai dažas ceļam perpendikulāras līnijas (attēls 4.1). Šāds algoritms ir zīmīgi mazāk prasīgs pret skaitļošanas jaudām. Turklāt, izveidojot algoritmu, kas pēc iespējas vairāk derīgas informācijas var iegūt no līnijas, iespējams to paplašināt par daudzu līniju sistēmu ar augstu precizitāti automašīnu atklāšanā.

Att. 4.1: Mašīnām atbilstošo intervālu noteikšana uz ceļam perpendikulāras līnijas

Tālāk apskatīts attēlu apstrādes algoritma pirmais posms, kas saskaita līniju šķērsojušās mašīnas un nosaka aptuvenu objekta ātrumu (pikseļu skaitu, par kuru objekts pārvietojas kadra laikā).

Piedāvātais algoritms:

1. Ieejā pienāk video kadri. Atkarībā no video materiāla, tiek ievadītas konstantes - nepieciešamās vērtības gala algoritmam ir jāievada uzstādot iekārtu – veicot kalibrēšanu:
   - Kadra *platums* un *augstums*;
   - Pētāmās līnijas atrašanās rinda *LiinijasR*;
   - Pētāmais apgabals uz līnijas *(No, Liidz)*, kas iekļaus tikai to ceļa apgabalu no kadra, kurā automašīnas brauc vienā virzienā;
   - Mašīnai atbilstošais minimālais un maksimālais intervāla garums uz līnijas *IntGarMin* un *IntGarMax*;

- Sliekšņu vērtības *Sl1*, *Sl2*;
- Kadru skaits, pēc kuriem pārbauda intervālu nemainīgumu un salīdzina robežas (k).

Vairākas nepieciešamās vērtības algoritms var noteikt pats – līdz ar to, tās ir arī adaptīvas:

- Mašīnām atbilstošais vidējais intervālu garums;
- Mašīnām atbilstošais vidējais intervālu dzīves ilgums;

2. Video tiek konvertēts uz melnbaltu. Melnbalto attēlu bez automašīnu skaitīšanas var izmantot arī kameras noraustīšanās detektēšanai:

   2.1. Kadrs tiek aizturēts par diviem kadriem.

   2.2. Aizturētais kadrs tiek atņemts no šī brīža kadra.

   2.3. Starpībai tiek ņemta absolūtā vērtība.

   2.4. Tiek iegūta visu starpības pikseļu summa.

   2.5. Ja summa pārsniedz noteiktu vērtību, tiek pieņemts lēmums, ka kamera ir noraustījusies.

   Ja kameras noraustīšanās netiek noteikta, tad nepieciešamās skaitļošanas jaudas samazināšanai pirms konvertēšanas uz melnbalto attēlu no video tiek izgriezts interesējošais apgabals – pētāmā līnija un dažas apkārt esošās līnijas. Tālāk tātad ir jāapstrādā tikai šis nelielais apgabals.

3. Tiek izdalīta interesējošā līnija, ar iespēju ņemt vērā arī kādu skaitu L blakus esošu līniju.

   3.1. Izdalītās līnijas, tiek aizkavētas par vienu vai vairākiem kadriem (koeficients *aizture*). Koeficients ir atkarīgs no video kadru skaita sekundē un mašīnu vidējā ātruma (no tā, cik kadru laikā mašīna pārbrauc pāri līnijai).

   3.2. Aizkavētās vērtības uz līnijām tiek atņemtas no šī brīža vērtībām.

   3.3. Starpībai tiek ņemta absolūtā vērtība, iegūstot izejā līniju ar starpības absolūtajām vērtībām.

   3.4. Vairāku līniju gadījumā, katram izejas līnijas pikselim piešķir maksimālo no L atbilstošajām starpības absolūtajām vērtībām. Attēlā 4.2.a parādītas laikā uzkrātas starpības līnijas vērtības. Attēla augšējā rinda atbilst iegūtajai līnijai pirms 150 kadriem, otrā – pirms 149 utt. Pēdējā rinda ir šī brīža kadrā iegūtā starpības līnija.

4. No starpību līnijas tiek izgriezts pētāmais ceļam atbilstošais intervāls *No-Liidz*.

5. Starpības līnijai tiek pielietots mediānas filtrs, lai samazinātu liekus pārrāvumus starp mašīnai atbilstošajiem intervāliem.

6. Starpības līnija tiek sliekšņota – rezultāts parādīts attēlā 4.2.b. Tiek izmantotas divas sliekšņa vērtības *Sl1* un *Sl2*, kur *Sl2*<*Sl1*.

   6.1. Šī brīža starpības līnijai tiek pielietots lielais slieksnis *Sl1*.

   6.2. Iegūtajā binārajā līnijā tiek noteiktas visu savienoto balto intervālu sākuma Sak_x un beigu $Beig_x$ koordinātes.

   6.3. Tiek pārbaudīts, vai starp kādiem diviem intervāliem reizē izpildās sekojošas sakarības:

   6.3.1. $|(Beig_{x_1} - Sak_{x_1}) - Beig_{x_2} - Sak_{x_2}| < Sl_{liidziigs}$ // Intervāli ir līdzīgi pēc garuma (domāti lukturi), to starpība nepārsniedz kādu sliekšņa vērtību.

   6.3.2. $Beig_{x2} - Sak_{x1} > IntGarMin$ // Intervāli savienojot būs pietiekami lieli automašīnai.

   6.3.3. $Beig_{x2} - Sak_{x1} < IntGarMax$ // Intervāli savienojot nebūs pārāk lieli automašīnai.

   6.5. Tiek noteikti slieksni pārvarējušo intervālu garumi.

   6.6. Ja uz iepriekšējā kadra atbilstošās līnijas tiek atklāts $(Beig_x - Sak_x) > IntGarMin$ un $(Beig_x - Sak_x) < IntGarMax$, tad šī brīža kadrā intervāla apņemtajā apgabalā un tā apkārtnē (plus 10 pikseļi uz abām pusēm no intervāla gala koordinātēm) tiek lietots mazais slieksnis *Sl2*.

7. Tālāk notiek iegūto intervālu apstrāde:

   7.1. Uz līnijas iegūtie intervāli tiek aizkavēti par kadru.

   7.2. Ja šī brīža kadrā kāds intervāls daļēji pārklājas ar iepriekšējā kadra intervālu, tie tiek apvienoti kopējā intervālā.

Tiek panākts, ka laikā intervāli tiek apvienoti. Piemēram, vairāki mazāki intervāli, kas sākotnēji atbilda atsevišķām automašīnas detaļām, šādā veidā dod ieguldījumu nākamajos kadros veidojot kopīgo automašīnas intervālu. Attēlā jau ir arī pielietota vēlāk sistēmā iegūstamā informācija, kad intervāli jāslēdz.

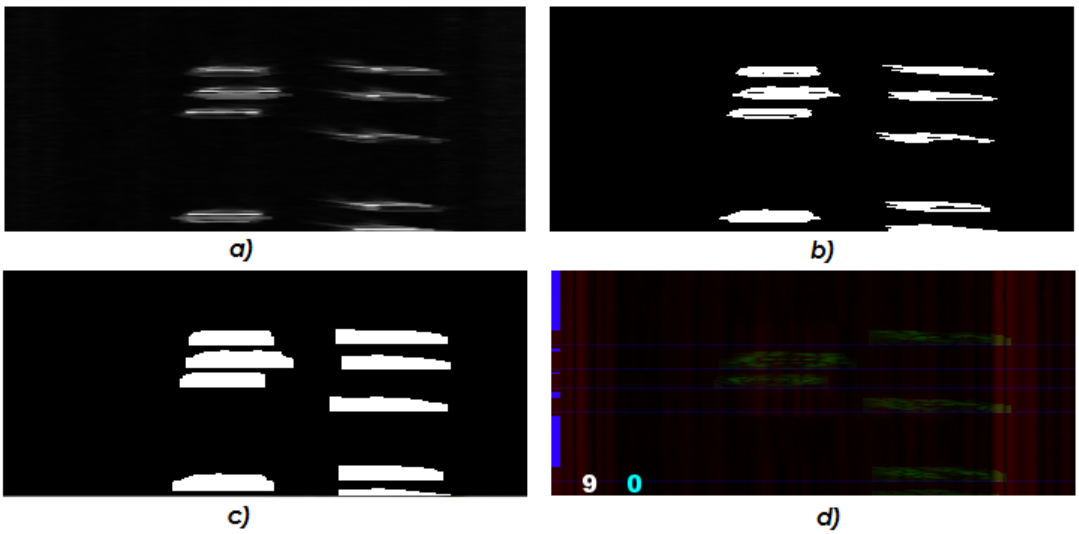
7.3. Ja kāds no iegūtajiem intervāliem atbilst punkta 6.6. prasībām, tas tiek uzskatīts par iespējamu mašīnai atbilstošo intervālu.

7.4. Ja kāds no intervāliem ir pārāk garš (pārsniedz vidējo automašīnām atbilstošo intervālu pareizinātu ar koeficientu), tas jāsadala sīkākos intervālos:

7.4.1. Garais intervāls $(Sak_{gar}, Beig_{gar})$ tiek salīdzināts ar iepriekšējā kadrā uz līnijas esošajiem intervāliem $(Sak_{vec}, Beig_{vec})$, pēc sekojošas sakarības: $Sak_{gar}\ Sak_{vec}||Beig_{vec}\ Sak_{gar}$.

7.4.2. Tas intervāls, kurš iepriekšējā kadrā bijis lielā intervāla robežās, pie tam, ir pārsniedzis lielumu *IntGarMin*, un atrodas vistuvāk kreisajai lielā intervāla malai, tiek ņemts par atskaites intervālu dalīšanai.

7.4.3. Atskaites intervāla gala koordināte (labajā galā) tiek ņemta par atdalīšanas koordināti, kura kļūst par melnu pikseli starp sadalītajiem intervāliem.



Att. 4.2: Agloritma darbības posmi: a) blakus kadru starpība; b) līnija pēc divu sliekšņu pielietošanas; c) apvienotie intervāli; d) fona robežas (sarkanās) un robežas uz līnijas(zaļās)

8. Ja mašīna aizbraukusi – intervāls jāslēdz.

   8.1. Intervāla gala koordinātes tiek salīdzinātas ar iepriekšējo k kadru intervāliem.

   Ja intervāls ir bijis nemainīgs šo emphk kadru laikā, tiek pieņemts lēmums, ka intervālu jāslēdz, ja arī tālāk algoritmā izpildīsies slēgšanas nosacījums pie robežu noteikšanas. Apvienotie un pēc automašīnas aizbraukšanas slēgtie intervāli parādīti attēla 4.2.d.

9. Uz pētāmās līnijas interesējošā apgabala tiek noteiktas arī robežas.

   9.1. Līnijai tiek pielietots nogludinošais gausa filtrs.

   9.2. Tiek iegūta katru blakus pikseļu starpība, šī starpība veido robežu līniju.

   9.3. Robežu līnijai tiek ņemta absolūtā vērtība.

   9.4. Ja uz šī brīža pētāmās līnijas neeksistē neviens intervāls (pat īss), šī līnija tiek saglabāta un tiek ņemta vērā pie tukša ceļa robežu līnijas.

   9.5. Fona robežu līnijas ieguvei, katram pikselim tiek ņemta moda no 10 saglabātajām līnijām.

   9.6. Ja ieejā ir mašīnas izmēram atbilstoši intervāli, šo intervālu atstāj nemainīgu robežu līnijā – pārējo, intervālam nepiederīgo apgabalu pielīdzina nullēm.

   9.7. Robežu līniju aiztur par k kadriem.

9.8. No izdalītās, aizkavētās robežu līnijas atņem uzkrāto fona līniju.

9.9. Starpības pikseļiem tiek pielietots slieksnis.

9.10. Tiek izskaitīts, cik pikseļu ir pārvarējuši slieksni.

9.11. Ja slieksni pārvarējuši mazāk kā noteikts skaits pikseļu, tas nozīmē, ka intervāla apgabalā nav pietiekami daudz ceļam neatbilstošu robežu, un konkrētais intervāls jāslēdz (ja izpildās arī nosacījums 8.2.). Attēlā 4.2.d sarkanās līnijas ir maz mainīgās, uzkrātās fona robežas - zaļā krāsā ir fonam nepiederošu objektu robežas. Zilās horizontālās līnijas parāda momentus, kuros intervāla slēgšanas lēmuma pieņemšanai tikuši skaitīti slieksni pārvarējušie robežas pikseļi. Baltais skaitlis rāda cik šādu pikseļu ir šī brīža kadrā, gaiši zilais - cik šādu pikseļu bijis pēdējā lēmuma pieņemšanas brīdī. Biezā vertikālā līnija kreisajā attēla malā parāda, kuros brīžos ir atjaunojies fona robežas uzkrājošais buferis.

10. Tiek atklāta automašīna, ja intervāls tiek slēgts un slēgšanas brīdī tas atbilst sekojošiem parametriem:

    10.1. Intervāls nav nedz par platu, nedz par šauru: $Beig_x - Sak_x > IntGarMin$ un $Beig_x - Sak_x < IntGarMax$. Intervāls ir pastāvējis vismaz noteiktu skaitu kadrus.

11. Pie atbilstošo intervālu slēgšanas tiek saglabāti intervālu garumi. No vairākiem šiem garumiem tos summējot un izdalot ar skaitu iegūst vidējo intervālu garumu.

4.3.2 Atklāšanas algoritma uzlabojumi

Iepriekšējā nodaļā 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. Salīdzinājumā ar iepriekš apskatītām metodēm, kas apstrādāja visu kadru, testi parādīja, ka piedāvātais algoritms ir perspektīvs risinājums, tādēļ rūpnieciskā pētījuma rezultātā turpmākai uzlabošanai izvēlēts algoritms, kas apstrādā nelielu kadra daļu – ceļa virzienam perpendikulāru līniju.

Tālāk piedāvātais algoritms tika pārnests uz C++ valodas kodu. Tas ļauj veikt testus tiešsaistes režīmā, jo uz C++ īstenots algoritms kļūst ātrāks nekā darbinot to kopā ar MATLAB programmu. Turklāt, C++ arī ir projekta kopējās sistēmas valoda. 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 koriģējamu. 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.

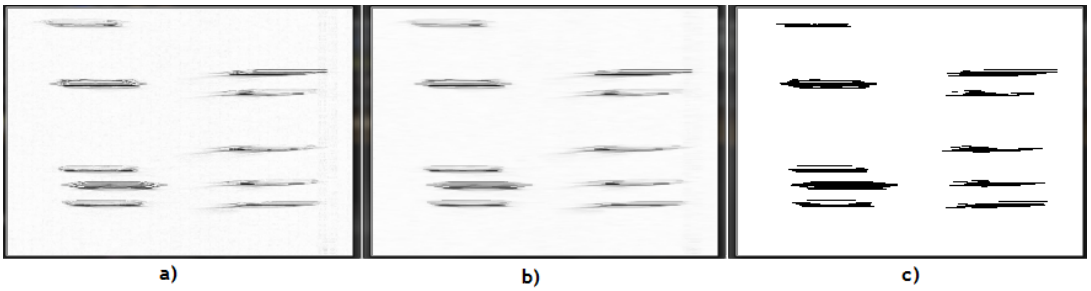
VideoProcessor ciklā tiek izsauktas *FewLines* metodes, kur nozīmīgākā ir *processLine* metode. Šajā metodē tiek apstrādāti *Line* un *Interval* klašu objekti. *Line* objekti ir attēli (matricas) ar vienu rindu. Šie attēli var būt oriģinālā kadra intensitātes pikseļi uz pētāmās līnijas, krāsu komponentes, robežas, blakus kadru starpības uz līnijas vai citi parametri. Dažādi parametri var papildināt viens otru, automašīnu atklāšanas procesā, kas sīkāk parādīts nodaļā 4.3.2. Katra parametra līnija ir tātad atsevišķs *Line* klases objekts. *FewLines* klasē izsauktās *Line* metodes apstrādā dažādo parametru līnijas, piemēram, veic gludinošo filtrāciju, uzkrāj līnijai atbilstošo, adaptīvo, tukšam ceļam atbilstošo fona līniju (viens no *Lines* mainīgajiem), atņem parametru līniju no atbilstošā fona, atrod starpības absolūto vērtību, atrod adaptīvu slieksni, 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 pikseļu ir pārvarējuši slieksni konkrētā intervāla apgabalā. Atrastie intervāli tiek saglabāti vektorā, kur jaunos 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) slieksni pārvarējušie pikseļi, 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 4.3.a, 4.3.b, 4.3.c), kā arī pēc jebkuras citas modifikācijas. Attēlos vertikālais virziens atbilst laikam - pirmā rinda parāda līnijas izskatu šī brīža kadrā, otrā rinda rada izskatu vienu kadru ieprieks, pēdeja rinda atbilst izskatam pirms 200 kadriem.



Att. 4.3: Vizualizācijas klases rezultātu izvade a) blakus kadru līniju starpības izmaiņas laikā; b) starpība pēc gludionošās filtrēšanas; c) rezultāts pēc slieksņošanas

Vizualizācijas klase ļauj arī iegūt uzskatāmu rezultāta izvadīšanu – 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.

Efektīva kameras izkustēšanās atklāšana

Uzlabota metode kameras izkustēšanās vai strauju, visu kadru aptverošu apgaismojuma izmaiņu atklāšanai. Iepriekšējā metodē tika izmantoti visi kadra pikseļi, kurus algoritms salīdzināja ar iepriekšējā kadra visiem pikseļiem. Kad starpkadru pikseļu starpību absolūto vērtību summa pārsniedza iestādāmu, konstantu slieksni, tika detektēta kameras izkustēšanās, un, šī informācija tika izmantota metodes posmos, kuros tika uzkrāti adaptīvi fona parametri.

Uzlabotajā versijā netiek pētīts viss kadrs, bet gan atsevišķi pikseļi. Lai kameras izkustēšanās atklāšanu būtiski neietekmētu braucošās automašīnas, pētāmos pikseļus jāizvēlas tā, lai daļa no tiem (ja iespējams pie dotā kameras novietojuma), atrastos ārpus ceļa. Mūsu pieejā pikseļi tiek ņemti vienmērīgi pa visu kadru – starp konkrētu un nākošo pikseli vienā rindā ir 20 pikseļi, kuri netiek ņemti vērā. Tāds pats attālums ir starp pētāmajiem pikseļiem kadra kolonnās.

Izdalīto pikseļu intensitātes tiek saglabātas un salīdzinātas ar šo pašu pikseļu intensitātēm nākamajā kadrā, iegūstot starpības absolūtās vērtības starp šīm intensitātēm. Katra iegūtā intensitātes vērtība tiek salīdzināta ar slieksni. Jaunajā algoritma versijā tiek izmantots adaptīvs slieksnis. Tā kā mērķis ir nodetektēt kameras kustību, tad šeit par sliekšņa vērtību tiek izmantota vērtība kustības detektēšanai, kas jau atrasta un lietota vienā no iepriekšējiem metodes posmiem, kurā pēc kustības esamības tika atklātas automašīnas.

Ja pietiekams skaits no starpkadru starpības pikseļiem pārvar slieksni, tiek atklāta kameras izkustēšanās. Precīza slieksni pārvarējušo pikseļu skaita attiecība pret visiem vērā ņemtajiem pikseļiem,

kas norādītu uz kameras izkustēšanos, atkarīga no tā, cik lielu daļu no kadra aizņem ceļš un cik intensīva satiksme ir uz šī ceļa. Testos ar mūsu tiešsaistes sistēmu labi rezultāti bija pie attiecības 1/8.

Mašīnu atklāšanas pazīmju klāsta paplašināšana

Intervālu slēgšanas nosacījumus veido vairāki parametri. Pirmais parametrs ir pietiekami zīmīga atšķirība uz līnijas blakus kadros. Algoritmā tika arī paredzēts novērtēt tukšam ceļam neatbilstošu robežu esamību šobrīd eksistējoša 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. Tālāk algoritms papildināts ar jauniem parametriem.

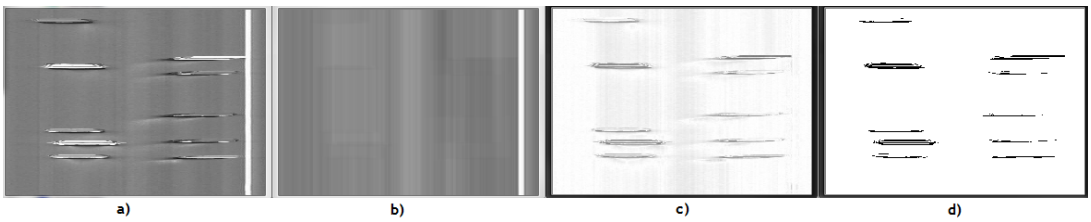
Par mašīnu var liecināt melnbalto pikseļu intensitātes atšķirība no tukša ceļa pikseļu intensitātes. Tātad ir jāiegūst tukša ceļa intensitātes līnija, un mainīgo āra apstākļu dēļ, tai jābūt adaptīvai. Fona iegūšana realizēta tādējādi, ka laikā, kamēr uz līnijas nav atrasts kāds intervāls, fons ir vienāds ar šī brīža līniju. Pretējā gadījumā, katram fona pikselim tiek nedaudz modificēta vērtība pēc formulas 4.1.

$$F_{i,k} = (1 - a_{i,k}) \cdot (0.98 \cdot F_{i,k-1} + 0.02 \cdot P_{i,k}) + a_{i,k} \cdot F_{i,k-1}, \tag{4.1}$$

kur *a=0*, ja pikselis i kadrā k atrodas šobrīd eksistējoša 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 4.1 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 4.4.a parādīta intensitātes līnijas izmaiņa 200 kadru laikā, savukārt 4.4.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 4.4.c. Tad starpības līnijai tiek noteikts un pielietots adaptīvs slieksnis, kas piešķir visiem pikseļiem tikai divas no iespējamām vērtībām – 0 vai 1. Sliekšņotas intensitātes līnijas izmaiņas laikā redzamas 4.4.d.

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

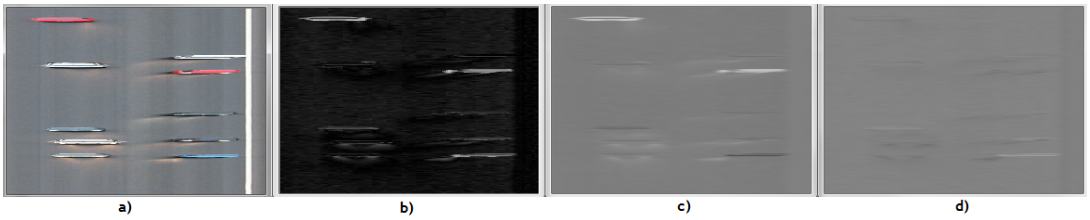


Att. 4.4: Pikseļu intensitātes izmantošana par intervālu slēgšanas parametru

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 (krāssātspozmes) telpas attēlu, kurš tāpat sastāv no 3 komponentēm, iespējams izdalīt krāsu piesātinājuma komponenti S. Attēlā 4.5.b parādīta 4.5.a attēlam atbilstošā krāsu piesātinājuma izmaiņa laikā.

Cita krāsu telpa YCrCb ļauj izdalīt sarkanās Cr un zilās Cb krāsas objektus, kas parādīts attēlos 4.5.c un 4.5.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 pikseļus. Slieksnis tiek meklēts un pielietots jau lietojot apvienoto līniju, kas gala rezultātā dod 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 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,



Att. 4.5: Līnijas krāsu komponenšu izmaiņas laikā

lai uz detektējošās līnijas uz laiku apstājusies mašīna nepaspētu kļūt par fona sastāvdaļu. Šobrīd fona adaptācijas laiks piemeklēts 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 svara koeficienti. Piešķiršana ir atkarīga no tā, cik izteikti šis nosacījums norāda uz automašīnas esamību intervālā. Piemēram, ja sliekšņotā krāsu komponenšu līnijā (CrCb) intervāla apgabalā vairāk kā trešdaļa no pikseļiem ir pārsnieguši slieksni, šim nosacījumam tiek piešķirts svars 3. Ja mazāk kā divi pikseļi pārsnieguši slieksni – svars ir 0, pārējos gadījumos – 2.

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

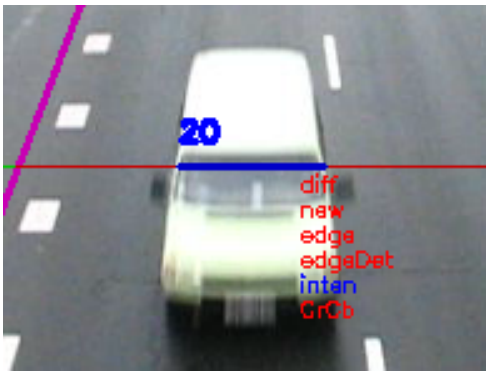
Intervāla slēgšanas nosacījumu papildināšana

Testos ar izveidoto automašīnu atklāšanas algoritmu konstatēts, ka ar līdz šim lietotajiem parametriem (intensitāte, krāsa, robežas) var nepietikt, lai noturētu intervālu eksistējošu, līdz mašīna ir izbraukusi no pētāmās līnijas. Iespējami gadījumi, kad mašīna apstājusies brīdī, kad uz līnijas atrodas tās vējstikls, kā parādīts attēlā 4.6. Ja pētāmā ceļa segums ir asfalts, tad bieži vējstikla tumšais tonis un krāsa maz atšķiras no uzkrātā fona. Sarkanie vārdi attēlā zem intervāla rāda, ka lielākā daļa parametru nenorāda uz mašīnas esamību uz līnijas. Viens vidēji stiprs parametrs (zils) konkrētajā piemērā ir intensitāte, kas nav pietiekams nosacījums, lai noturētu intervālu. Attēlā parādītajā gadījumā intervāls tātad tiek uzturēts tikai pateicoties iestrādātajai 3 kadru aizturei – ja mašīna apstātos šādā stāvoklī vismaz uz 3 kadriem, tad intervāls tiktu slēgts.

Pārsvarā gadījumos uz vējstikla, tāpat kā uz vienmērīga ceļa seguma, nav izteiktu robežu, līdz ar to arī šis parametrs nav pietiekams, lai konkrētajā situācijā intervāls netiktu slēgts. Tomēr robeža starp vējstiklu (automašīnu) un ceļu arī šajos gadījumos var būt izteikta, tādēļ, kā papildus parametrs, tiks lietota izteiktu robežu esamība eksistējoša intervāla galos. Iepriekš lietotais robežu parametrs tika skatīts visā intervāla apgabalā.

Robežu iegūšana un sliekšņošana mūsu algoritmā jau tiek veikta (robežu parametrs visā intervāla apgabalā), tādēļ papildus robežu parametra izmantošanai atsevišķa pikseļu apstrāde nav vajadzīga. Ir nepieciešams tikai nodefinēt, ka jau sliekšņotā robežu līnijā, slieksni pārvarējušie pikseļi tiek skatīti tikai konkrētos intervāla apgabalos, kuros atrodas robežas starp iespējamo mašīnu un ceļu. Šie apgabali ne vienmēr atrodas pētāmā intervāla pašos galos, jo, atšķirībā no automašīnas reālā kontūra uz līnijas, intervāls nemazina savu izmēru (šī brīža izpildījumā intervāls var tikai paplašināties, palikt nemainīgs, vai tikt slēgts). Šis apstāklis noved pie vajadzības iegūt reālo mašīnas platuma novērtējumu.

Iespējamās automašīnas platuma novērtējums mūsu risinājumā tiek balstīts uz kustības detektēšanu. Tiek lietota, jau iepriekš algoritmā iegūtā, ar divām sliekšņiem apstrādātā, blakus kadru starpības līnija. Pētāmā intervāla apgabalā tiek analizēti sliekšņotās starpības līnijas pikseļi. Gadījumā, kad gan pirmais, gan pēdējais šīs līnijas pikselis intervāla apgabalā (intervāla galos) ir pārvarējis sliekšņus, tiek pieņemts lēmums, ka mašīnas kontūrs uz detektēšanas līnijas nav samazinājies. Ja intervāla galos (katrā atsevišķi) netiek detektēta kustība, tiek apskatīta iespēja par kontūra sašaurināšanos konkrētajā intervāla galā. Piemēram, ja intervāla kreisajā pusē, tātad uz pirmā intervāla pikseļa netiek atklāta kustība, secīgi tiek



Att. 4.6: Vējstikls uz detektējošās līnijas

skatīti pikseļi pa labi no sākotnējā. Tuvākais pikselis, kas pārvarējis sliekšņus tiek pieņemts par mašīnas kontūra jauno robežu.

Šāda pieeja var būt kļūdaina, gadījumos, kad tikai daļa no kustībā esošas automašīnas tiek detektēta kā kustīga, vai, kad automašīna ir apstājusies un par tās kontūru var tikt nosaukts kāds trokšņa dēļ slieksni pārvarējis pikselis ne uz mašīnas robežas. Tādēļ algoritms tika papildināts, iestrādājot tajā nosacījumu, ka mašīnas kontūrs kadra laikā nevar samazināties pārāk strauji. Papildinātajā variantā, ja intervāla galos netiek detektēta kustība, un tiek atrasta jauna iespējamā mašīnas kontūra robeža, šīs jaunās robežas atrašanās vieta tiek salīdzināta ar atrasto robežas koordināti iepriekšējā kadrā. Ja starp robežām attālums ir lielāks par 15 pikseļiem, tad robežas koordināte netiek mainīta uz jauno. Šī pieeja ietver risinājumu gan gadījumam, kad iespējamā mašīna ir apstājusies un kontūrs līdz ar to nav jāsašaurina, kā arī gadījumam, kad kādu kadru skaitu mašīnas apgabals tuvu robežai netiek detektēts kā kustīgs objekts. Ja, pēdējā gadījumā, pēc vairākiem kadriem mašīnas kontūra robežas apgabalā atkal tiek detektēta kustība, robeža var bez krasiem lēcieniem mainīties no iepriekš atrastās, saglabātās pozīcijas.

Kad ir zināma iespējamās automašīnas kontūra atrašanās vieta eksistējoša intervāla apgabalā, tiek meklēts, vai šo kontūru apgabalā ir zīmīgas, no tukša ceļa atšķirīgas robežas, izmantojot sliekšņoto robežu līniju. Katrā mašīnas pusē (labajā un kreisajā), sākot no kontūra robežas, tiek skatīts apgabals, kas ir 1/6 no mašīnas platuma. Ja, piemēram, kreisās puses apgabalā vairāk kā 2/3 robežu līnijas pikseļu ir pārvarējuši slieksni, tiek uzskatīts, ka ir atrasta kreisā automašīnas robeža. Ja atrasta tikai viena no divām robežām, tiek uzskatīts, ka kontūra robežu parametrs vidēji ticami norāda uz automašīnas atrašanos intervālā, savukārt, ja atrastas abas robežas, parametrs ir ticams. Jaunais intervāla slēgšanas parametrs iestrādāts programmā, papildinot iepriekš izmantoto parametru klāstu (intensitāte, krāsa robežas visa intervāla apgabalā).

Adaptīvu sliekšņu iegūšana

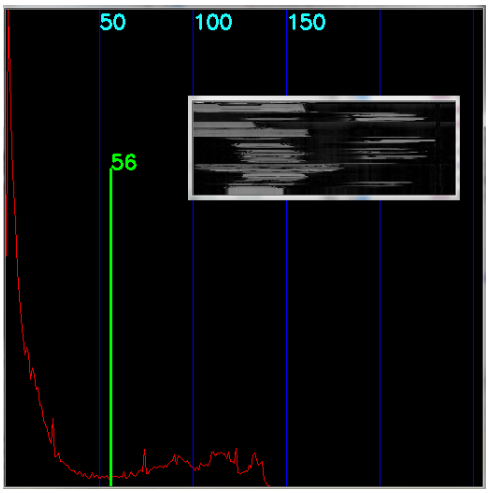
Gan intensitāšu, gan robežu, gan krāsu komponenšu un blakus kadru intensitāšu starpību līniju apstrādē nepieciešams pielietot sliekšņus, kas noteiks, kuri pikseļi ir ņemami vērā pie intervālu slēgšanas. Sliekšņa vērtībai jābūt tādai, lai attēlā, kurā redzama, piemēram, šī brīža intensitātes līnijas starpība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

Automātiskai sliekšņu atrašanai tika apskatīta Otsu metode. Metode izmanto attēla histogrammu, lai sadalītu attēlu priekšplāna 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 svari W_p un W_f, kas priekšplānam ir tajā ietilpstošo pikseļu skaita attiecība pret kopējo attēla pikseļu skaitu – līdzīgi iegūstams fona svars. Katrai sliekšņa vērtībai tiek iegūta kopējā dispersija pēc formulas 4.2.

$$\sigma_k^2 = W_p \cdot \sigma_p^2 + W_f \cdot \sigma_f^2 \tag{4.2}$$

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

Tā kā mūsu algoritma gadījumā slieksnis tiek pielietots vienai līnijai, kurā var gan būt, gan nebūt priekšplāna objekti, Otsu metodes pielietošanai 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 4.7.a redzams no vektorā saglabātajām intensitātes līnijām izveidotais attēls, kā arī šī attēla histogramma, kurā iezīmēta automātiski atrastā sliekšņa ar vērtibu 56 atrašanās vieta.



Att. 4.7: Otsu metodes pielietošana uzkrātam līniju attēlam, šī attēla histogramma un atrastais slieksnis

Eksperimentos secināts, ka Otsu metodes pielietošana atrod labu slieksni robežu pazīmju izdalīšanai, tomēr intensitātes un krāsu komponenšu gadījumos atrastais slieksnis ne vienmēr bija optimāls. Tādēļ 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 slieksni pārvarējušu pikseļu. Citos gadījumos slieksnis tiek pielāgots iegūtajiem rezultātiem – ja ārpus intervāliem ir vairāk slieksni pārvarošu pikseļu nekā intervālu apgabalos vai ārpus intervāliem šādi pikseļi aizņem vairāk kā noteiktu daļu no ārpus intervālu kopējā apgabala, tad slieksnis tiek katrā kadrā palielināts par 1. Savukārt, ja ārpus intervāliem ir maz slieksni pārvarējušo pikseļu, kā arī intervālu apgabalos to ir maz, slieksnis tiek samazināts par 1.

Aizklāšanās atklāšana

Būtisks kļūdu avots izstrādātajam algoritmam ir tuvu braucošas automašīnas. Tā kā kamera nav novietota precīzi ar skatu pret ceļu, bet gan ir vērsta ar leņķi pret to, kadrā tuvu braucošas mašīnas var aizklāt viena otru. Tas, savukārt, noved pie tā, ka vairākām automašīnām var tikt izveidots viens kopīgs intervāls uz atklāšanas līnijas, līdz ar to iegūtais automašīnu skaits ir kļūdains.

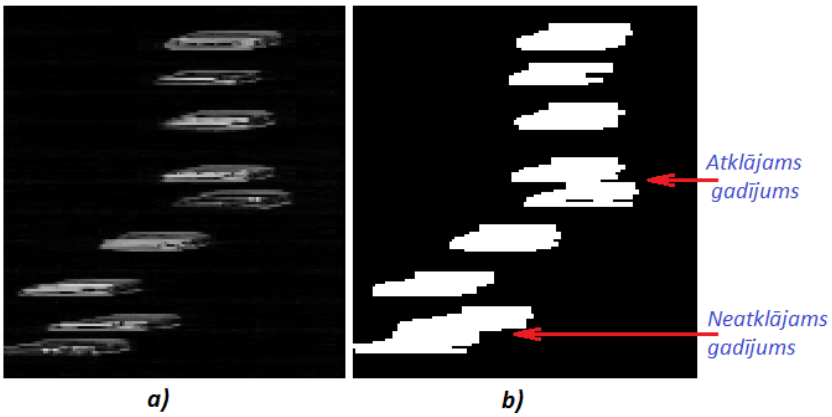
Kopīgs intervāls var tikt izveidots arī mašīnām, kas kadrā ir tuvas, bet neaizklāj viena otru. Tā kā iepriekšējā algoritma versijā intervāls pastāvēšanas laikā var palikt tikai platāks, tad, mašīnai izbraucot no atklāšanas līnijas, intervāls var būt lielāks par apgabalu, kas uz līnijas atbilst automašīnas beigu posmam. Šādam intervālam ir lielāka iespējamība apvienoties ar citai, tuvu braucošai mašīnai atbilstošo intervālu. Lai samazinātu šo iespējamību, algoritms modificēts, ļaujot intervālam palikt šaurākam, ja mašīnas platākais posms līniju jau ir šķērsojis.

Sašaurināšanas pamatā ir jau iepriekš aprakstītais paņēmiens robežu atrašanai starp mašīnu un ceļu intervāla apgabalā. Zinot šīs robežas, intervāla gala koordinātes var nomainīt uz šo robežu koordinātēm, tādējādi izsekojot mašīnas platuma izmaiņai uz atklāšanas līnijas.

Informāciju par platuma izmaiņu var izmantot specifisku mašīnu aizklāšanās gadījumu atklāšanai. Mašīnai pametot atklāšanas līniju, atbilstošais intervāls bieži paliek šaurāks. Ja vairākas pēdējās intervāla platuma izmaiņas ir bijušas sašaurināšanās, bet tad intervāls ir pēkšņi palicis platāks, un, gan pirms, gan pēc paplašināšanās intervāls ir pastāvējis gana ilgi un bijis gana plats, lai atbilstu atsevišķām automašīnām, tad tiek pieņemts lēmums, par aizklāšanās esamību. Šādā gadījumā algoritms pieņem lēmumu, ka mašīnu skaits jāpalielina par divām. Šāda aizklāšanās gadījuma piemērs parādīts attēlā 4.8, kurā attēlotas uz atklāšanas līnijas izveidoto intervālu platumu izmaiņas 150 kadru laikā. Šajā pašā

attēlā parādīts arī aizklāšanās gadījums, kuru pēc aprakstītā algoritma nevar atklāt. Attēlos x asij atbilst pikseļi uz atklāšanas līnijas, y asij atbilst laiks – augšējā attēlu rinda atbilst šī brīža kadram, pēdējā rinda atbilst pikseļiem uz atklāšanas līnijas pirms 150 kadriem.

Algoritms tālāk paplašināts, lai tiktu atklāti arī gadījumi, kad viena otru aizklāj vairāk kā divas automašīnas. Šādos gadījumos intervālam būs vairāk pēkšņu paplašinājumu pēc sašaurināšanas periodiem.



Att. 4.8: Aizklāšanās gadījumi a) intensitātes izmaiņas uz līnijas 150 kadros; b) intervālu platuma izmaiņas 150 kadros

Mašīnu ēnu atklāšanas metode

Automašīnām atbilstošie intervāli uz atklāšanas līnijas var nesakrist ar automašīnas platumu, ja kadrā redzamajām mašīnām uz sāniem (perpendikulāri līnijai) ir izteikta un gara ēna, kuru algoritms uzskata par kustīgā objekta sastāvdaļu. Ēnu piemēri uz intensitātes līnijas 150 kadros parādīti attēlā 4.9.a. Papildus nepatiesam mašīnas platuma novērtējumam, šāda ēna var novest pie dažādām mašīnām atbilstošo intervālu apvienošanās, jo mašīnas mestā ēna var pārklāties ar blakus braucošu mašīnu. Tādēļ tika izstrādāts paņēmiens ēnu atklāšanai.

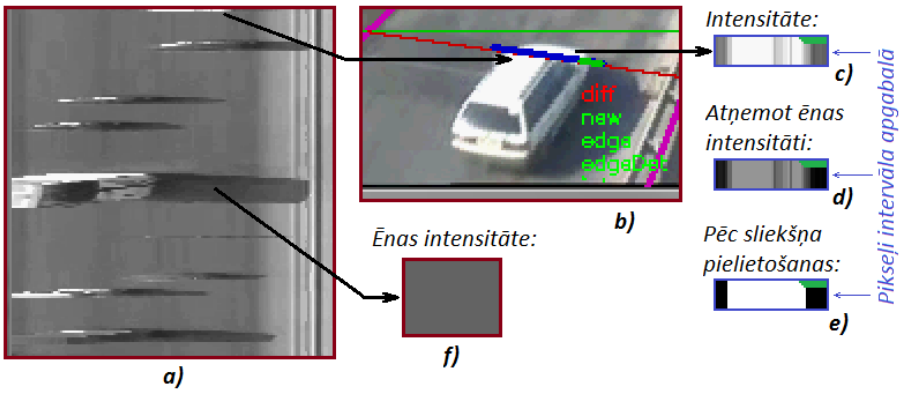
Metodē tiek noteikta ēnai atbilstošā vidējā pikseļu intensitāte un ēnas virziens (kurā pusē mašīnai ēna atrodas). Mašīnai atbilstošā intervāla apgabalā (attēls 4.9.b) tiek skatītas pikseļu intensitātes vērtības – no melnbaltās atklāšanas līnijas tiek izgriezts intervālam atbilstošais apgabals (attēls 4.9.c). Tālāk algoritms darbojas dažādi, atkarībā no tā, vai jau ir zināma ēnai atbilstošā intensitāte un virziens.

Ja ēnas parametri vēl nav noteikti, intensitātes intervāls tiek sliekšņots izmantojot Otsu metodi automātiskai sliekšņu noteikšanai. Sliekšņota intervāla piemērs parādīts attēlā 4.9.e. Gadījumos, kad intervāla apgabalā ir gan mašīna, gan arī tās mestā ēna, tiek pieņemts, ka ēnas intensitāte ir mazāka par mašīnas pikseļu intensitāti (tumšu mašīnu gadījumā pieņēmums var neizpildīties). Šādā gadījumā automātiskais slieksnis, atdalot gaišākus priekšplāna pikseļus no tumšākiem fona pikseļiem var izveidot baltus pikseļus automašīnas apgabalā un melnus pikseļus ēnas apgabalā, kā parādīts attēlā 4.9.e. Ja baltajā apgabalā ir nelieli melni pārrāvumi, tos var aizpildīt lietojot morfoloģiskās aizvēršanas operāciju, kurā par masku izmantota $1 \times n$ izmēra matrica, kur no n lieluma atkarīgs, cik pikseļu plati pārrāvumi tiks aizvērti.

Pēc sliekšņa pielietošanas intervālā tiek meklēti platākie nepārtrauktie baltie un melnie apgabali, kas var potenciāli atbilst automašīnai un ēnai. Gan baltais, gan melnais apgabals tiek salīdzināti ar platuma konstanti (minimālo iespējamo automašīnas platumu). Ja abi apgabali pārsniedz minimālo platumu, tiek pieņemts lēmums, ka šī brīža kadrā intervāla apgabalā atrodas gan automašīna, gan ar tās platumu salīdzināma (tātad vērā ņemama) ēna. Mašīnai un ēnai atbilstošo apgabalu sākuma un beigu koordinātes tiek saglabātas. Šādā veidā ēna tiks atklāta tikai gadījumos, kad mašīna ir gaišāka par ēnu, tādēļ šī metode tiek pielietota tikai, kamēr nav noteikta ēnas vidējā intensitāte un virziens. Attēlos 4.9.b, 4.9.c, 4.9.d, 4.9.e atklātais ēnai atbilstošais apgabals atzīmēts ar biezu, zaļu līniju

Kad atrasti plati ēnas un mašīnas apgabali, tiek noteikta ēnas vidējā intensitāte. Vispirms, ja tas iespējams, slieksni nepārvarējušais ēnas apgabals tiek sašaurināts par 5 pikseļiem no abiem galiem, lai izvairītos no ēnas apgabalā iekļuvušām tumšākām mašīnas vai daļēji apēnota ceļa daļām. Sašaurinātā

apgabala pikseļiem tiek aprēķināta pikseļu vidējā intensitāte. Šādas intensitātes piemērs, kas iegūts no ieejas kadriem 4.9.a, parādīts attēlā 4.9.f. Ēnas virziens tiek atrasts, salīdzinot plato baltā un melnā apgabalu sākuma koordinātes – ja baltais apgabals sākas ātrāk par melno apgabalu, ēna atrodas pa labi no mašīnas.



Att. 4.9: Ēnu atklāšanas posmi

Kad zināms ēnas virziens, tiek skatīts, vai sliekšņotajā apgabalā aiz ēnas neparādās vēl kāds neliels balts apgabals. Šāds apgabals var atbilst ceļa apgabalam aiz ēnas, kas ietilpst eksistējošā intervālā uz atklāšanas līnijas. Apgabals intervālā var būt iekļauts, piemēram, ja mašīnas intervāls nav ticis sašaurināts, kad tas būtu bijis nepieciešams. Ja šāds neliels apgabals aiz ēnas tiek atklāts, tiek pieņemts lēmums sašaurināt mašīnai atbilstošo intervālu, neiekļaujot lieko apgabalu.

Atrastie ēnas parametri (intensitāte un virziens) netiek uzreiz izmantoti turpmāko ēnu atklāšanai. Vispirms tiek uzkrāta informācija par vairākām atklātajām ēnām. Kad ēna tikusi detektēta noteiktam skaitam automašīnu, no visām ēnu intensitātēm un virzieniem tiek aprēķinātas vidējās vērtības. No šī brīža algoritms turpmākajām automašīnām ēnu detektēšanai ir atšķirīgs no iepriekš aprakstītā.

Pie zināmiem ēnu parametriem, no intervāla apgabalā esošo pikseļu intensitātes pirms sliekšņošanas tiek atņemta ēnas vidējā intensitāte – attēla 4.9.c un intensitātes 4.9.f starpības absolūtā vērtība attēlota 4.9.d. Starpības absolūtās vērtības būs mazākas tiem pikseļiem, kas pēc intensitātes mazāk atšķiras no ēnas vidējās intensitātes, tādējādi par ēnu tumšāku mašīnu pikseļi pēc sliekšņošanas arī ir atšķirami no ēnas.

Pēc sliekšņa pielietošanas vairs netiek meklēts lielākais melnais apgabals intervālā, bet gan melns apgabals zināmajā intervāla pusē, kas turpmāk tiek pieņemts par ēnu, un, kura koordinātes tiek saglabātas. Nākamajos intervāla pastāvēšanas kadros šis ēnas apgabals tiek modificēts, salīdzinot saglabātās koordinātes ar jaunajām, kas jaunajos kadros atrastas ar to pašu sliekšņošanas metodi. Saglabātās ēnas sākuma vai beigu koordinātes tiek mainītas tikai tad, ja šī izmaiņa blakus kadros nav pārāk liela, tātad tiek ņemts vērā, ka ēnas izmērs konkrētai automašīnai var mainīties, bet tas nevar izmainīties par pārāk lielu vērtību blakus kadros. Ja kādā kadrā parādās plats, balts apgabals aiz ēnas apgabala (pretējā pusē mašīnas apgabalam), tas var liecināt par vēl vienas mašīnas iekļaušanu intervālā. Ja, savukārt, aiz ēnas apgabala ir īss balts apgabals, kam atkal seko melns apgabals, tiek pieņemts, ka ticamāks ēnai atbilstošais apgabals ir tas, kurš atrodas tuvāk tai intervāla malai, uz kuru norāda zināmais ēnas virziens.

Katrā kadrā tiek atrasta arī aktuālā ēnas vidējā intensitāte. Šis lielums tiek izmantots, lai pārrēķinātu jau zināmo ēnas intensitāti, kas ļauj ēnas intensitātei pielāgoties mainīgajiem āra apstākļiem kadrā. Tāpat kā iepriekš, tiek arī atklāti un no intervāla izslēgti nelieli, mašīnai un ēnai nepiederoši apgabali aiz ēnām, kas varētu būt nevajadzīgi iekļauti intervālā

Ja vairākām automašīnām pēc kārtas netiek atklātas ēnas ne vienu brīdi, kamēr tās šķērso atklāšanas līniju, tiek pieņemts lēmums, ka ēnu šobrīd kadrā nav. Šajā gadījumā tiek arī atmesta informācija par ēnas intensitāti un virzienu. Turpmāk ēnas tiek atklātas pēc iepriekš aprakstītā algoritma, kas neizmanto informāciju par ēnas intensitāti un virzienu, līdz šie lielumi atkal tiek noteikti.

4.3.3 Mašīnu atklāšana naktī

Nakts laikā kadrā redzamākie objekti ir automašīnu lukturi, to atspīdums uz ceļa, kā arī gaismas ceļa malā vai citi gaismas avoti. Mašīnas korpuss nav labi saskatāms, ja vien ceļš nav labi apgaismots. Tā kā mašīnu lukturu atstarojumi uz ceļa ir ar lielāku intensitāti kadrā nekā pati mašīna, var rasties problēmas ar esošā algoritma pielietošanu nakts režīmam. Algoritms jāpielāgo lukturu atklāšanai, bet, lai lukturi būtu vienīgais redzamais objekts, tika veikti eksperimenti, lai atrastu nakts apstākļiem piemērotus kameras parametrus.

Attēlā 4.10.a parādīts krāsainais ieejas kadrs, kad kameras parametri nav mainīti. Gan mašīnas lukturi, gan lampas ceļa malā šajā attēlā ir pārgaismoti – atbilstošajiem pikseļiem attēlā ir maksimālā intensitātes vērtība 255. Tātad, mainot kameras parametrus iespējams samazināt kameras jutīgumu pret gaismu tādējādi, ka citiem objektiem kļūstot tumšākiem, lukturiem joprojām atbildīs liela intensitāte.

Tīkla kamerā AXIS P1344 ir iestrādāts nakts režīms, kurā tiek atslēgts gaismas infrasarkanās spektra daļas filtrs – šāda režīma piemērs parādīts attēlā 4.10.b Šajā režīmā zūd informācija par krāsām, kas naktī tāpat nav aktuāla. Ar izslēgtu IR filtru joprojām gaismas avoti kadrā, kā arī lukturu atspīdums uz ceļa ir pārgaismoti.

Lai samazinātu kameras uztvertās gaismas daudzumu, tika mainīti kameras aizvara ātrumi. Jo mazāk laika aizvars ir atvērts, jo mazāk gaismas nokļūst uz kameras sensora. Noklusētais maksimālais aizvara ātrums ir 1/60 sekundes. Tika veikti eksperimenti, samazinot šo ātrumu. Attēlos 4.10.c un 4.10.d parādīti piemēri ar aizvara ātrumu 1/6000 un 1/24500 sekundēm. Secināms, ka ar mazāko aizvara ātrumu lukturi kadrā vairs nav pārgaismoti. Citi objekti kadrā vairs nav saskatāmi, tādēļ šāds kameras režīms turpmāk tiks izmantots, lai naktīs skaitītu automašīnas, atklājot to lukturus.

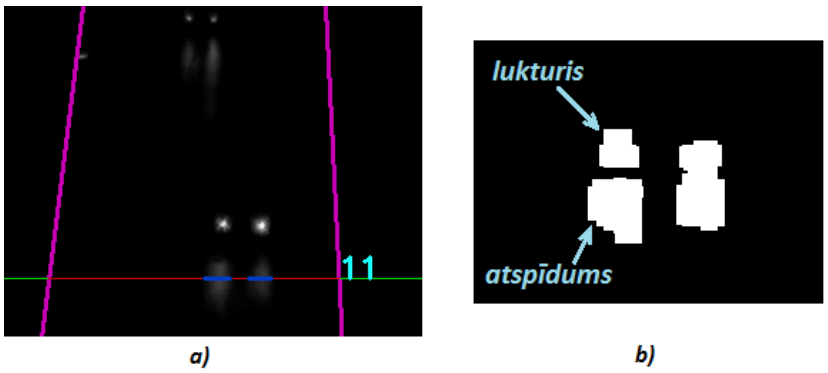
Att. 4.10: Kameras parametru maiņa nakts režīmā

Balstoties uz ieejas kadriem, kuros gan sausā, gan slapjā laikā pārsvarā vienīgie redzamie objekti ir automašīnu lukturi, automašīnu atklāšanas algoritms tika modificēts, ļaujot skaitīt automašīnas naktī.

Intervālu atklāšana uz līnijas paliek tāda pati kā dienā ar atsevišķām izmaiņām. Naktī tiek saskaitīti arī īsāki intervāli (minimālais intervāla platums, kas jau var atbilst automašīnai ir 3 pikseļi). Lai atklātu uz līnijas kustību vai fonam nepiederošus objektus, naktī tiek lietoti mazāki sliekšņi. Tā kā fons ir melns lielā kameras aizvara ātruma dēļ, un, laikā fons praktiski nemainās, tas tiek pieņemts par nemainīgu, kā arī tiek lietota nemainīga sliekšņu vērtība. Eksperimentāli izvēlēta sliekšņa vērtība 5 tiek lietota gan kustības, gan objektu atklāšanai pēc dažādiem parametriem.

Slēdzot kādu no intervāliem, tā koordinātes vēl tiek saglabātas vairākus kadrus, un, tās tiek salīdzinātas ar citu slēgto intervālu koordinātēm šajos kadros. Tādējādi tiek meklēti lukturi pāri, kas raksturīgi automašīnām. Pāri veido tie slēgtie intervāli, kuri tika slēgti ar nelielu laika atšķirību, un, kuru koordinātes ir pietiekami tuvas, lai iespējamās automašīnas (abu intervālu apvienojuma) platums nepārsniegtu iestādītu slieksni. Pāris tiek izskaitīt kā viena automašīna. Ja intervāls ir bijis slēgts pirms vairāk kā 10 kadriem, un, nav ticis atrasts pāris, tiek atklāts un kā transportlīdzeklis saskaitīts viens lukturis - šādā veidā var skaitīt motociklus.

Kameras aizvara ātruma palielināšana ne visos gadījumos pilnībā atrisina lukturu atspīduma problēmu, kas var parādīties uz slapjiem ceļiem (attēls 4.11.a). Tas tiek ņemts vērā mašīnu skaitīšanas algoritmā. Ja dažus kadrus pēc intervāla slēgšanas uz tām pašām atklāšanas līnijas koordinātēm tiek izveidots vēl viens intervāls, tiek pieņemts, ka iepriekš slēgtais intervāls ir bijis atspīdums, un kļūdaini palielinātais automašīnu skaits tiek samazināts par vienu. Algoritms balstās uz pieņēmumiem, ka lukturu atspīdums uz mašīnas braukšanas virzienam perpendikulāras atklāšanas līnijas atrodas tajā pašā



Att. 4.11: Automašīnu skaitīšana naktī, pie traucējoša lukturu atspīduma uz ceļa

apgabalā kur pats lukturis, un, ka attālums starp lukturi un tā atspīdumu ir mazāks, kā starp vienai aiz otras braucošu automašīnu lukturiem. Atklātais un pie mašīnām nepieskaitītais lukturu atspīdums parādīts attēlā 4.11.b., kas ir laika-telpas attēls, un, kas atbilst vairākiem kadriem, kuros bildē 4.11.a paradītā automašīna ir pilnībā šķērsojusi atklāšanas līniju. Attēlā redzamais kreisā luktura atspīdums tiek izskaitīts kā automašīna, tomēr, pēc paša luktura intervāla slēgšanas un pieskaitīšanas, tiek atklāts, ka pirmais intervāls atbilst atspīdumam, tādēļ atklāto automašīnu skaits tiek par vienu samazināts. Attēla labās puses intervālā atspīdums un lukturis ir saplūduši par vienu atklātu objektu, tādēļ nenotiek nedz lieka mašīnas pieskaitīšana, nedz arī to labojošā atskaitīšana.

Lai automātiski noteiktu nakts režīma iestāšanos (kad kamerai tiek uzstādīts maksimālais aizvara ātrums), tiek pārbaudīts katrs melnbaltais kadrs. Tiek iegūta kadra pikseļu intensitāšu summa, un, tā tiek dalīta ar kadra pikseļu skaitu. Ja šis dalījums pārsniedz noteiktu sliekšņa vērtību, tiek pieņemts lēmums, ka pētāmajā kadrā ir pietiekoši daudz gaišu pikseļu, un, tiek novērota dienas aina. Piemēram, kadrā ar izmēru 800×600 pikseļu ir 5 automašīnas. Ņemot vērā, ka nakts režīmā pārsvarā redzami tikai mašīnu lukturi, un par piemēru pieņemot, ka lukturu diametrs ir 18 pikseļi un maksimālā iespējamā intensitāte ir 255, tiek iegūta aptuvena attiecība:

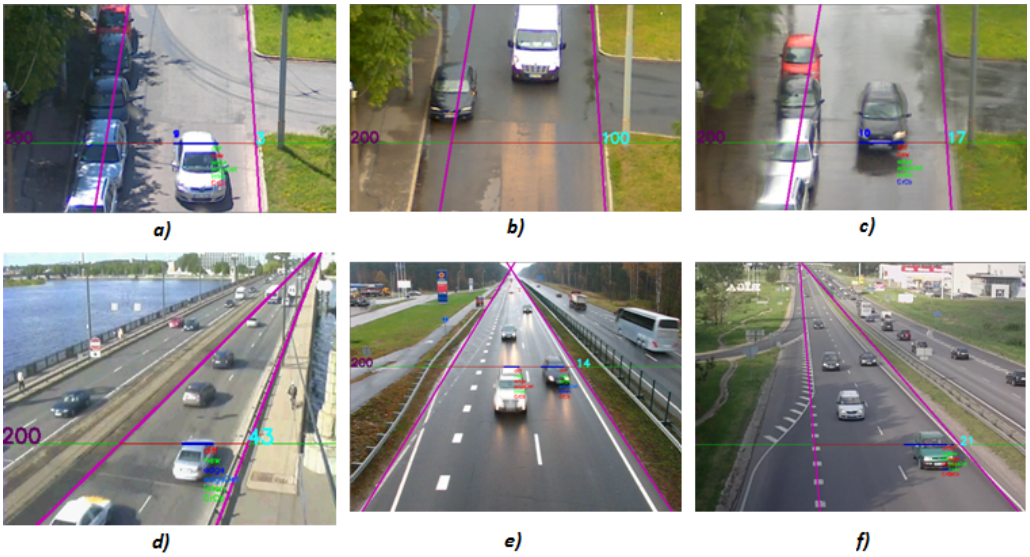
$$\frac{2 \cdot \pi \cdot 9^2 \cdot 10 \cdot 255}{800 \cdot 600} \approx 2.7, \tag{4.3}$$

Attiecības 4.3 rezultāts ir lielāks salīdzinot ar to, kas biežāk parādās praksē, jo lukturiem pieņemta maksimālā intensitāte 255 - reāli intensitāte būs mazāka, jo kameras aizvara ātrums ir pieskaņots tam, lai lukturi kadrā nebūtu pārgaismoti. Dienas laikā visbiežāk katram no 480000 pikseļiem būs ne nulles vērtība, līdz ar to, pat ja katra pikseļa intensitāte ir tikai 10 (kas dienas laikā nav reālistiski), tad attiecība jau būs 10. No šiem apsvērumiem, par sliekšņa vērtību, kas nosaka dienas/nakts režīmu pieņemts 10.

4.3.4 Testi ar uzlaboto algoritmu mašīnu atklāšanai

Papildinātā algoritma mašīnu skaitīšanas precizitāte pārbaudīta uz vairākiem video, dažādos laika apstākļos un pie dažādas satiksmes plūsmas intensitātes. Testi veikti uz attēlā 4.12 parādītajiem video. Gadījumos 4.12.a, 4.12.b un 4.12.c testi ir veikti uz reālā laikā pienākošiem video. Šim nolūkam programmā iestrādāta iespēja saņemt datus no tīmekļa kameras AXIS P1344. Testi veikti dažādos laika apstākļos, kur gadījumā *b* ceļš ir slapjš un uz tā ir intensīvs lukturu atspīdums, savukārt *c* gadījumā ir stiprs lietus. 4.12.d, 4.12.e un 4.12.f ir testi ar lielāku satiksmes plūsmu.

Testu rezultāti apkopoti tabulā 4.13, kurā algoritma atklāto automašīnu skaits salīdzināts ar reālo atklāšanas līniju šķērsojušo automašīnu skaitu. Video *c* gadījumā skaita precizitāte ir uzrādīta 100%, lai gan algoritms ir pieļāvis 6 kļūdas – šeit kļūdas ir izslēgušas viena otru, līdz ar to iegūtais automašīnu skaits atbilst reālajam. Lielākais kļūdu skaits ir *d* video gadījumā, ko nosaka vairākas šī video īpašības. Video *d* automašīnām ir izteiktas ēnas, un, tā kā tiek skatītas divas braukšanas joslas, ēnu dēļ blakus braucošām mašīnām var tikt izveidots kopīgs intervāls. Aizklāšanos šajā video veicina arī kameras slīpais novietojums – ja pa labo joslu brauc autobuss, tas pilnībā aizsedz skatam blakus joslu, līdz ar ko atkal parādās aizklāšanās, kas šajā video bija visbiežākais kļūdu iemesls. Bez aizklāšanās un apvienošanas bija sastopami arī retāki kļūdu iemesli, piemēram, ceļu šķērsojoši cilvēki, vai gar kameru lidojoši putni.



Att. 4.12: Algoritma testēšanai izmantotie video

Video	Reālais mašīnu skaits	Atklātais mašīnu skaits	Kļūdas			Skaitīšanas precizitāte
			Aizklāšanās	Izskaitīta kā divas	Citas	
a)	100	105		5		95%
b)	100	105		4	1	95%
c)	100	100	3	3		100%
d)	77	71	7	3	2	91%
e)	42	44			2	95%
f)	48	46	2			96%

Att. 4.13: Testu rezultāti

4.4 Mašīnas izsekošana, ātruma un izmēru noteikšana

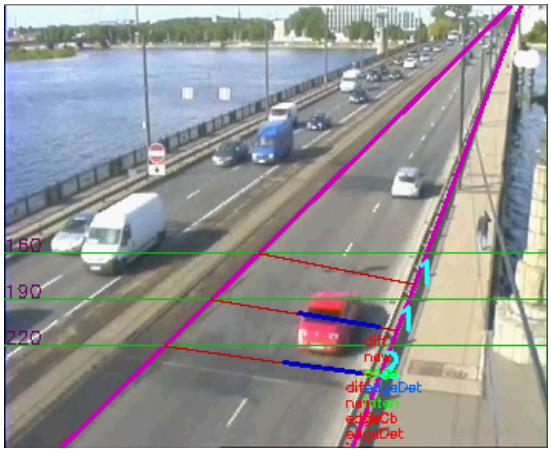
4.4.1 Atklāšanas līniju izvietošanas iespēju paplašināšana

Līdz šim automašīnu atklāšanas algoritmā tika izmantota atklāšanas līnija, kas ir horizontāla attiecībā pret kadru. Šāda veida līniju iespējams samēra vienkārši nodefinēt un izdalīt kā apgabalu no kādas kadra rindas. Tomēr, ne visos gadījumos šāda līnija ir perpendikulāra ceļam, jo kadrā ceļš var būt redzams slīpi. Tādēļ algoritmā tika iestrādāta iespēja izvietot kadrā slīpu atklāšanas līniju.

Lai izvēlētos slīpās līnijas pikseļus, var izmantot līnijas vienādojumu. Šādā gadījumā nāktos pielietot reizināšanas un dalīšanas operācijas, kas nav efektīvākās darbības attiecībā pret skaitļošanas jaudām. Lai turpinātu automašīnu atklāšanas algoritmu veidot efektīvu, slīpās līnijas izveidošanai pielietots Bresenhama algoritms, kas ļauj izvairīties no reizināšanas un dalīšanas operācijām.

Programmā iestrādāta iespēja izmantot vairākas atklāšanas līnijas vienlaicīgi. Katra no līnijām darbojas kā atsevišķs detektors, katra no tām atklāj un skaita automašīnas. Izmantojot informāciju no vairākām līnijām iespējams iegūt ticamāku automašīnu skaitu, jo atsevišķas līnijas kļūdains rādījums var tikt neņemts vērā, ja tas atšķiras no vairuma citu atklāšanas līniju rādījumiem.

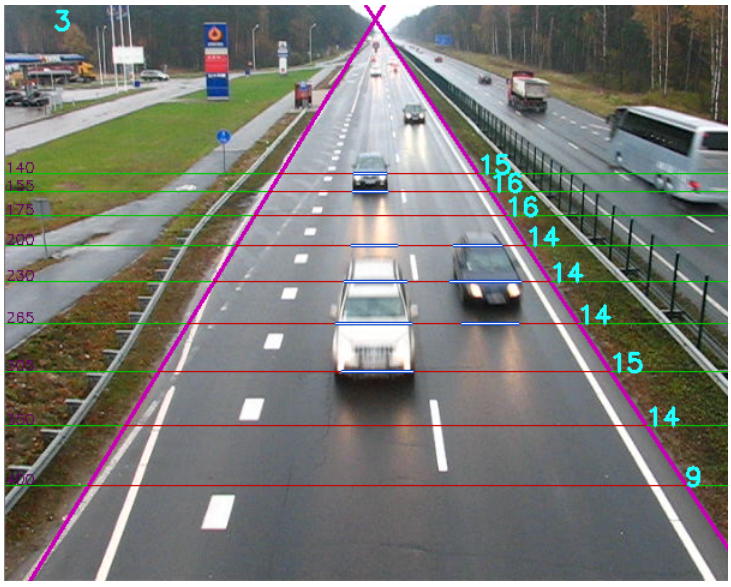
Vairāku atklāšanas līniju ērtu izvietošanu atļauj algoritma īstenošana objektorientētā kodā. Katra līnija ir detektoru klases objekts, kurš iekļauj visas nepieciešamās metodes un mainīgos. Līdz ar to, atliek nodefinēt vēlamo detektoru skaitu un attālumu starp tiem – tālāk katra atklāšanas līnija tiek apstrādāta pēc viena un tā paša algoritma. Attēlā 4.14 parādīts piemērs ar vairāku slīpu atklāšanas līniju izvietošanu kadrā.



Att. 4.14: Vairāku slīpu atklāšanas līniju izvietošana kadrā

Projekcijas transformācijas pielietojums

Piedāvātais automašīnu atklāšanas algoritms ļauj izvietot kadrā vairākas atklāšanas līnijas. Attēlā 4.15 parādīta deviņu atklāšanas līniju izvietošana kadrā. Virs katras līnijas, pa labi no kadrā redzamā ceļa, norādīts automašīnu skaits, kas iegūts uz šīs līnijas. Skaitot no kadra augšas, 2. un 3. līnija ir atklājušas 16 automašīnas. Nākošās līnijas ir atklājušas tikai 14. automašīnas, jo divas no tām vēl nav izbraukušas no zemākām atklāšanas līnijām (šo mašīnu intervāli vēl nav slēgti uz zemākām līnijā). 7. līnija uzrāda 15 atklātas automašīnas, kas nozīmē, ka skaitīšanā ieviesusies kļūda, izskaitot vienu automašīnu par daudz. Uz pirmās līnijas ir notikusi pretēja kļūda - viena automašīna ir palaista garām bez pieskaitīšanas. Secināms, ka apvienojot rādījumus no vairākām līnijām, iespējams palielināt kopējo skaitīšanas precizitāti, jo var tikt atklātas atsevišķu līniju kļūdas, salīdzinot to rādījumus ar pārējam līnijām. Zemākā no atklāšanas līnijām ir pieļāvusi daudz kļūdu - palaidusi garām 5 no 14 automašīnām. Secināms, ka tik zemu kadrā, kur automašīnu redzamais ātrums ir liels un kontūras izplūdušas, atklāšanas līniju izvietot nav ieteicams.



Att. 4.15: Vairāku atklāšanas līniju izvietošana kadrā

Lai noteiktu automašīnu parametrus, tiek apvienoti dažādo līniju intervālu, noskaidrojot, kuri intervāli pieder vienam un tam pašam transportlīdzeklim. Taisna ceļa gadījumā var izmantot priekšzināšanas, ka vienas mašīnas intervāli atrodas viens zem otra uz blakus esošām atklāšanas līnijām. Tomēr, perspektīvas dēļ, intervālu garumi atšķiras - uz augstākām līnijām izveidotie intervāli ir īsāki

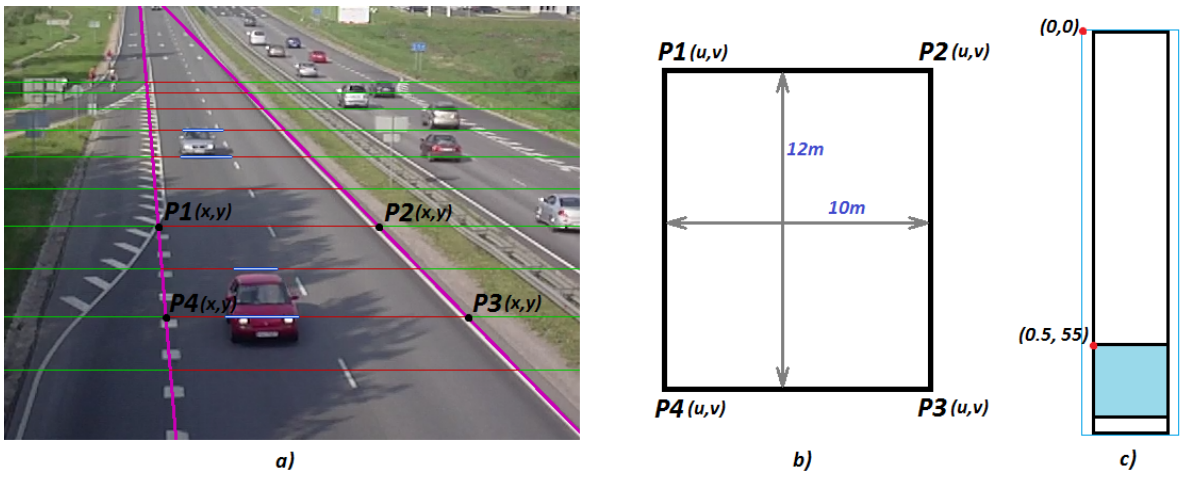
par uz zemākām līnijām izveidotajiem. Šī perspektīva arī jāņem vērā mašīnas parametru iegūšanai, jo kadrā augstāk esošās automašīnas izskatās mazākas un lēnāk braucošas nekā tās pašas automašīnas pie tā paša ātruma kadra lejasdaļā, kas atbilst tuvāk kamerai esošajam ceļa apgabalam. Šādu apsvērumu dēļ, tika izvēlēts projekcijas transformācijas lietojums, kas kadra plakni transformētu uz ceļam atbilstošo plakni. Jaunā transformācija atbilstu gadījumam, kad kamera novietota perpendikulāri virs ceļa, un, automašīnas būtu redzamas precīzi no augšas.

Projekcijas transformācijai no vienas plaknes uz citu, jāatrod pirmās plaknes punktu koordinātes (x,y) otrajā (u,v). Jaunās koordinātes iegūstamas pēc formulām 4.4, 4.5

$$u = \frac{a \cdot x + b \cdot y + c}{g \cdot x + h \cdot y + 1}, \tag{4.4}$$

$$v = \frac{d \cdot x + e \cdot y + f}{g \cdot x + h \cdot y + 1}, \tag{4.5}$$

Transformācijas koeficientus a, b, c, d, e, f, g, h var atrast, ja ir zināma vismaz četru punktu atbilstība starp divām plaknēm. Tātad katram interesējošam kameras novietojumam vispirms ir jāveic kalibrācija - vismaz četriem punktiem kadrā manuāli jāiestāda atbilstošās vērtības vēlamajā ceļa plaknē. Piemēram, ja attēlā 4.16.a pieņem, ka redzamais ceļš ir taisns un līdzens, tad punkti $P1, P2, P3$ un $P4$ transformētajā plaknē veido taisnstūri (attēls 4.16.b). Šis taisnstūris ir ceļa apgabals starp trīs atklāšanas līnijām. Lai nokalibrētu sistēmu, būtu jāizmēra attālumi dabā starp šiem četriem punktiem. Pirmajiem testiem tika pieņemti aptuveni attālumi - 10 metri starp punktiem $P1$ un $P2$ un 12 metri starp $P1$ un $P4$. Transformācija tiks veikta visam pētāmajam ceļa apgabalam, tādēļ par koordinātēm punktam $P1(u, v)$ šajā piemērā tiek pieņemtas vērtības $u = 0.5m$ (ceļa kreisā mala sāksies 0,5m attālumā no jaunās plaknes pirmās kolonnas) un $P1 = 55m$ (pieņemta pietiekama rezerve ceļa posmam virs punkta P1). Kalibrācijā izmantotā četrstūra atrašanās vieta transformācijas plaknē visam ceļam attēlota 4.16.c.



Att. 4.16: Sistēmas kalibrēšana, lai pēc projekcijas transformācijas iegūtu plakni, kurā interesējošais ceļa apgabals ir taisnstūris

Pēc kalibrācijas, transformācijas koeficientu vērtības iegūstamas ar vienādojumu sistēmu:

$$\begin{aligned} ax_1 + by_i + c = gx_i + u_i + hy_iu_i + u_i \\ dx_1 + ey_i + f = gx_i + v_i + hy_iv_i + v_i \end{aligned} \tag{4.6}$$

Sistēmā 4.6 ievietojot zināmos x, y, u, v, iegūst atrisināmu sistēmu:

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0u_0 & -y_0u_0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0v_0 & -y_0u_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1u_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n-1} & y_{n-1} & 1 & 0 & 0 & 0 & -x_{n-1}u_{n-1} & -y_{n-1}u_{n-1} \\ 0 & 0 & 0 & x_{n-1} & y_{n-1} & 1 & -x_{n-1}v_{n-1} & -y_{n-1}u_{n-1} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \\ u_1 \\ v_1 \\ \vdots \\ u_{n-1} \\ v_{n-1} \end{bmatrix} \tag{4.7}$$

Koeficientu aprēķins īstenots kodā, un, tagad pirmais solis ko veic automašīnu atklāšanas algoritms ir projekcijas transformācijas koeficientu aprēķins. Kodā īstenota arī metode, kas (x, y) plaknē uzdotās koordinātes pārveido uz (u, v) plaknes koordinātēm. Lai pārietu no ceļa plaknes koordinātēm atpakaļ uz kadra plaknei atbilstošajām vērtībām, veic inverso projekcijas transformāciju. Šajā gadījumā vienādojumu sistēmā 4.6 var apmainīt vietām (x, y) un (u, v) koordinātes. Tomēr kodā īstenota cita inversās transformācijas koeficientu $a_{inv}, b_{inv}, c_{inv}, d_{inv}, e_{inv}, f_{inv}, g_{inv}, h_{inv}$ atrašanas metode, kas balstās uz atrastajiem tiešās transformācijas koeficientiem:

$$\begin{aligned} a_{inv} &= \frac{e - f \cdot h}{a \cdot e - b \cdot d} \\ b_{inv} &= \frac{c \cdot h - b}{a \cdot e - b \cdot d} \\ c_{inv} &= \frac{b \cdot f - c \cdot e}{a \cdot e - b \cdot d} \\ d_{inv} &= \frac{f \cdot g - d}{a \cdot e - b \cdot d} \\ e_{inv} &= \frac{a - c \cdot g}{a \cdot e - b \cdot d} \\ f_{inv} &= \frac{d \cdot c - a \cdot f}{a \cdot e - b \cdot d} \\ g_{inv} &= \frac{d \cdot h - e \cdot g}{a \cdot e - b \cdot d} \\ h_{inv} &= \frac{b \cdot g - a \cdot h}{a \cdot e - b \cdot d} \end{aligned} \tag{4.8}$$

Atklāšanas līniju automātiska izvietošana

Pēc transformācijuas koeficientu iegūšanas, satiksmes analīzes algritms spēj izvietot līnijas kadrā tādējādi, lai pēc projekcijas transformācijas starp visām līnijām būtu vienādi attālumi, kā arī, lai tās visas būtu perpendikulāras ceļam. Šāds līniju izvietojums īstenots attēlā 4.18.

Lai automātiski panāktu šādu līniju izvietojumu algoritms aprēķina katras atklāšanas līnijas atrašanās vietu kadrā un to leņķi pret kadra rindām. Katrai līnijai šis leņķis var atšķirties, ja kamera stāv ceļa malā, nevis virs tā.

Kadrā pašas augstākās atklāšanas līnijas sākuma koordināte tiek iestādīta manuāli. Līnijas beigu koordinātei jāatrodas uz ceļa robežas (kas arī jau ir iestādīta). Lai atrastu precīzo rindas koordināti uz iestādītās robežas, līnijas sākuma koordinātei tiek veikta projekcijas transformācija. Sākuma punkta projekcijas transformācijas y (rindas) koordinātei jāsakrīt ar beigu punkta y koordināti, lai līnija būtu paralēla x asij. Ceļa robežas transformācijas projekcija ir paralēla y asij. Tādēļ projekcijas transformācijā atklāšanas līnijas gala punktam tiek piešķirta sākumpunktam atrastā y koordināte, un par x koordināti pieņem ceļa robežas kolonnu. Veicot atrastajam punktam inverso projekcijas transformāciju, tiek iegūts atklāšanas līnijas beigu punkts kadra plaknē. Līdz ar to, kadrā iespējams izvietot ceļam perpendikulāro līniju. Lai iegūtu pārējās līnijas vienādos attālumos vienai no otras, projekcijas transformācijas plaknē (reālā ceļa plaknē), nemainot atklāšanas līniju galapunktu x koordinātes, y koordinātēm (gan līnijas sākumam gan galam) tiek pieskaitīts vēlamais attālums starp līnijām. Izmainot galapunktu y koordinātes un veicot inverso projekcijas transformāciju, tiek iegūta nākamā līnija kadra plaknē.

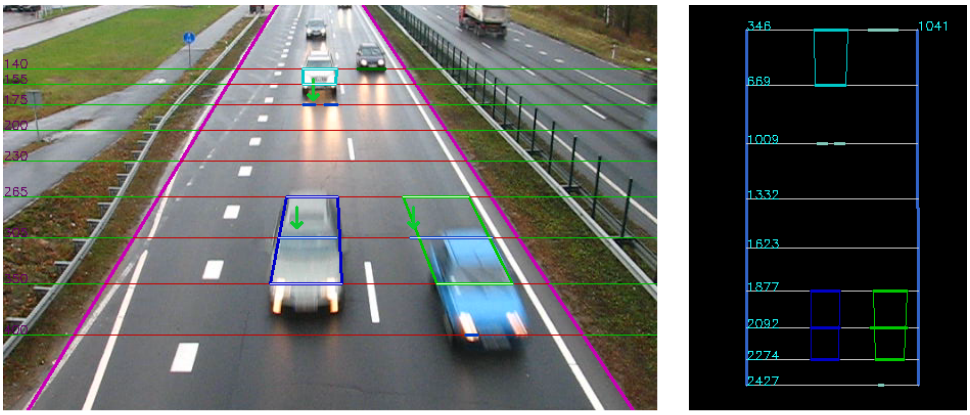
4.4.2 Vairāku līniju rādījuma apvienošana

Automašīnu atklāšanas algoritmā, projekcijas transformācija tiek pielietota pēc tam, kad ir atklāti intervāli uz līnijas. Transformācija nav prasīga pret skaitļošanas jaudām, jo katram intervālam tiek transformētas tikai gala koordinātes. Jaunās intervālu koordinātes tālāk tiek apstrādātas, ar mērķi noteikt intervālus, kuru veidošanos izraisījusi viena un tā pati automašīna. Ja tas ir noteikts, intervāls tiek pievienots $Vehicle$ klases objektam. Šīs klases objekti ir automašīnas, kuru mainīgie ir mašīnu veidojošie intervāli, mašīnas koordinātes, parametri.

Kad uz līnijas paradās jauns intervāls, tas nepieder nevienai automašīnai, ka arī netiek uzskatīts par pirmo izveidojušos intervālu jaunai automašīnai. Intervāls tiek salīdzināts ar jau esošo automašīnu intervāliem. Ja intervāls atrodas uz atklāšanas līnijas, kas ir blakus citai līnijai, uz kuras ir mašīnai piederošs intervāls, un, ja šo intervālu gala koordināšu starpību absolūtās vērtības nepārsniedz sliekšņa vērtību, tiek pieņemts lēmums, ka jaunais intervāls ir pievienojams jau eksistējošai automašīnai. Savukārt, ja tuvumā jaunajam intervālam nav nevienas derīgas mašīnas, tad intervāls tiek uzskatīts par iespējamu automašīnu. Nākamreiz, kad parādīsies jauns intervāls, tas tiks salīdzināts gan ar automašīnām, gan ar intervālu, kas ir iespējamā automašīna. Ja jaunais intervāls tiek pievienots iespējamai automašīnai, tā kļūst par pilnvērtīgu mašīnu ar diviem intervāliem.

Attēlā 4.17.a parādīti vairāki intervāli uz atklāšanas līnijām un attēlā 4.17.b ir šo līniju un uz tām esošo intervālu projekcijas transformācijas rezultāts. Vienas un tās pašas automašīnas intervāli transformācijas

attēlā ir vienā krāsā, tātad šobrīd algoritms ir saskaitījis 3 mašīnas un vienu iespējamo mašīnu ar līnijām noklātajā kadra apgabalā. Automašīnas objektam ir arī četras stūru koordinātes, starp kurām novilktas taisnes, kas veido attēlos redzamo automašīnas rāmi. Mašīnas stūru koordinašu aprēķinā tiek ņemtas vērā visu mašīnā esošo intervālu koordinātes.



Att. 4.17: Ceļš un automašīnas ar projekcijas transformāciju atjaunotajā ceļa plaknē

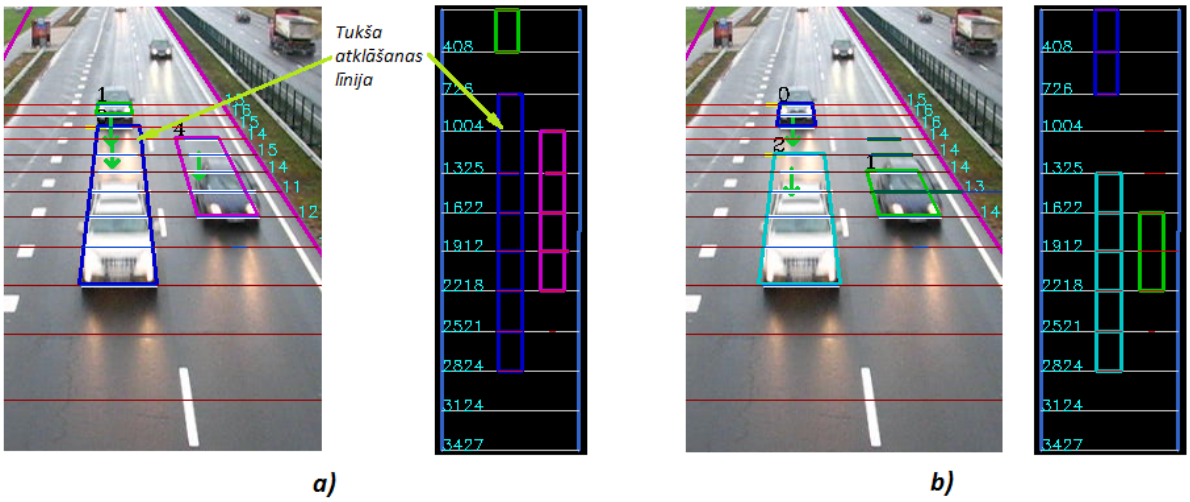
Attēlā 4.17.a atklāšanas līnijas kadrā ir izvietotas ar nevienmērīgu attālumu, jo tiek ņemts vērā, ka perspektīvas dēļ, kadra augšdaļā redzamais ceļš ir vairāk saspiests kadra plaknē, nekā kadra apakšā. Šī īpašība pamanāma arī transformācijas attēlā, kur starp augšējam līnijām ir lielāks attālums nekā starp apakšējām. Programmā arī iestrādāta metode, kas, pirms izvietot kadrā atklāšanas līnijas, izrēķina, vai sākotnēji uzdotais atklāšanas līniju skaits un uzdotais attālums starp tām nenoved pie mēģinājuma izvietot vienu vai vairākas līnijas ārpus kadra. Šādā gadījumā programmas izpildē rastos kļūda, tādēļ programma automātiski samazina atklāšanas līniju skaitu, līdz maksimāli pieļaujamajam pie uzdotajiem attālumiem starp līnijām.

Lieko mašīnu un intervālu dzēšana

Lai palielinātu atklāto mašīnu skaita un noteikto koordinašu precizitāti un samazinātu iespējas tam, ka vairākas tuvu braucošas mašīnas tiktu atklātas kā viena mašīna, tika īstenota metode mašīnai nepiederošu intervālu atmešanai. Ja mašīna atrodas uz vairākām atklāšanas līnijām, ideālā gadījumā uz katras no šīm līnijām jābūt mašīnai atbilstošam intervālam. Tomēr praksē ir gadījumi, ka kāds intervāls netiek izveidots (uz konkrētās līnijas mašīna kļūdaini netika atklāta). Šī iemesla dēļ, mašīnas apgabalā esoša tukša (bez intervāla) līnija nav drošs rādītājs, ka šajā posmā atklātā mašīna būtu jāsadala ar domu, ka patiesībā šeit atrodas divas automašīnas.

Vairāku mašīnu kļūdainas apvienošanas notikums (piemērs parādīts attēlā 4.18.a) ir vairāk ticams, ja tukšā līnija atrodas tuvāk mašīnas aizmugurei, vai arī mašīnā ir vairākas blakus esošas tukšas līnijas. Ar šādu apsvērumu izveidots algoritms, kas sanumurē un augošā secībā sakārto mašīnai piederošo intervālu līnijas pēc to atrašanās vietas vienai pret otru - pirmā līnija ir kadrā pati augstākā līnija, zem tās esošā ir otrā utt. Zinot automašīnas kustības virzienu, sākot no šīs mašīnas paša aizmugurējā intervāla, tiek iegūtas starpības starp blakus intervālu līniju numuriem. Tātad, ja uz divām blakus līnijām ir pa intervālam, starpība būs 1. Ja kāda līnija būs tukša, blakus intervāli atradīsies uz līnijām, kuras nav blakus atklāšanas līnijas. Šādā gadījumā starpība būs lielāka par viens.

Gadījumos, kad starpība ir viens, tiek palielināts nesadalīšanas koeficients. Ja, iegūstot starpības secīgi pa mašīnas intervāliem, šī starpība pārsniedz nesadalīšanas koeficientu, tiek pieņemts lēmums, ka visi mašīnas intervāli, kas pēc atklāšanas līniju numerācijas atrodas pirms šobrīd apskatāmās (ar kuru iegūta pēdējā starpība) tiek izslēgti no mašīnas. Šāds algoritms noved pie tā, ka, piemēram, ja mašīnas aizmugurē ir viens intervāls, kas no pārējās mašīnas atdalīts ar vismaz vienu tukšu atklāšanas līniju, tas tiek atmests. Ja aizmugurē ir divi intervāli, to atmešana norisinās tikai tad, ja vismaz divas tukšas līnijas tos atdala no pārējās mašīnas. Līdzīgi notiek ar 3 un vairāk intervāliem automašīnas aizmugurē. Starpības iegūšana notiek līdz automašīnas vidējam (pēc skaita) intervālam. Tātad intervāli var tikt atmesti tikai tad, ja tie atrodas mašīnas aizmugurējā daļā. Attēlā 4.18.b parādīts gadījums, kad nošķirtais intervāls ticis atrasts un atmests no kadrā viszemāk esošās automašīnas.



Att. 4.18: Lieko intervālu izmešana no automašīnas: a) Zemākajai automašīnai ir pieskaitīts intervāls, kas īstenībā būtu pieskaitāms tuvu aizmugurē braucošai automašīnai, un, gan kadrā gan perspektīvas transformācijas attēlā parādās tukša atklāšanas līnija mašīnas apgabalā; b) Pēc intervālu atmešanas algoritma pielietošanas.

Jau iepriekš kodā tika īstenotas metodes, kas no mašīnas izslēdz pārāk garus vai īsus intervālus (salīdzinot ar intervālu koordinašu mediānas vērtībām pēc projekcijas transformācijas). Tagad kods modificēts, lai intervālu garumu salīdzināšana ar mašīnas vidējiem parametriem notiktu pirms vēl intervāli tiek pievienoti automašīnai, tādējādi nodrošinot, ka šādi ekstrēmi intervāli neietekmē mašīnu vidējos parametrus.

Papildus intervālu dzēšanai, uzlabota arī automašīnas objektu dzēšana. Katram intervālam ir koeficients, kas norāda uz šī intervāla pareizības ticamību, atkarībā no tā, cik daudzi parametri (kustības, fonam nepiederošu robežu, intensitāšu un krāšu esamība intervāla apgabalā) noved pie intervāla uzturēšanas. Īstenota metode, kas summē visu mašīnā esošo intervālu ticamības koeficientus, iegūstot skaitli, kas raksturo visas mašīnas esamības ticamību. Ja mašīna sastāv no intervāliem, kuri ar lielu varbūtību ir noteikti kļūdaini, tad arī visai mašīnai ir lielāka varbūtība būt kļūdaini noteiktai. Tādēļ, šādi automašīnas objekti tiek dzēsti. Praksē šis uzlabojums noved pie tā, ka reālai mašīnai kadrā pametot atklāšanas līniju aptverto ceļa apgabalu, mašīnas objekts tiek slēgts ātrāk nekā tas notiktu bez šī uzlabojuma.

4.4.3 Automašīnu parametru noteikšana

Īstenota arī metode, kas izslēdz no mašīnas neiederīgus intervālus. Ja kāds no mašīnas intervāliem ir būtiski garāks nekā pārējie šīs mašīnas intervāli, tiek pieņemts lēmums šo intervālu izmest no konkrētās mašīnas intervāliem. Ir iespējami gadījumi, kad blakus joslās braucošu automašīnu izveidotie intervāli uz vienas atklāšanas līnijas var saplūst un izveidot kopēju garu intervālu, kas tagad tiek izmests no abu automašīnu intervāliem. Zinot automašīnu koordinātes, īstenota metode, kas garo intervālu sadala divos, kas var tikt atkal pievienoti attiecīgajām mašīnām.

No automašīnā ietilpstošajiem intervāliem tiek iegūts automašīnas platums. Konkrētā kadrā mašīnas platums ir mediānas vērtība starp visiem šobrīd mašīnai piederošajiem intervāliem. Kad mašīna izbraukusi no līniju apgabala, tiek atrasts tās kopējais vidējais platums, kas ir mediānas vērtība starp visos kadros mašīnai aprēķinātajiem platumiem.

Kad eksistējošai, vai iespējamai automašīnai tiek pievienots intervāls, tiek noteikts automašīnas braukšanas virziens - mašīna brauc kadrā uz leju, ja jaunais intervāls atrodas jau atklātas mašīnas apakšā, uz augšu, ja jaunā intervāla līnija ir virs mašīnas. Noteiktie virzieni mašīnām parādīti ar zaļām bultām attēlā 4.15.a. Zinot virzienu, tiek atrasts arī automašīnas ātrums, ko var iegūst izdalot attālumu starp līnijām ar kadru skaitu starp blakus līniju intervālu parādīšanos. Pēc ātruma iespējams noteikt arī mašīnas garumu, ko iegūst mašīnas intervāla pastāvēšanas laiku izdalot ar šī intervāla mašīnas ātrumu.

Ja nosakot ātrumu tika skaitīti kadri tikai starp mašīnai pēdējo un priekšpēdējo pievienoto intervālu parādīšanos, iegūstami neapmierinoši rezultāti. Šādi intervāli visticamāk parādās uz blakus esošām

atklāšanas līnijām, jo mašīna kadrā tās secīgi šķērso. Mazā attāluma dēļ (starp blakus esošām atklāšanas līnijām), izskaitītais kadru skaits ir mazs - vidēji 2-3 kadri. Līdz ar to, pārrēķinot pagājušo kadru skaitu un līniju attālumu uz ātrumu (ātrums = attālums/kadri), tiek iegūts stipri noapaļots rezultāts. Turklāt, atsevišķo līniju mašīnu atklāšanas kļūdu dēļ, novērojami gadījumi, kad uz blakus esošām līnijām intervāli parādās vienlaicīgi, vai pat mašīnas braukšanas virzienam pretējā virzienā. Tas savukārt noved pie kļūdainas ātruma noteikšanas.

Aprakstīto iemeslu dēļ, tiek piedāvāts modificēts ātruma noteikšanas algoritms. Jaunajā variantā pie katra jauna intervāla pievienošanas automašīnai, tiek pierakstīts atbilstošais kadra numurs. Kad mašīna izbrauc no pētāmā apgabala, tiek aprēķināts pagājušo kadru skaits starp visām atklāšanas līnijām. Piemēram, trīs līniju gadījumā tiek iegūts pagājušo kadru skaits starp līnijām 1 un 2, 1 un 3, 2 un 3. Iegūtie ātrumi starp katrām līnijām tiek sakārtoti augošā secībā. Daļa mazāko un lielāko ātrumu tiek atmesta, tādējādi izvairoties no iespējamiem nepareiziem ekstrēmiem rādījumiem. No atlikušajiem lielumiem tiek iegūta vidējā vērtība, kas arī ir noteiktais ātrums. Šīs metodes trūkums gan ir tas, ka tiek atrasts mašīnas vidējais ātrums atklāšanas līniju noklātajā apgabalā.

Papildus ātruma noteikšanas metodes izmaiņām, modificēta arī mašīnas sākotnējā braukšanas virziena noteikšana. Virziens tiek noteikts pēc tā, kurā mašīnas pusē tiek pievienots jauns intervāls. Jaunas mašīnas gadījumā, kad tai pieder tikai divi intervāli, ir svarīgi, lai šie intervāli parādītos pareizā secībā. Kā jau pieminēts pie ātruma algoritma apraksta, testos secināts, ka šis nosacījums ne vienmēr izpildās. Tādēļ tika izmainīts tieši sākotnējais (mašīna tikko atklāta un tai pieder divi intervāli) virziena noteikšanas algoritms.

Jaunajā metodē atkālšanas līnijas tiek sadalītas trīs grupās - augšējas, apakšējās un vidus atklāšanas līnijas. Konkrētas līnijas piederību grupai nosaka kopējais atklāšanas līniju skaits, kā arī tās atrašanās vieta kadrā attiecībā pret pārējām atklāšanas līnijām. Ja līniju skaits ir mazāks vai vienāds ar seši, tad tiek noteikta vidējā atklāšanas līnija, virs kuras (nepāra atklāšanas līniju gadījumā) un zem kuras atrodas vienāds skaits atklāšanas līniju. Tad visas virs vidējās esošās līnijas tiek definētas kā augšējās, kamēr pārējās līnijas ir apakšējās. Šajā gadījumā, kad ir maz atklāšanas līniju, neviena netiek pieskaitīta pie vidus līnijām.

Gadījumos ar vairāk kā sešām līnijām, kopējais līniju skaits tiek dalīts ar 3. Dalījuma un atlikuma summa nosaka cik līnijas no augšas tiek ieskaitītas pie augšējām līnijām - tikpat no apakšas tiek pieskaitītas pie apakšējām. Pārējās līnijas ir vidus līnijas.

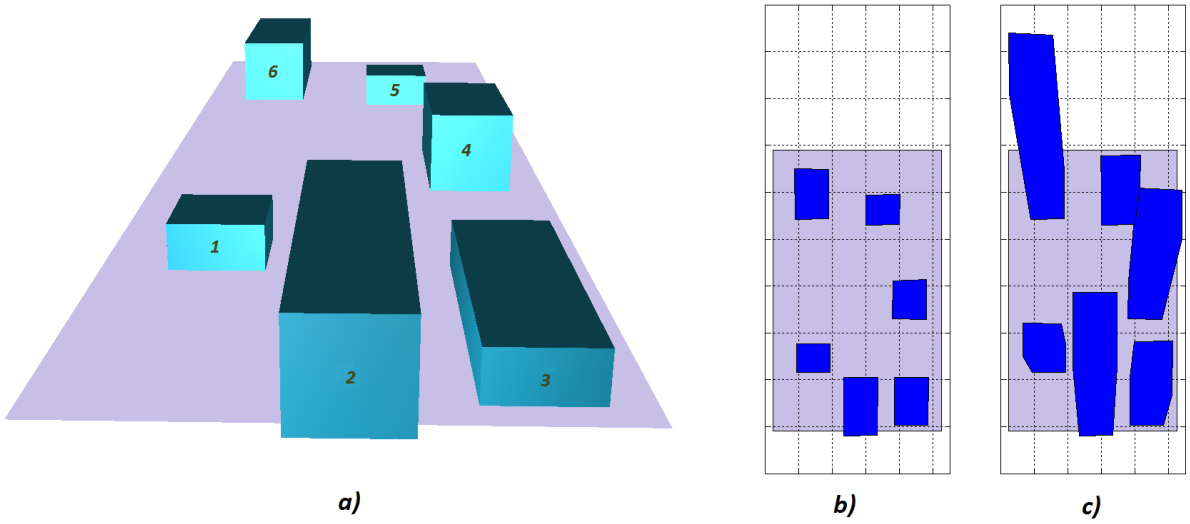
Kad kadrā tiek atklāta jauna automašīna, tās sākotnējais virziens tiek noteikts atkarībā no tā, uz kādām līnijām atrodas mašīnas intervāli. Ja mašīna vispirms atklāta uz augšējām līnijām, tiek pieņemts, ka virziens ir no kadra augšas uz leju, savukārt, ja mašīna vispirms parādās uz apakšējām līnijām, virziens tiek pieņemts uz augšu. Šāds algoritms pamatojas uz to, ka mašīna var sasniegt augšējās līnijas pirms apakšējām tikai tajā gadījumā, ja tā kadrā pārvietojas no augšas uz leju. Ja mašīna parādās uz vidējām līnijām, tad vēl netiek noteikts mašīnas virziens.

Turpmāka virziena noteikšana vai koriģēšana konkrētai mašīnai notiek līdzīgi kā iepriekšējā variantā, vadoties pēc jauno intervālu pievienošanas virziena. Tomēr, lai izvairīto no kļūdām, ko var izraisīt viens nepareizā mašīnas pusē pievienots intervāls, tika ieviesta noteiktā mašīnas virziena inerce. Ja šī brīža kadrā mašīnai tiek pievienots jauns intervāls, un, no tā noteiktais virziens sakrīt ar iepriekš noteikto globālo mašīnas virzienu (jaunatklātai mašīnai globālais virziens ir vienāds ar sākotnējo noteikto), par vienu tiek palielināta pārliecība (virziena nemainīšanas koeficients), ka globālais virziens ir pareizs. Ja jaunais intervāls norāda uz pretēju kustības virzienu, par vienu tiek palielināta pārliecība, ka jāmaina globālais virziens. Ja maiņas koeficients pēc vērtības pārsniedz nemainīšanas koeficientu, tiek mainīts automašīnas globālais kustības virziens.

Mašīnu augstuma noteikšana

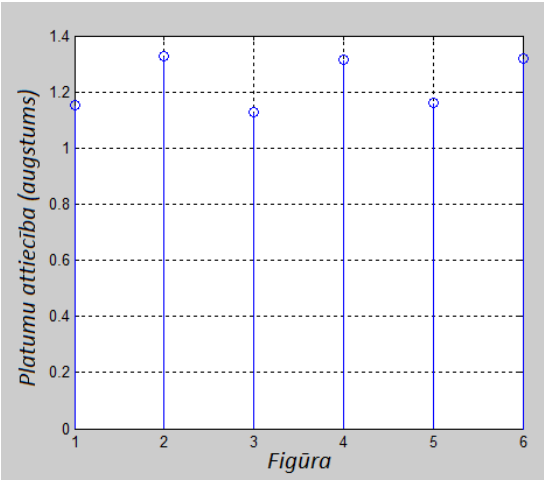
Izmantojot to, ka kamera vērsta slīpi pret ceļu, un, ka mašīnas nav redzamas precīzi no augšas, iespējams iegūt vēl vienu mašīnas parametru - augstumu. Attēlā 4.19.a parādīts simulēts kadrs ar ceļa plakni un mašīnas reprezentējošiem heksaedriem. Ja kadra projekcijas transformācija dotu ideālu skatu no augšas, tiktu iegūta attēlā 4.19.b parādītā projekcija. Tomēr, tā kā kadrā redzamas arī heksaedru sānu skaldnes, reālā iegūtā projekcija ir 4.19.c.

Ja pieņem mašīnu kustības virzienu no kadra augšas uz leju, tad uz atklāšanas līnijas vispirms nonāk automašīnas priekšējā apakšējā daļa. Pēc laika uz līnijas būs nonācis automašīnas jumts. Jumta daļa būs tuvāk kamerai nekā mašīnas pamats. Ja pieņem, ka automašīnas platums būtiski nemainās no tās pamata līdz jumtam, tad intervāla platuma izmaiņa projekcijas transformācijā starp brīdi, kad uz līnijas bija mašīnas apakšējā daļa un brīdi, kad uz līnijas nonāca jumts, skaidrojama ar jumta tuvāku



Att. 4.19: Sintezēts ceļa kadrs a) Heksaedra figūras uz ceļa plaknes; b) Mākslīgā gadījuma augšskats; c) Kadra projekcijas transformācija

atrašanos pret kameru. Jo automašīna augstāka, jo lielākai jābūt intervāla platuma izmaiņai. Attēlā 4.19.a ir seši heksaedri 1., 3. un 5. ir 1m augsti, savukārt 2., 4., 6. - 2m augsti. Figūru garumi ir (pēc augošas numerācijas): 2m, 8m, 6m, 4m, 1m, 6m. No iepriekš izteiktā pieņēmuma, projekcijas transformācijā (attēls 4.19.c) katrai figūrai apakšējās daļas platumam attiecībā pret augšējās daļas platumu būtu jānorāda uz mašīnas augstumu. Attēlā 4.20 parādīta katras figūras augšējās un apakšējās daļas platumu attiecības. Neraugoties uz garuma atšķirībām, novērojams, ka mašīnas ar vienādiem augstumiem raksturojas ar līdzīgām intervālu platuma attiecībām pēc projekcijas transformācijas.



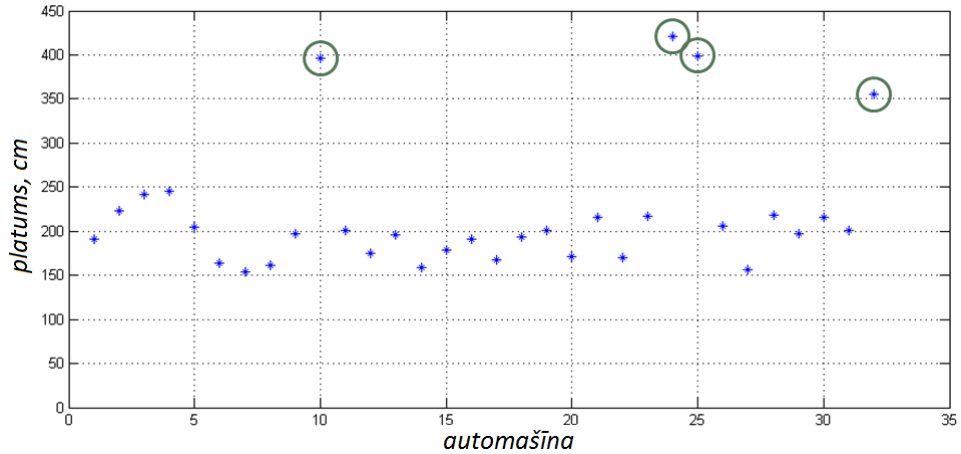
Att. 4.20: Attiecība starp figūras projekcijas transformācijas apakšējās daļas un augšējās daļas platumu

Programmas kodā augstuma noteikšana īstenota tādējādi, ka katram izveidotajam intervālam pēc projekcijas transformācijas tiek pierakstītas tā mazākā un lielākā platuma vērtības. Katram intervālam tiek iegūta šo platumu attiecība. Atklātai mašīnai pametot pētāmo kadra apgabalu, tiek vidējotas visu mašīnā ietilpušo intervālu platumu attiecības. Šī vidējā attiecība tiek pieņemta par mašīnas augstumu raksturojošu skaitli. Lai pārveidotu attiecību uz augstuma metriem, jāveic kalibrāciju pie katra kameras novietojuma. Ja video ir zināmi divu dažādu mašīnu augstumi, tad, izmantojot tiem noteiktās intervālu platuma attiecības, var iegūt lineāru sakarību starp attiecību un reālo augstumu.

4.4.4 Automašīnu klasificēšana

Kad ir iegūti dažādi automašīnas parametri, tos var izmantot, lai klasificētu automašīnas vieglajās un smagajās. Izveidotajā klasifikatorā, katrs mašīnas parametrs tiek salīdzināts ar sliekšņa vērtību. Piemēram, ja noteiktais automašīnas augstums ir virs 2.5m, tad šis parametrs norāda, ka ir atklāta smagā automašīna. Līdzīgi sliekšņi tiek pielietoti arī noteiktajiem garumam un platumam. Lai pieņemtu gala lēmumu par mašīnas klasificēšanu, tiek ņemti vērā visi trīs parametri. Automašīna tiek klasificēta kā smagā, ja vismaz divi no parametriem uz to norāda. Tāpat, ja vismaz divi parametri norāda, ka tikusi atklāta vieglā automašīna, tad tāds ir arī gala lēmums par konkrēto transportlīdzekli.

Lai pārliecinātos par atsevišķo noteikto parametru (platums, garums, augstums) ticamību, veikti testi, kuros analizēta dažādo noteikto parametru aptuvenā atbilstība īstenībai. Katras mašīnas reālie izmēri kadrā nav zināmi, tomēr, tā kā klasificēšana notiek tikai divās klasēs - smagā vai vieglā mašīna - tad noteikšanas precizitātes analīzē uzsvars tiek likts uz to, vai noteiktais parametrs ir ticams attiecībā uz kadrā redzamo mašīnas klasi. Piemēram, vai vieglās mašīnas garums ir lielāks par diviem metriem un nepārsniedz piecus metrus.

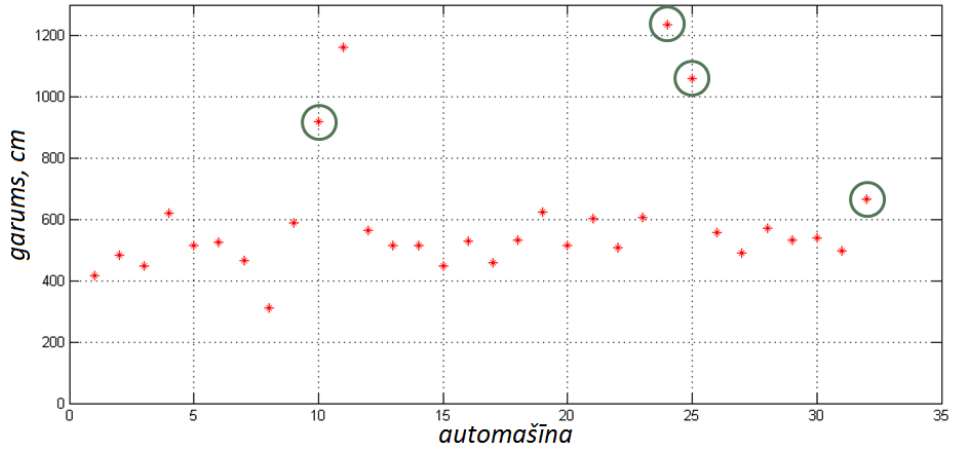


Att. 4.21: Noteiktie automašīnu platumi

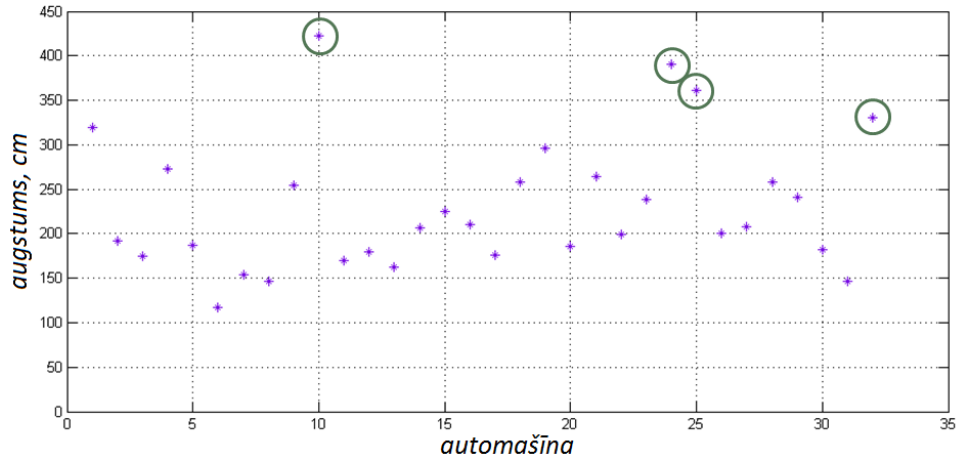
Attēlā 4.21 parādīts platuma noteikšanas rezultāts iepriekš ierakstītam video. Grafikā X ass atbilst mašīnas numuram pēc kārtas. Konkrētajā video atklāšanas apgabalu kadrā ir šķērsojušas 32 automašīnas. Ar apli ir apvilktas smagās automašīnas. Uz Y ass ir parādīts noteiktais automašīnu platums centimetros. Piemērā novērojams, ka noteiktie vieglo automašīnu platumi ir savstarpēji līdzīgi un ticami. Četras smagās automašīnas būtiski atšķiras no vieglajām pēc noteiktajiem platumiem. Šajā piemērā klasificēšanu varētu veikt jau tikai pēc platuma parametra. Tomēr, jāņem vērā, ka ja kamera būs novietota slīpāk pret ceļu, platums var tikt noteikts neprecīzāk. Noteikto platumu var ietekmēt arī uz mašīnu sāniem krītošas ēnas attiecīgos laikapstākļos.

Attēlā 4.22 tam pašam video, kas izmantots attēlā 4.21, noteikti mašīnu garumi. Lai gan lielākā daļa noteikto garumu ļauj izdalīt vieglās un smagās automašīnas, tomēr divos gadījumos novērojamas būtiskas kļūdas. Vienpadsmitajai automašīnai, kas bija vieglā automašīna, garums noteikts pārāk liels, savukārt 32. mašīnas garums noteikts pārāk mazs, lai to droši varētu klasificēt kā smago automašīnu. Šādas kļūdas parāda, kādēļ automašīnu klasificēšanā netiek izmantos tikai viens izmēra parametrs, bet gan kombinācija no trīs. Ja divi citi parametri norādīs uz smagās automašīnas esamību tur, kur garuma parametrs norāda uz vieglo mašīnu, tiks pieņemts lēmums par smagās automašīnas atklāšanu. Mašīnu garuma noteikšanas precizitāte ir atkarīga no ātruma noteikšanas precizitātes. Ātruma noteikšana atkarīga no vairākiem faktoriem, to starpā no video kadru skaita sekundē. Iespējams, palielinot kameras kadru skaitu sekundē, varētu mazināt garuma noteikšanas kļūdas.

Tam pašam piemēram attēlā 4.23 parādīti noteiktie augstumi. Piemērā augstums ļauj pareizi atdalīt smagās un vieglās automašīnas uzstādot augstuma slieksni, tomēr iegūtajiem vieglo automašīnu augstumiem piemīt samērā liela izkliede, kas neļauj augstumam būt pietiekoši uzticamam parametram, lai to lietotu kā vienīgo parametru mašīnu klasifikācijā. Daļēji augstuma noteikšanas precizitāte atkarīga no kameras novietojuma kalibrācijas, kas veikta augstuma noteikšanai.



Att. 4.22: Noteiktie automašīnu garumi



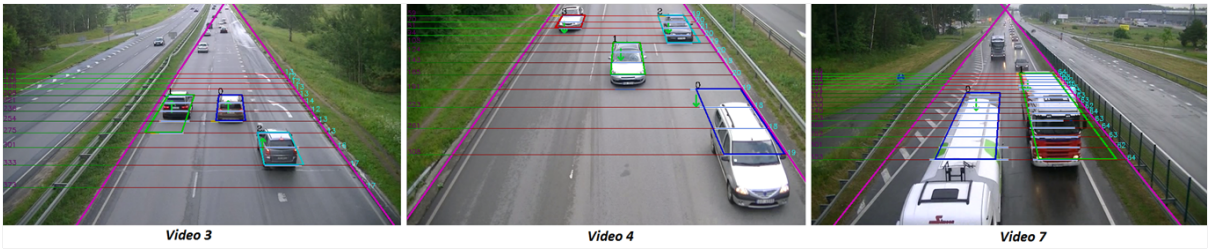
Att. 4.23: Noteiktie automašīnu augstumi

4.4.5 Testi ar vairāku atklāšanas līniju algoritmu

Tika izveidots mašīnu klasificēšanas algoritms, kas, apvienojot rādījumus no vairākām kadrā izvietotām atklāšanas līnijām, nosaka automašīnas izmēru parametrus, kustības virzienu, ātrumu un nosaka, vai atklātā mašīna ir vieglā vai smagā. Iespējamā algoritma papildus priekšrocība varētu būt lielāka mašīnu atklāšanas precizitāte, salīdzinot ar vienas atklāšanas līnijas izmantošanu. Tā kā līnijas ir izvietotas dažādos ceļa apgabalos, kādas konkrētas līnijas kļūdaina mašīnas neatrašana vai viltus mašīnas pieskaitīšana var tikt labota, ja vairākas citas atklāšanas līnijas tomēr noteikušas mašīnas esamību pareizi. Tāpat, ņemot vērā, ka piedāvātais algoritms, kas apvieno intervālus no dažādām atklāšanas līnijām, atklājot automašīnas spēj ignorēt kļūdaini izveidotus intervālus, paredzams, ka tas būtu precīzāks automašīnu skaitīšanā nekā vienkārša dažādo atklāšanas līniju rādījumu vidējošana. Šis pieņēmums tika pārbaudīts testos, kas veikti uz dažādiem jauniem testa video (attēls 4.24).

Šajos video ceļam ir divas līdz trīs joslas. Vidējais katra video ilgums ir 7 minūtes. Video 3 mašīnas kustas prom no kameras, pārējos tās brauc kameras virzienā, 6-8 video līst lietus. Attēlā 4.25 parādīti testu rezultāti katram video. Tabulā dots reālais mašīnu skaits video. Tad parādīts, kādu skaitu atgriež viena pati atklāšanas līnija. No visām atklāšanas līnijām kadrā, šeit izvēlēta kadrā vidējā līnija. Tālāk parādīts mašīnu skaits, kas iegūts vienkārši vidējojot visu atsevišķo līniju rādījumus. Secināms, ka vairāku līniju rādījuma vidējošana dod vai nu mazu precizitātes uzlabojumu, vai pat ir mazāk precīza kā vienas līnijas gadījumā. Visbeidzot parādīts skaits, kas iegūts ar jauno līniju rādījuma apvienošanas algoritmu, kas arī ir precīzākais paņēmiens no dotajiem. Tas apstiprina pieņēmumu, ka piedāvātā intervālu apvienošana automašīnu objektos un to izsekošana palielina mašīnu atklāšanas precizitāti.

Lai noteiktu automašīnu klasificēšanas precizitāti, tika izmantoti tie paši video. Lai novērtētu



Att. 4.24: Kadri no dažiem jaunajiem testa video

Video	Reālais skaits	Viena līnija		Vairāku līniju vidējais rādījums		Piedāvātā metode	
		skaits	precizitāte	skaits	precizitāte	skaits	precizitāte
1	32	30	94%	30	94%	32	100%
2	122	121	99%	124	98%	122	100%
3	159	168	94%	164	97%	158	99%
4	43	45	95%	44	98%	44	98%
5	221	232	95%	235	94%	231	95%
6	208	209	99%	209	99%	209	99%
7	246	233	95%	239	97%	239	97%
8	117	118	99%	118	99%	116	99%
9	88	92	95%	93	94%	88	100%

Att. 4.25: Testu rezultāti atklājot automašīnas ar vienu līniju, vidējojot rezultātus pa visām kadrā izvietotām līnijām, kā arī izmantojot piedāvāto līniju rādījumu apvienošanas algoritmu

klasificēšanas precizitāti ir jāzina brīži, kuros atklāšanas līniju aptverto apgabalu šķērso smagās un kuros tos šķērso vieglās automašīnas. Tādēļ katram izmantotajam video tika manuāli pierakstīti kadri, kuros pēdējo atklāšanas līniju šķērsoja attiecīgās klases automašīna. Mašīnu klasificēšanas algoritms, līdz ar atklātās automašīnas klasi pieraksta arī kadru, kurā automašīna atklāta. Abu kadru un klašu sarakstu salīdzināšanai tika izveidota programma MATLAB vidē. Programmā katrai manuāli pierakstītajai automašīnai tiek piemeklēta attiecīgā algoritma atklātā automašīna, ja reālais kadra numurs un atklātais ir pietiekoši tuvi. Piemēram, ja video failam ar 30 kadriem sekundē pierakstītais un algoritma atrastais mašīnai atbilstošais kadra numurs atšķiras par 30, tad algoritms ir atklājis konkrēto automašīnu par sekundi vēlāk, vai ātrāk nekā to pierakstījis cilvēks - šāda laika atšķirība mūsu gadījumā tiek pieļauta. Ja laika starpība pārsniedz 3 sekundes, tiek pieņemts lēmums, ka atklātā un pierakstītā automašīna ir dažādas. Ja, piemēram, atklātai smagajai mašīnai neatbilst neviena reāla smagā mašīna konkrētā laika apgabalā, tiek atrasta viltus atklāšanas kļūda. Savukārt, ja manuāli pierakstītai smagajai mašīnai neatbilst neviena no atklātajām, tiek atrasta mašīnas neatklāšanas kļūda. MATLAB programma pēc video faila manuālas aprakstīšanas ļauj automātiski iegūt klasificēšanas precizitāti, darbinot algoritmu ar šo video failu pie dažādām algoritma konstantēm. Maināms ir arī laika logs, kurā tiek meklētas atbilstošās mašīnas reālajā un algoritma atgrieztajā sarakstā, jo dažādiem video failiem var būt dažāds kadru skaits sekundē.

Iegūtie testu rezultāti apkopoti tabulā 4.26. Tabulā parādīts, cik automašīnu šķērsoja pētāmo apgabalu katrā video, kā arī tas, cik no šīm automašīnām bija smagās. Attēlots, cik smagās automašīnas algoritms atklāja, kā arī parādīts, cik daudz bijis viltus atklāšanas gadījumu. Šajos video pareizi tika noklasificētas 88% smago automašīnu, kā arī bija samērā mazs viltus atklāšanas gadījumu skaits.

4.5 Avārijas situāciju un pārkāpumu atklāšana

Algoritms spēj noteikt avārijas situācijas uz ceļa pēc divām pazīmēm – netipisku apstājušos automašīnu esamība uz ceļa, kā arī netipiska mašīnu braukšanas trajektoriju izvēle.

Video	Reālais skaits		Pareizi atklātās automašīnas	Viltus smago mašīnu atklāšana
	Visas automašīnas	Smagās automašīnas		
1	32	4	4	0
2	122	8	8	0
3	159	8	7	1
4	43	0	0	0
5	221	12	7	0
6	208	5	5	1
7	246	12	12	3
8	117	4	4	1
9	88	7	6	0

Att. 4.26: Klasificēšanas algoritma testu rezultāti

4.5.1 Avāriju atklāšana pēc mašīnu apstāšanās

Automašīnu apstāšanās tiek atklāta, ja vismaz kādu iepriekš noteiktu laiku(piemēram, pēdējos 20 kadrus) nav mainījušies mašīnā ietilpstošie intervāli. Ja šajā laikā nav nācis klāt un nav dzēst neviens automašīnas intervāls, šis automašīnas objekts tiek uzskatīts par apstājušos, līdz brīdim, kad tam tiek pievienots jauns vai dzēst kāds no esošajiem intervāliem. Ja uz pētāmā ceļa mašīnām nav paredzēts apstāties, tad apstājušos automašīnu atklāšana var norādīt uz netipisku satiksmes notikumi. Automašīnu izsekošanas algoritms var nosegt samērā plašu ceļa apgabalu, atkarībā no atklāšanas līniju izvietojuma. Uzstādot kameru, iespējams manuāli iestādīt apgabalus uz ceļa, kuros ir iepriekš definēts normāls automašīnu kustības virziens, kā arī tas, vai šajā apgabalā mašīnu apstāšanās ir normāla. Apgabaliem programmas kodā tiek izveidota jauna apgabalu klase „DirectionRegion".

Kustības virzienu apgabalu piemērs parādīts attēlā 4.27.a. Piemēra kadrā ir manuāli iestatīti divi ceļa apgabali ar pretējiem braukšanas virzieniem. Pirmajā apgabalā ir atklātas divas automašīnas, kas kustas šim apgabalam normālā virzienā (norāda bultas pie automašīnu objektiem). Otrajā apgabalā atklātas trīs automašīnas, no kurām vienai vēl nav noteikts virziens. Automašīnas objektu apraksta četrstūra stūri. Katrs braukšanas virziena apgabals arī tiek aprakstīts ar četrstūri. Lai noteiktu, kurā apgabalā atrodas mašīna, tās stūru koordinātes tiek salīdzinātas ar visu apgabalu koordinātēm. Ja vismaz trīs mašīnas stūri atrodas apgabala četrstūrī, tiek pieņemts, ka mašīna atrodas attiecīgajā apgabalā. Ja mašīnas virziens nesakrīt ar apgabalam iepriekš uzstādīto normālo virzienu, mašīnas objektam tiek pierakstīta īpašība: brauc nepareizā virzienā.

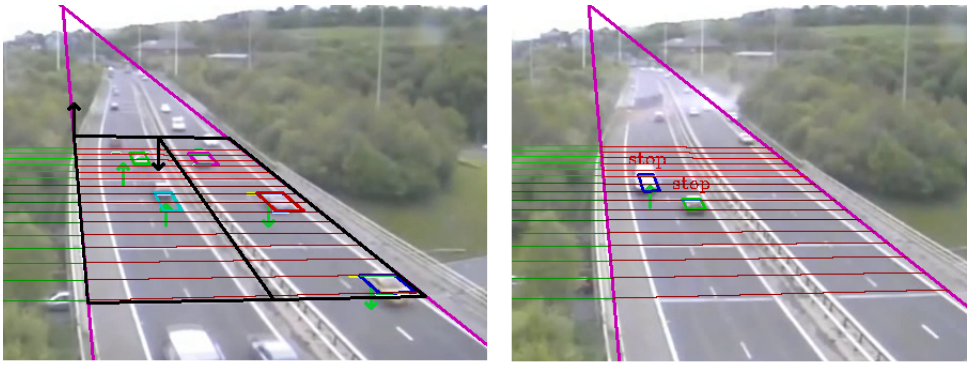
Ja iestādot ceļa apgabalu norāda, ka tajā mašīnām nav paredzēts apstāties, līdzīgi kā ar braukšanas virzienu noteikšanu, iespējam atrast, kuras mašīnas ir netipiski apstājušās. Piemērā 4.27.b algoritms ir noteicis, ka pētāmajā apgabalā atrodas divas apstājušās automašīnas. Iemesls šajā gadījumā ir avārija, kas notikusi ārpus apgabala, un, kuru algoritms netieši atklājis caur apstājušos mašīnu noteikšanu. Kad atklāta šāda netipiska situācija, algoritms var dot signālu, lai neilgs (dažas pēdējās minūtes) video posms tiktu saglabāts MITS punkta atmiņā. Šim nolūkam jālieto atmiņas buferis, kurā glabājas nesenais video materiāls. Saglabātais materiāls var noderēt vēlākai avārijas iemeslu noskaidrošanai.

4.5.2 Avāriju atklāšana pēc mašīnu trajektoriju analīzes

Papildus parametrs, kas norāda uz iespējamas avārijas esamību uz ceļa, ir mašīnu braukšanas trajektoriju izvēles izmaiņas attiecībā pret tipiskām trajektorijām. Mašīnu trajektorijas tiek iegūtas izsekojot automašīnu kustībai caur atklāšanas līnijām. Trajektoriju pierakstam un apstrādei kodā tiek izveidota jauna trajektorijas objektu klase „Trajectory".

Trajektorija sastāv no vairākiem punktiem – pa vienam punktam uz katras atklāšanas līnijas, kuru šķērsojusi automašīna. Šis punkts uz katras līnijas ir šīs līnijas mašīnas intervāla viduspunkts. Tā kā intervāla platums un viduspunkts laikā var mainīties, šis viduspunkts tiek atrasts katrā intervāla pastāvēšanas kadrā. Kad intervāls tiek slēgts, no tā visiem saglabātajiem viduspunktiem tiek atrasta apcirpta vidējā vērtība, kas arī tiek pieņemta par mašīnas trajektorijas punktu uz konkrētas atklāšanas līnijas.

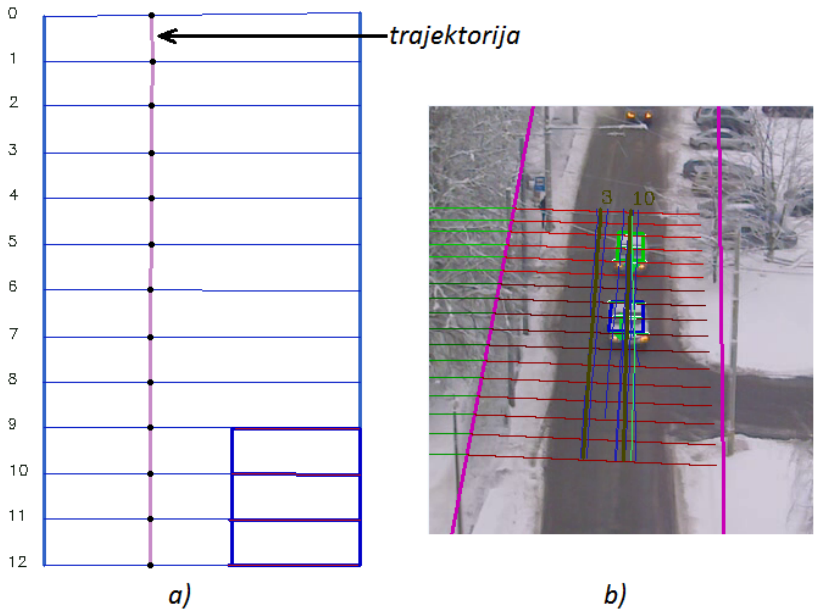
Kad mašīna pamet atklāšanas apgabalu, tās trajektorijas punkti ir iegūti uz katras no šķērsotajām



Att. 4.27: Nepareizā virzienā braucošu un avarējušu mašīnu noteikšana

atklāšanas līnijām. Vizualizācijas nolūkiem šie punkti var tikt savienoti (attēls 4.28.a) iegūstot nepārtrauktu mašīnas trajektorijas līniju atklāšanas apgabalā.

Programma saglabā kādu skaitu no pēdējo mašīnu trajektorijām. Trajektoriju klasē īstenota metode, kas ļauj savā starpā salīdzināt saglabātās trajektorijas. Salīdzināšanā tiek skatīts cik tāli savā starpā ir dažādo trajektoriju punkti, kas atrodas uz vienas un tās pašas atklāšanas līnijas. Ja lielākā daļa salīdzināmo trajektorijas punktu ir savā starpā tuvi, tiek secināts, ka dažādās mašīnas braukušas pa vienu un to pašu trajektoriju, ar nelielām atšķirībām. Uzkrājot un salīdzinot vairākas pēdējās mašīnu trajektorijas, algoritms atrod izteiktas normālas trajektorijas pētāmajā ceļa apgabalā. No savstarpēji tuvām trajektorijām tiek aprēķināti normālu trajektoriju punkti, vidējojot normālo trajektoriju veidojošo atsevišķo trajektoriju punktus uz katras atklāšanas līnijas.

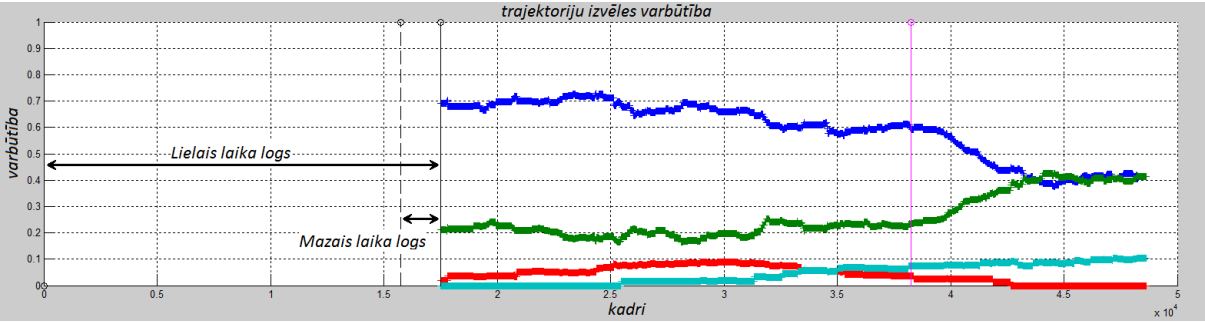


Att. 4.28: Noteiktās mašīnu trajektorijas a) viena trajektorija pasaules koordinašu plaknē; b) 10 pēdējās un divas vidējās noteiktās trajektorijas

Attēlā 4.28.b parādītas pēdējās 10 trajektorijas, kā arī divas atklātās normālās trajektorijas. Kad ir izveidota normālā trajektorija, tā tiek salīdzināta ar jaunām tikko iegūtām mašīnu trajektorijām. Ja jaunā trajektorija ir tuva normālajai, tad normālās trajektorijas punkti tiek pārrēķināti, ņemot vērā pēdējo pievienoto, kā arī visas iepriekšējās trajektorijas, kas atbilda par šīs normālās trajektorijas izveidi. Tiek skaitīts un pierakstīts, kuros kadros normālajai trajektorijai atbilst jauna pagaidām pēdējā trajektorija, jeb pa kuru normālu trajektoriju ir izbraukusi pēdējā mašīna. Attēlā 4.28.b virs trajektorijām

ir skaitļi 3 un 10, kas atbilst mašīnu skaitam, kas pa attiecīgajām trajektorijām ir izbraukušas.

Skaitot pa katru no normālajām trajektorijām izbraukušu automašīnu kādā laika logā (piemēram 5 minūtes), un dalot šo skaitu ar kopējo laika logā atklāto automašīnu skaitu, iegūst varbūtību, ar kādu mašīnas šajā laikā ir izvēlējušās braukt pa katru no tipiskajās trajektorijām. Ja mazākā laika logā (piemēram minūti) šī varbūtība kādai no trajektorijām stipri atšķiras, mašīnu trajektoriju izvēle ir izmainījusies, kas, savukārt, var norādīt uz avārijas situācijas esamību uz ceļa. Attēlā 4.29 parādīta trajektoriju izvēles izmaiņa laikā 4.28.b parādītajā video. Trajektoriju izvēles varbūtība tiek rēķināta vairāku minūšu garā laika logā. Attēlā novērojamas četras normālas trajektorijas, no kurām divas ir izteiktas, un, divas ir reti gadījumi, kad mašīnas braukušas kā savādāk nekā kadrā esošās divas ceļa joslas.No grafika secināms, ka attiecībā uz divām būtiskajām trajektorijām, ilgākā laika posmā ir vienmērīgi mainījusies mašīnu braukšanas varbūtība pa katru no tām.



Att. 4.29: Mašīnu trajektoriju izvēles varbūtība, rēķinot vairāku minūšu garā laika logā

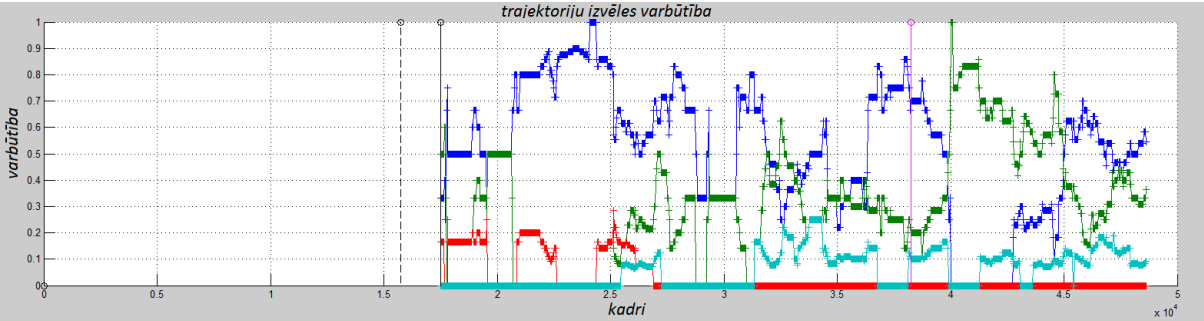
Ir zināms, ka konkrētajā video, ap kadru 38250 noticis simulēts ceļa nagadījums – divas automašīnas apstājušās ceļa malā un stāv tur dažas minūtes (attēls 4.30), liekot pērējām automašīnām biežāk izvēlēties ierpeikš mazāk populāru trajektoriju, lai apbrauktu cietušos.

Att. 4.30: Simulētas avārijas gadījums

Līdzenā varbūtības izmaiņa grafikā 4.29 iegūta pētot varbūtības pietiekami lielā laika logā. Lai iegūtu informāciju par strujāku mašīnu izvēles varbūtības izmaiņu, tiek pētīts īsāks laika logs. Varbūtības izmaiņa īsākā logā parādīta attēlā 4.31.

Algoritmā uz avārijas gadījumu norādīs mazā laika loga un lielā laika loga varbūtību salīdzinājums. Ja notikumi mazajā laikā logā ir līdzīgi ar lielo, tad mazā loga datus var uzskatīt par lielā laika loga datu apakškopu. Šādā gadījumā mašīnu uzvedība uz ceļa nav būtiski mainījusies, un, algoritms neceļ traukcsmi par ārkārtas gadījumu. Ja varbūtības būtiski sāk atšķirties, konstatējama avārija.

Laika logu varbūtības algoritmā tiek salīdzinātas. Lielais laika logs $V_{lielais}$ ir matrica, kas sastāv no 0 un 1. Laika momentos, kad mašīna ir izvēlējusies konkrēto trajektoriju, matricā tiek ierakstīts vienīnieks.



Att. 4.31: Mašīnu trajektoriju izvēles varbūtība, rēķinot īsākā laika logā

Kad atklāta mašīna, kas ir izvēlējusies kādu citu braukšanas trajektoriju, matricā tiek ierakstīta nulle. Līdzīgi tiek izveidota mazā laika loga matrica V_{mazais}. Abas matricas tiek salīdzinātas ar statistiskās hipotēzes testu, atrodot kopu līdzības parametru z.

$$z = \frac{abs(mean(V_{lielais}) - mean(V_{mazais})) \cdot \sqrt{n}}{\sigma}, \tag{4.9}$$

kur $mean(V_{lielais})$ - lielā logā esošo vērtību vidējā vērtība; σ - lielā loga vērtību standarta novirze; n - konkrētās trajektorijas izvelēšanās skaits mazajā laika logā.

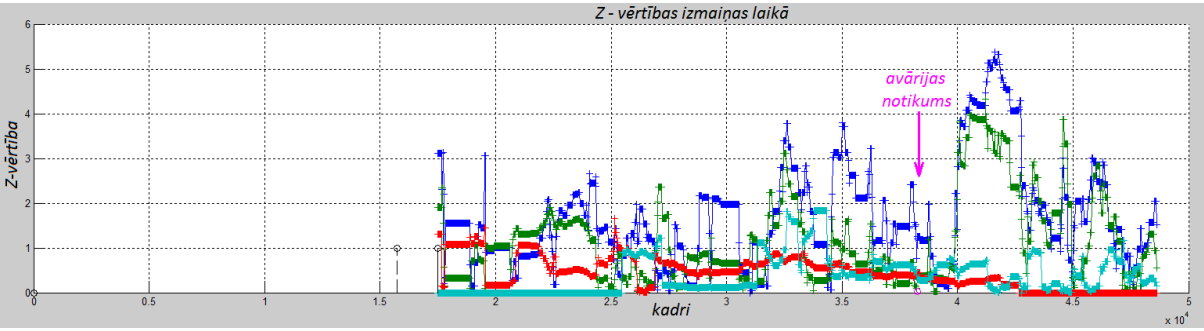
Par nulles hipotēzi tiek pieņemts, ka mazais laika logs ir lielā laika loga apakškopa. Jo lielāka ir iegūtā z vērtība, jo mazāka iespējamība, ka nulles hipotēze izpildās. Tātad pie lieliem z ir notikušas būtiskas izmaiņas uz ceļa. Pirms algoritma darbināšanas, iestādot slieksni, algoritms var dot trauksmes signālu un saglabāt pēdējos notikumus video, kad z vērtība iestādīto slieksni pārsniedz.

Z vērtībā uz ļoti reti izvēlētām trajektorijām var parādīties viltus izsitieni, jo laika logi statistiski ļoti atšķiras, ja pēdējās 10 minūtes (lielais laika logs) pa kādu retu trajektoriju ir braukusi tikai viena automašīna, kas noticis pēdājās minūtes laikā (mazais laika logs). Tādēļ Z vērtība tālāk tiek modificēta:

$$z_{fix} = \frac{z \cdot sum(V_{lielais})^{\frac{1}{3}}}{2}, \tag{4.10}$$

kur $sum(V_{lielais})$ - matricas $V_{lielais}$ elementu summa, kas atbilst mašīnu skaitam lielajā loga, kas šo trajektoriju izvēlējušās.

Pēc formulas modificēšanas video no attēla 4.30 tiek iegūtas sekojošas z vērtības līknes (attēls 4.32). Šajā grafikā lielākais izsitiens ir palicis laikā, kad video tiešām tika novērota avārijas situācija. Pielietojot slieksni, piemēram, kad z vērtība lielāka par 4, algoritms var dot signālu par atklātu negadījumu.

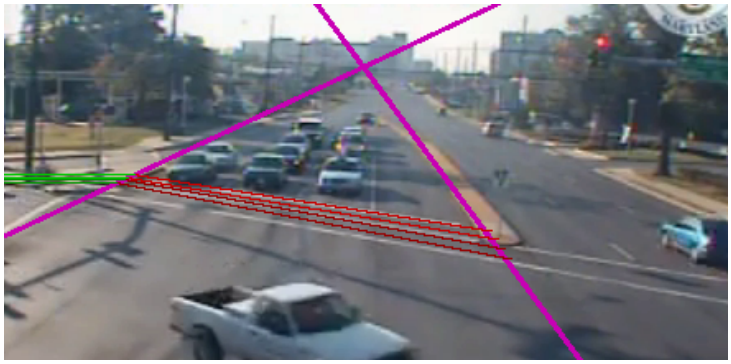


Att. 4.32: Z-vērtība pēc formulas modifikācijas

4.5.3 Luksofora signāla neievērošanas atklāšana

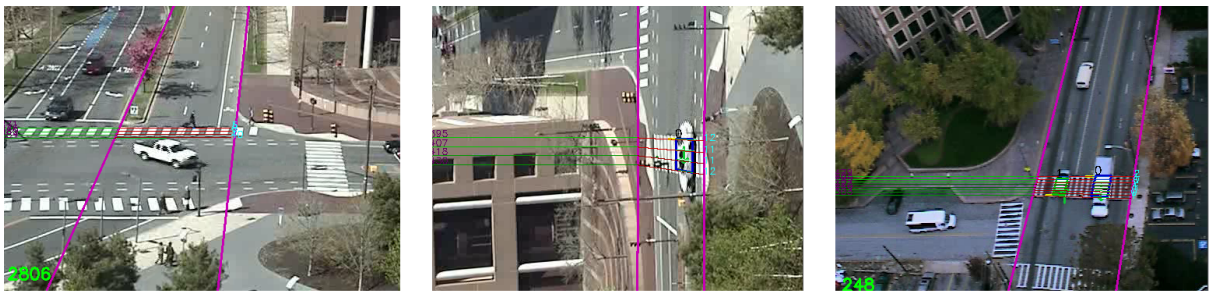
Līnijas intervālu metode ir pielāgojama arī, lai atklātu gadījumus, kad mašīna neievēro sarkano luksofora signālu. Virtuālās līnijas izvietotas pie stop līnijas (attēls 4.33), līdz ar to tiek noteiktas mašīnas, kas

šo stop līniju šķērso. Ja šķērsošanas brīdī luksoforā deg sarkanā gaisma, tiek atklāts pārkāpums. Tā kā vairāku līniju lietošana ļauj noteikt arī mašīnas braukšanas virezienu, tad algoritmā iespējams uzstādīt, lai tas kā pārkāpējus nodetektē tikai konkrētā virzienā braucošas mašīnas.



Att. 4.33: Atklāšanas līniju izvietošana pie krustojuma stop-līnijas

Atsevišķi eksperimenti veikti, izvietojot atklāšanas līnijas uz gājēju pārejām. Šāds izvietojums ir interesants, jo ļauj pārbaudīt gājēju ietekmi uz skaitīšanas precizitāti. Ja līnijas izvietotas perpendikulāri ceļam uz gājēju pārejām (attēls 4.34), tās bieži šķērso cilvēki, kas iet pāri šīm pārejām. Mērķis ir šos cilvēkus nepieskaitīt pie pāreju šķērsojošām automašīnām. Ja pāreja ir regulējama ar luksoforu, tad, atklājot mašīnas, iespējams noteikt noteikumus pārkāpjošus vadītājus, kas šķērso pāreju pie sarkanās gaismas. Gājēju pārejas ir arī interesants testa fons atklāšanas līnijām, jo atšķirībā no iepriekš testos izmantotajiem ceļiem, fons ir nevis vienmērīgs ceļa krāsas apgabals, bet gan apgabals ar daudziem izteiktiem kontūriem starp mainīgo balto un asfalta krāsas segumu.



Att. 4.34: Atklāšanas līnijas uz gājēju pārejām

Eksperimenti veikti ar attēlā 4.34 parādītajiem video. Vairumā gadījumu pāreju šķērsojošie cilvēki, kas kustējās paralēli atklāšanas līnijām, neietekmēja atklāto automašīnu skaitu. Tomēr kļūdas radās atklāšanas līniju izvietojuma dēļ attiecībā pret visu krustojumu. Gadījumā 4.34.a pētāmā gājēju pāreja atrodas tuvu šķērsojošai ielai. Ja kadrā redzamās baltās automašīnas vietā (mašīna nav šķērsojusi gājēju pāreju, tā brauc pa pārejai paralēlu ielu) būtu busiņš vai smagā automašīna, kameras slīpā novietojuma pret ceļu dēļ, šī mašīna nosegtu lielu daļu atklāšanas līniju. Tādējādi tiek pieskaitītas augstas automašīnas, kas brauc pa citu ceļu. Šajā pašā kadrā redzama cita pāreja, kas atrodas tālāk no krustojuma - attēlā 4.34.b tiek skaitītas mašīnas uz šīs pārejas, pirms tam rotējot kadru par 90 grādiem. Šajā gadījumā minētā problēma vairāk nav aktuāla. Cits variants, kā izvairīties no lieko mašīnu pieskaitīšanas ir attēlots 4.34.c. Šeit kamera ir novietota mazāk slīpi pret ceļu.

4.6 Kopsavilkums un secinājumi

Projekta laikā ir izveidots un kodā īstenots pret skaitļošanas jaudām efektīvs un pret apgaismojuma un laika apstākļu maiņu noturīgs algoritms, kas no video kameras datiem spēj atklāt automašīnas. Atklāto mašīnu kustība kadrā var tikt izsekota. Algoritms spēj noteikt mašīnu parametrus – augstumu, garumu,

platumu, ātrumu, kustības virzienu. Pēc noteiktajiem parametriem automašīnas tiek klasificētas vieglajās un smagajās. Nakts apstākļos algoritms var skaitīt automašīnas, bet to parametri netiek noteikti. Algoritms spēj noteikt pār sarkano gaismu braucošas automašīnas, pat ja interesējošais apgabals, kuru mašīnas nedrīkstētu šķērsot pie attiecīgā signāla, ir gājēju pāreja.

Ņemot vērā algoritma iespējas un projekta laikā veiktos testus, apstākļos, kad video kamera ir uzticams sensors (ne biezas miglas, sniegputeņa vai spēcīga lietus apstākļos), algoritms ir būtisks un noderīgs posms multifunkcionālā inteliģentā transporta sistēmas punktā.

Papildus metodes apraksti atrodami publikācijās un patenta pieteikumā:

- R. Kadiķis and K. Freivalds, "Efficient video processing method for traffic monitoring combining motion detection and background subtraction,"Lecture Notes in Electrical Engeneering, volume 221, issue vol. 1, 131-141, (2013)
- R. Kadiķis and K. Freivalds. "Vehicle classification in video using virtual detection lines." Sixth International Conference on Machine Vision (ICMV 13). International Society for Optics and Photonics, (2013)
- WO2013187748 A1. System and method for video-based vehicle detection/ Freivalds K., Kadiķis R., Greitans M. – Iesniegts 12.06.2012. Nr. PCT/LV2012/000009; Publicēts 19.12.2013.

Nodaļa 5

Transporta līdzekļu reģistrācija numuru noteikšana un algoritmi

5.1 Numuru atpazīšanas algoritma apraksts

Videokameru, kuras paredzēts izmantot LPR sistēmā, prasību analīze parādīja, ka līdzās ar kopējām, pastāv arī rinda specifisku prasību, kas saistītas ar LPR attiecīgo aplikāciju īpatnībām. Pie vispārējām prasībām pieskaitāmas, piemēram, līniju skaits kadrā (vēlams ne mazāk, kā 480 līniju), darbība arī sliktā apgaismojumā, darbība «āra» apstākļos, t.i. temperatūras diapazonā no +50 līdz -35. Specifiskās prasības nosaka LPR konkrētais pielietojums. Ja sistēmu paredzams izmantot, lai noteiktu numurus automašīnām, kas pārvietojas pa ceļu, tad ņemot vārā automašīnu pārvietošanās ātrumu, attiecīgajām videokamerām nepieciešama iespēja uzstādīt fiksētu slēdža «shutter» ātrumu, kura vērtība ir vismaz 1/1000s, bet vēlams vēl ātrāku. Vēl efektīgākas šim pielietojumam ir videokameras ar „rolling shutter" vai „global shutter". Ātri kustīga objekta iegūstamie attēli, izmantojot dažāda veida slēdžus, parādīti attēlā 5.1.

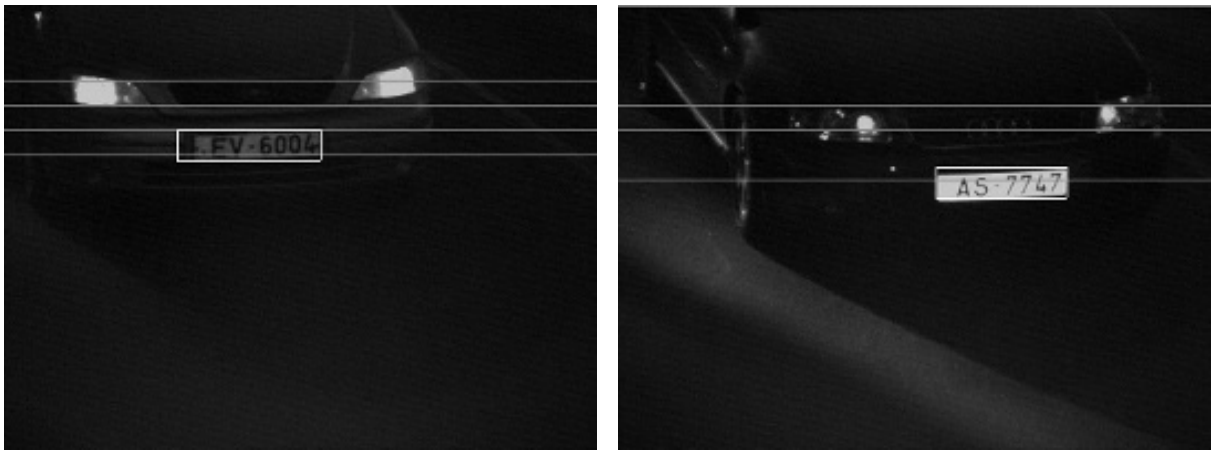
Att. 5.1: Attēlu paraugi ātri kustīgiem objektiem izmantojot dažāda veida slēdžus

Pašlaik tirgū vēl ir pieejamas ļoti maz kameras, kas attēlu iegūšanai izmanto „global shutter". Viena no pirmajām tāda veida kamerām ir firmas IqinVision kamera IQeye510. Tā kā numuru atpazīšana jāveic kā dienā tā arī naktī, t.i. pie laba apgaismojuma un tumsā, tad vienīgais risinājums ir infrasarkano staru prožektora izmantošana, kā arī atbilstošu kameru izvēle, kas, spējīgas darboties infrasarkanajā diapazonā un aprīkotas ar parastās gaismas filtru, lai novērstu attēla pārgaismošanu, piemēram, ar automašīnas gaismām. Ja nav iespējams izmantot videokameru ar „global shutter", tad nepieciešama prasība ir vismaz fiksēta slēdža konfigurēšanas iespēja. Automašīnu numuru atpazīšanas procedūrā var izdalīt vairākus neatkarīgus etapus:

- kameras sūtītā kadra saņemšana; - kadra apstrāde, lai paaugstinātu attēla kvalitāti atbilstoši LPR darbības uzlabošanai (histogrammas paplašināšana, trokšņu filtrācija utt.); - reģistrācijas numura (RN) plāksnītes meklēšana kadrā; - RN plāksnītes attēla apstrāde un iespējamo numura zīmes simbolu atrašanās zonu izdalīšana; - grafisku attēlu atpazīšanas līdzekļu OCR (optical character recognition) izmantošana lai noteiktu konkrētus RN simbolus. Iepriekšminētie etapi atkārtojas ar visiem kadriem, kuros tiek atrasts RN. Katram atpazītajam simbolam OCR dod atpazīšanas ticamības novērtējumu. Tad pēc visu vienādo RN kadru apstrādes tiek veikti sekojoši etapi: - iegūto rezultātu statistiska analīze un visvarbūtīgāko simbolu izvēle; - iegūtās informācijas – RN un to pavadošās informācijas (laiks, labākais kadrs utt.) saglabāšana datubāzē.

Primārai attēla apstrādei šajā etapā jānodrošina minimāls laiks RN plates meklēšanai. Tika veikta analīze, lai noteiktu raksturlielumus, kas raksturo attēla zonu, kurā atrodas plate ar reģistrācijas numuru

(PRN) un izvēlēts optimāls pēc kritērija laiks/kvalitāte. Attēla primārās apstrādes būtība ir samazināt tā izmērus. Tiek veikti sekojoši sākuma attēla pārveidojumi. Sākuma attēls tiek sadalīts mini zonās ar izmēru 20 x 4 pikseļi; katra mini zona tiek aizvietota ar vienu pikseli, kura intensitāte tika noteikta kā mini zonas pikseļu intensitāšu starpība I(max) - I(min). Šāda pārveidojuma rezultātā iegūstam mini kadru, kura izmērs 64 reizes mazāks par sākuma attēla izmēru. Šo pārveidojumu rezultātā iegūtais mini kadrs tiek analizēts saskaņā ar izstrādāto metodi. Tās būtība ir noteikt plates ar reģistrācijas numuru īpatnības un mini kadra analīze, ar mērķi atrast attiecīgās zonas. Tika izmēģināti vairāki kritēriji, kas raksturo PRN attēlu. Rezultātā, kā optimālākais tika izvēlēts sekojošs: attēla apgabalam, kurš satur PRN, ir raksturīgs noteikts daudzums melno un balto pikseļu. Nav apskatīti taksometru PRN (ar dzeltenu fonu), diplomātiskie PRN (ar sarkanu fonu) un citi speciālie PRN. Izveidotajā algoritmā vērā ir ņemts fakts, ka zīmes uz PRN tiek attēlotas melnā krāsā, bet PRN fons - baltā. Līdz ar to mini zonas, kurās ir pietiekoši daudz melno un balto pikseļu tiks attēlotas ar augstāku intensitātes vērtības pikseli (gaiši toņi), bet mini zonas ar mazu vai nulles atšķirību I(max)-I(min) raksturīgas viendabīgiem krāsu laukumiem – gaišiem, viendabīgi tumšiem vai pelēkiem toņiem. Tad, izmantojot slieksni, tiek veikta mini kadra binarizācija: visi pikseļi ar intensitātes vērtību zemāku par slieksni, tiek pārkrāsoti melnā krāsā, bet visi pikseļi ar intensitātes vērtību augstāku par slieksni – baltā. Baltās zonas norāda uz iespējamo PRN atrašanās vietu. Tālāk tiek veikta balto zonu papildus filtrācija, kas balstīta uz zināšanām par PRN ģeometriskajiem izmēriem. Ja pieņem, ka kamera fiksē mašīnas parādīšanos, sākot no 30 m attāluma, tad, zinot numura reālos izmērus, (augstumu un platumu) var noteikt kāds daudzums pikseļu attēlo PRN. Līdz ar to visas baltās zonas, kuru izmērs mazāks par noteikto, tiek pārkrāsotas melnas un ar to tiek izslēgtas no tālākas analīzes. No atlikušajām baltajām zonām izslēdzam arī tās, kurām kaut viens izmērs ir mazāks nekā nepieciešams. Lai ekonomētu skaitļošanas resursus, var, izmantojot ātrdarbīgus algoritmus, atrast tos attēla apgabalus, kuros attēlotie objekti kustās. Mūs neinteresē automašīnas attēls kopumā, bet tikai numura zīme, tad iespējams pielietot mūsu, iepriekš izstrādāto, algoritmu numura zīmes atpazīšanai ne visā attēlā, bet tikai tajos attēla apgabalos, kuros attēloti kustīgi objekti. Attiecīgi, ja attēlos nav kustīgu objektu, bet tikai tukšs ceļa posms, reģistrācijas numuru atpazīšanas algoritms netiks pielietots, un skaitļošanas resursu patēriņš būs minimāls. Kustības noteikšanas algoritma darbības princips ir samērā vienkāršs. Tas ir iterācijas algoritms, kas tiek pielietots katram kadram. Katrā iterācijā tiek salīdzināti divi laikā nobīdīti kadri. Abos kadros tiek skenēti horizontālo pikseļu rindas fragmenti. Fragmentu koordinātes pa Y asi abos attēlos sakrīt. Intervāls starp fragmentiem tiek uzdots ar parametru. Mūsu eksperimentos tā vērtības bija no 10 līdz 24 pikseļiem. Jo lielāks būs izvēlēts šis intervāls, jo ātrāk algoritms darbosies, bet būs zemāka precizitāte.



Att. 5.2: Atrastie numura plates reģioni

Diviem fragmentiem ar vienādām Y koordinātēm: 1. Skaitītājam piešķiram nulles vērtību 2. Divos, laikā nobīdītos, attēlos savstarpēji tiek salīdzinātas pikseļu, ar vienādām X koordinātēm, intensitātes. Ja pikseļu pāra absolūto intensitātes vērtību atšķirība pārsniedz noteikto parametra vērtību, tad skaitītāja vērtību palielinām par 1. 3. Ja pēc fragmenta visu pāru apstrādes skaitītāja vērtība pārsniedz citā parametrā noteikto vērtību, tiek uzskatīts, ka rindā ar šādu koordināti Y ir kustība. Kustības noteikšanas algoritma darbības rezultāts tiek izmantots īpašā veidā. Skenējot no attēla apakšas uz augšu, visās līnijās, kurās noteikta kustība: 1. tiek apstrādāts attēla apgabals visa kadra platumā un ar uzdotu augstumu. 2. Ja šajā apgabalā tiek atrasta numura plate, tad algoritma darbība tiek uzskatīta par veiksmīgu. 3.

Pretējā gadījumā iepriekšminētā operācija tiek atkārtota ar nākošo rindiņu, kurā tiks noteikta kustība. To atkārto, līdz tiek atrasta numura plate 5.2, vai apstrādāti visi apgabali, kuros kustība tikusi detektēta. Ja skaitļošanas resursi ir ierobežoti, ieteicams izmantot modificētu kustības noteikšanas algoritmu. Tiek analizēti divi kadri, kas ir iegūti no videokameras un seko laikā viens aiz otra (apzīmēsim tos ar K un K1). Katrs kadrs nosacīti tiek pārklāts horizontālām līnijām Yj, j = 1, N ar noteiktu un mainīgu attālumu starp tām 5.3. Pēc tam pa pāriem, kadros K un K1, tiek salīdzināta pikseļu Pxyi(K) un Pxyi(K1), kas atrodas uz līnijas Yi un kuras X koordināta ir vienāda abos kadros, intensitāte. Pēc tam pa pāriem, kadros K un K1, tiek salīdzināta pikseļu Pxyi(K) un Pxyi(K1), kas atrodas uz līnijas Yi un kuras X koordināta ir vienāda abos kadros, intensitāte. Ja pikseļu Pxyi(K) un Pxyi(K1) intensitātes starpība ir lielāka par noteiktu sliekšņa vērtību, tad kustības apjoma skaitītājs tiek palielināts par vienu vērtību. Pēc visu līnijas Yi pikseļu pāru intensitātes salīdzināšanas kadros K un K1, skaitītāja "count threshol" vērtība tiek salīdzināta ar noteiktu kustības apjoma slieksni "motion threshold".

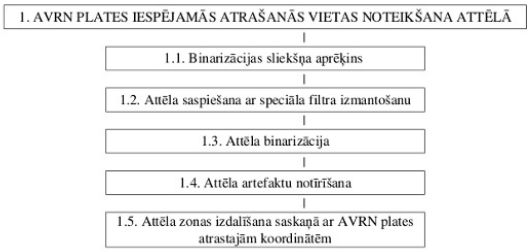
Att. 5.3: Atrastie numura plates reģioni

Ja tiek konstatēts ka "count thresholdi" lielāks nekā "motion threshold", tad tiek uzskatīts, ka kadra K1 līnijā Yi ir kustība. Attēlā ir parādītas tādas līnijas. Rezultātā automobiļa reģistrācijas numura zīmes plates var meklēt nevis pa visu kadra laukumu, bet tikai tajos apgabalos, kuros ir konstatēta kustība. Tādējādi samazinās kadra apstrādes laiks un samazinās apgabalu ar zemu kustības apjomu analīzes nepieciešamība, ko var radīt, piemēram, koku lapu kustība vai nelielu objektu (cilvēku, dzīvnieku u.tml.) kustība. Attēlā 5.4 ir parādīti taisnstūra kadra apgabali, kuros turpmāk tiek veikta automobiļa numura zīmes plates meklēšana.

Att. 5.4: Apgabali, kuros tiek meklēta numura plate

Taisnstūri tiek veidoti sekojoši: 1. pie K1 kadra līnijas Yi koordinātas, kurā konstatēta „kustība" sākumā tiek pieskaitīta vērtība "padding bottom" (pielaide uz leju) no līnijas Yi. Rezultātā tiek iegūta YB vērtība kas ir taisnstūra apakšējā robeža; 2. pie K1 kadra līnijas koordinātas Yi tiek atņemta vērtība "padding top" (pielaide uz augšu) no līnijas Yi. Rezultātā tiek iegūta vērtība YT augšējai taisnstūra robežai; 3. tiek atrasts līnijas Yi pirmais pikseli pa labi un pirmais pa kreisi kuros konstatēta kustība pēc augstāk noteiktā kritērija un attiecīgi šo pikseļu X koordinātas XL un XR; 4. taisnstūris, ko veido koordinātas YB, YT, XL un XR, nosaka reģistrācijas numura plates meklēšanas apgabalu. Analoģiski

tiek noteikti un tālāk analizēti taisnstūru apgabali visām kadra līnijām, kurās ir detektēta „kustība”. Mēs izmantojam tādas ”padding bottom” un ”padding top” parametru vērtības, kuras ir lielākas vai vienādas ar intervāla vērtībām starp kadra blakus līnijām. Šādā gadījumā taisnstūra apgabali, kas nosaka numura plates meklēšanas zonu, blakus līnijām, pārklājās. No vienas puses, tas rada zināmu redundanci, no otras, nodrošina lielu numura plates atrašanas varbūtību, sakarā ar gadījumiem, kad numura plate atrodas precīzi starp kadra līnijām. Tā kā ”padding bottom” un ”padding top” vērtības pārsniedz intervāla vērtību starp blakus līnijām ne vairāk, kā 1.3 reizes, tad pārklājuma apjoms un, attiecīgi, papildus analīzes laiks - neievērojami. Turpmākais automobiļu valsts reģistrācijas numuru (AVRN) noteikšanas algoritms sadalīts divos līmeņos: pirmais, (AVRN) plates atrašanās vietas noteikšana attēlā 5.5, otrais - (AVRN) noteikšana.



Att. 5.5: Reģistrācijas numura plates atrašanās vietas noteikšanas algoritma blokshēma

Savukārt pirmā un otrā 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 slieksni, 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. 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 5.6.

Att. 5.6: Reģistrācijas numura noteikšanas algoritma blokshēma

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*N) apzīmē ar f, bet filtra masku (ar izmēru m*n) ar w, tad, maskas un zonas konvolūcijas operāciju var izteikt pēc **??** formulas:

kur g(x,y) – konvolūcijas operācijas rezultāts; a=(m-1)/2; b=(n-1)/2; x=0,1,2,...,M-1; y=0,1,2,...,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

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

Att. 5.7: Konvolūcijas aprēķina formula

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 Rc. Ja Rmin < Rc <= Rmax, 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 Rmin tiek izmantots pikseļu skaits attēlā „1", bet kā Rmax – simbola „W" pikseļu skaits. Pēdējā etapā tiek startēta OCR apakšprogramma, kurai, kā ieejas dati, tiek secīgi padoti visi atrastie AVRN plāksnes apgabali, kas satur saistīto pikseļu kopumu. Tad tiek pārbaudīta atrasto simbolu atbilstība Latvijas AVRN 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. 5.8 parādīts, kā izskatās automobiļa attēls, izmantojot IR prožektoru. Jāpiezīmē, ka attēlu tumšais fons, uz kura redzama AVRN plāksne, tāds ir vienmēr, neatkarīgi no diennakts laika.

Att. 5.8: Iegūtais attēls izmantojot IR prožektoru

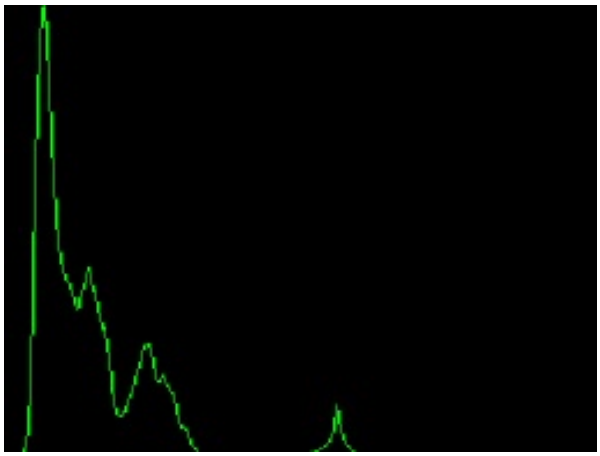
Sākuma attēla histogramma parādīta attēlā 5.9. Lielākās pikseļu masas intensitāte atrodas tuvāk nullei. Atsevišķs, neliels izvirzījums histogrammas labajā pusē, norāda uz AVRN plāksnes gaišo pikseļu atrašanos attēlā. Vispārējā gadījumā, attēla intensitātes vidējā vērtība būs diapazonā no 20 līdz 60. No histogrammas redzams, ka pikseļu intensitātes dispersija ir samērā neliela. Intensitātes vidējā vērtība un dispersija tiek izmantota, lai noteiktu binarizācijas slieksni.

Attēlā 5.10 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).

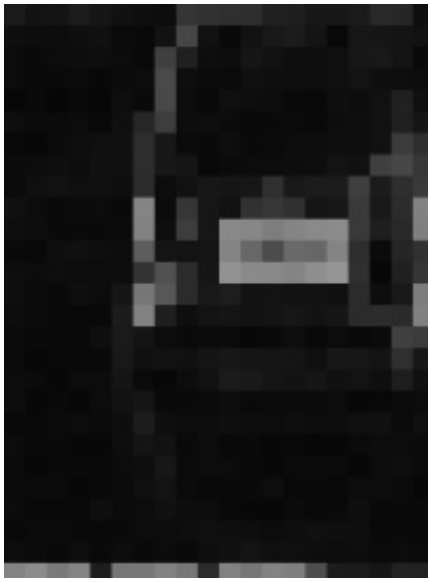
Attēls pēc binarizācijas attēlots 5.11. Šis un turpmākie attēli tiek mākslīgi izstiepti pa horizontāli, lai uzskatamāk demonstrētu piedāvātās tehnoloģijas pielietojuma efektu turpmākos etapos. Patiesībā, visu saspiesto attēlu formāts ir tāds, kā parādīts attēlā. Jāatzīmē, ka līdzās ar AVRN plāksnes zonu, 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.

Iegūtā rāmīša 5.12 koordinātu pārnešana uz oriģinālā attēla 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 sakrišanas precizitāte. Jo precīzāk tie sakrīt, jo precīzāki un efektīgāki būs otrā līmeņa etapu darbības rezultāti.

Dziļāks pirmā līmeņa rezultātu novērtējuma kritēriju apraksts sniegts nākošajā nodaļā. Kadri ar



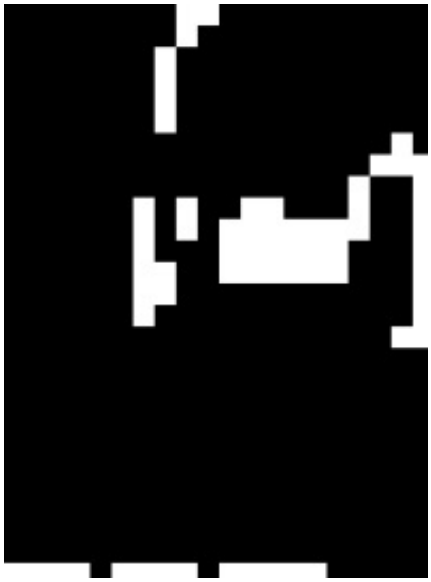
Att. 5.9: Sākuma attēla histogramma



Att. 5.10: Attēls pēc saspiešanas un filtrācijas

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. Nākošajā etapā realizēta saistītu pikseļu kopumu meklēšana, kas apmierina nosacījumu Rmin < Rc <= Rmax un artefaktu dzēšana. Pēdējā etapā, ar OCR palīdzību, tiek veikta simbolu atpazīšana. Šo etapu darba rezultāti parādīti attēlos 5.13.

Viens no mūsu sistēmai pievienotajiem moduļiem ir uz Tesseract OCR bāzēta teksta atpazīšanas sistēma. Visi attēla apgabali, kuros MITSRegNo paredz numura zīmes atrašanos, tiek nodoti šim jaunajam modulim. Jāpiezīmē, ka Tesseract paredzēts drukāta teksta lappušu atpazīšanai un pilnvērtīgai tā izmantošanai mūsu sistēmā nepieciešama papildus optimizācija. Teksta atpazīšanas sistēmai Tesseract paredzēta iespēja tās trenēšanai darbam ar konkrētu valodu un šriftu. Trenēšanas procesā tiek ģenerēts speciāls fails, kas vēlāk tiek izmantots sistēmai darbojoties. Tesseract trenēšanas procesam nepieciešams sagatavot attēlus, kuros kvalitatīvi attēlots teksts, kas ir identisks atpazīstamajam. Pēc vairāku utilītu pielietošanas un starpfailu ģenerācijas, notiek Tesseract OCR optimizācijas darbs, kas veicams ar roku.



Att. 5.11: Attēls pēc binarizācija

Att. 5.12: Reģistrācijas numura noteikšanas algoritma blokshēma

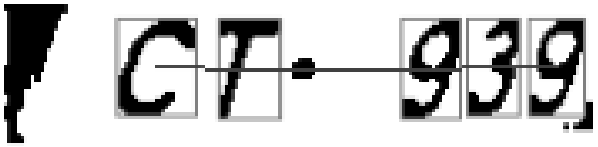
Att. 5.13: Iegūtie attēli pēc apstrādes

Citiem vārdiem, jāļauj Tesseract veikt treniņa teksta atpazīšanu, un pēc tam ar roku izlabot tā kļūdas, norādot pareizu visu simbolu atrašanās vietu, kā arī visus teksta simbolus UTF-8 kodējumā. Tomēr jāatzīmē, ka Tesseract arī bez papildus treniņa samērā labi nosaka numura zīmju tekstu 5.14, lai gan atsevišķos gadījumos tas varētu būt labāks.

Lai turpinātu numura zīmju teksta atpazīšanas sistēmas uzlabošanu, sākām izstrādāt savu teksta atpazīšanas apakšsistēmu. Mūsu gadījumā numura zīmes teksta atpazīšanas process tiek sadalīts divos galvenajos etapos: - Atsevišķu simbolu atrašana. - Teksta zīmes noteikšana katrā iepriekš atrastajā simbolā. Simbolu detektēšanas procesā izmantojam daudzus tos pašus primitīvos attēlu apstrādes

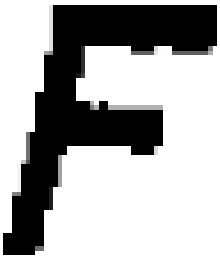
Att. 5.14: Atrastās numura zīmes

algoritmus, ko izmantojam pašu numuru zīmju atrašanai. Tie ir: - Binarizācija. - Noteikta garuma nogriežņu pārkrāsošana. - Savstarpēji savienotu pikseļu grupu noteikšana. Visi šie algoritmi jau iepriekš apskatīti. Simbolu atrašanas process ir sekojošs. Numura attēls tiek binarizēts, un veikta liela garuma horizontālo nogriežņu pārkrāsošana. Šādi tiek nodalītas atsevišķas savienotu pikseļu grupas kas pieder dažādiem simboliem. Tad notiek taisnstūru koordināšu detektēšana, kas aptver katru savienoto pikseļu grupu, piederīgu atsevišķiem simboliem. Daļa no objektiem tiek atfiltrēta to neatbilstošo izmēru dēļ. Atlikušie objekti tiek šķiroti pēc X koordinātes un to viduspunktiem. Ja atrastie objekti tiešām ir simboli uz numura zīmes, tie atradīsies vienā rindā uz horizontālas līnijas 5.15.



Att. 5.15: Uz vienas līnijas atrastie simboli

Tiek pielietota savdabīga atrasto objektu izvērtēšana to atbilstībai reālai automašīnas numura zīmei, kas piešķir skaitlisku novērtējumu tam, ka atrasts ir tieši numurs, ko apzīmējam ar P. Ņemam vērā arī Y koordinātu un augstumu izkliedi starp atsevišķiem objektiem un citus kritērijus. Sakarā ar to, ka šī algoritma kritiskākā vieta ir binarizācijas process, tieši pareiza binarizācijas sliekšņa noteikšana, ļauj izmantot šo metodi. Aprakstītais algoritms tiek izmantots binarizācijas sliekšņa vērtību sērijai un katru reizi tiek sarēķināta kritērija P vērtība. Tiek izvēlēta un pielietota tā binarizācijas sliekšņa vērtība, pie kuras P vērtība ir maksimālā. Pēc katru simbolu aptverošu taisnstūru koordinātu iegūšanas, var izpildīt to teksta vērtības noteikšanu. Šādas sistēmas veiksmīgai darbībai, simbolu attēliem ir jābūt samērā tīriem, jo ņemt vērā un izlabot visus iespējamos simbolu kropļojumu variantus nav iespējams. Tomēr jāpiezīmē, ka piedāvātā metode ir samērā noturīga pret ģeometriskiem attēla kropļojumiem, bet, diemžēl, neatrisina netīras numura zīmes problēmu. Turpmākais darbs notiek ar katra simbola monohromo attēlu. No simbola attēla varam izveidot tā saukto simbola "Profilu" 5.16.



Att. 5.16: Atrastais simbols

Attēlu skenējam pa rindām. Katrai rindai atrodam pašu malējo kreiso un pašu malējo labo melnās krāsas pikseļa X koordinātu, t.i. pikseli, kas pieder simbola attēlam. Uz šo vērtību bāzes var aprēķināt divas jaunas vērības. Simbolu šķērsojošā horizontālā nogriežņa garumu un šī nogriežņa centra koordināti. Abas vērtības tiek normalizētas diapazonam [0.0, 1.0]. Simbolu attēlu augstumi var atšķirties, tāpēc iegūtais profils tiek interpolēts līdz galējam konstantam punktu skaitam. Visus simbolus var iedalīt vairākās klasēs pēc to platuma profiliem, piemēram, H, N, ... , A , V , F , L , E , T, 7 , 9 , 6, G utt. Katrai klasei izgatavots profila šablons. Detektēšanas procesā katram simbolam ģenerējam profilu, kuru salīdzinām ar katru no esošajiem šabloniem. Iegūstam simbola piederību kādai no klasēm. Dažās no klasēm ir pa vienam iespējamam simbolam, tas ļauj uzreiz noteikt simbola teksta vērtību. Ja klasē iespējami vairāki eksemplāri, nepieciešama papildus analīze. Izmantojot katra nogriežņa centra koordinātes, iespējams noteikt tā slīpumu uz labo vai kreiso pusi, un pēc tam, izmantojot standarta ģeometriskos pārveidojumus izlabot slīpumu. Precīzu teksta vērtību simbolam var noteikt analizējot tā attēla atsevišķu apakšapgabalu intensitāti. Kā alternatīvs variants, var tikt izmantots Tesseract OCR atsevišķu simbolu atpazīšanas režīmā.

Izstrādājot automašīnu numuru atpazīšanas sistēmas arhitektūras finmāla variantu, tika nolemts pielietot moduļu arhitektūru, kas ļāva veikt elastīgu sistēmas konfigurēšanu, kā arī nākotnē paplašināt tās funkcionālās iespējas. Aprakstā izmantosim sekojošus terminus: • Objekts - jebkurš taisnstūris, kas nosaka attēla (kadra) apgabalu. Kadrs ir sākuma objekts, kuram ir maksimālais izmērs. • Transformācija - atpazīšanas sistēmas operācija. Transformācija paredz manipulācijas ar objektiem viena kadra kontekstā. Katra transformācija ieejā saņem vienu vai vairākus objektus un veido nulle, vienu vai vairākus jaunus objektus. • Avots - ieved sistēmā jaunus objektus, bet nepieņem nekādus objektus ieejā. Piemēram, līdz ar katru kadru, sistēmā ienāk objekts, kas apraksta tā robežas. • Apstrāde - izpilda apstrādi objektā, bet neveido jaunus objektus. Piemēram, teksta atpazīšanas sistēma analizē attēla daļu, veic kadra binarizāciju, kadra tīrīšanu u. c. • Mezgls - Avots, apstrāde vai transformācija. Atpazīšanas sistēmu var attēlot grafa veidā, kur katrs mezgls var būt avots, apstrādātājs vai veikt vienu no transformācijām. Avots vienmēr ģenerē vismaz vienu objektu. Katras transformācijas rezultāts - nulle, viens vai vairāki objekti. Pašlaik tika izmantotas sekojošas transformācijas: • “Kustības detektors horizontālās līnijās” - atrod “kustību” horizontālās līnijās. Izmanto iepriekšējā kadra kopiju. Salīdzinot pašreizējo kadru ar iepriekšējo, nosaka kadra apgabalus, kur ir kustība un veido šiem apgabaliem atbilstošus jaunus objektus. • “Zonu ģenerators” - rada virkni objektu (zonu) kadrā, turpmākai kontrasta noteikšanai. • “Zonas pārveidotājs” - objekta izmēra mainīšana. • “Kontrasta detektors” - veido jaunus objektus kadru apgabalos ar augstu kontrastu. • “Pārklājušos objektu apvienošana” - apvieno objektus kas savstarpēji pārklājas. Pārbauda ienākušo objektu pārklāšanos. Uz pārklājušos objektu bāzes ģenerē jaunus objektus, un nosūta uz izeju. Objekti, kas ne ar ko nepārklājas arī tiek nosūtīti uz izeju. • “Teksta detektors (OCR)” arī ir viens no sistēmas mezgliem. • Iespējami arī citu veidu mezgli. Iespējamas dažādas konfigurācijas, atkarībā no pieejamajiem skaitļošanas resursiem, kā arī no atpazīšanas uzdevuma nostādnes aspekta. Sistēma darbojas iteratīvi un vienlaicīgi apstrādā vienu kadru. Katrs mezgls var piekļūt pašreizējam kadram, kā arī var glabāt kadru kopijas, ja tas nepieciešams. Objekti tiek nodoti no mezgla uz mezglu bez nosacījumiem, vai arī saskaņā ar kādu kritēriju. Mezgli var glabāt objektu kopijas un apkopo noteiktu objektu skaitu, ja tas nepieciešams. Mezgli un citi sistēmas elementi var ģenerēt notikumus, un reaģēt uz notikumiem, šādi tiek realizēts grafa mehānisms. Sistēma realizēta programmēšanas valodā ”c++”, izmantojot Qt freimvorku. Tiek izmantota arī valoda ”ecmascript262(QtScript)”. Skriptu valodas izmantošana ļauj viegli īstenot elastīgu sistēmas konfigurāciju. Kā teksta atpazīšanas modulis tiek izmantota sistēma Tesseract-OCR. Tālāk doti sistēmas darbības rezultātu piemēri, izmantojot šo OCR moduli. Katrā no gadījumiem dots teksts, kas tika iegūts no vairākiem, laikā nobīdītiem kadriem, kuros redzama automašīna. Mūsu agrāk izstrādātais modulis kadrā atrod vairākus apgabalus, kuri, iespējams, satur numuru. Tad katrs no šiem apgabaliem tiek nodots apstrādei uz Tesseract. Katra rinda satur apstrādes rezultātus kādam no iepriekšminētajiem objektiem. Piemēram, detektējot automašīnas numuru AF7770 iegūti rezultāti: 43;â –5Â» /. I AF 7770] 17 .12 /4//” I AF-7770’ Lai veiktu automašīnu numura zīmju atpazīšanas sistēmas darbības analīzi un optimizāciju, apstrādes procesā radušos notikumu reģistrācijai tiek veidota un izmantota datubāze. Datubāzes struktūrai jāatbilst algoritma elastīgajai struktūrai. Jābūt iespējai izsekot katru sistēmas veikto operāciju. Notikumu reģistrāciju darbojoties veic katrs mezgls. Galvenās datubāzes tabulas: - Kadru eksemplāru tabula. Jauns kadra eksemplārs tiek veidots ar katru atpazīšanas algoritma iterāciju. FrameInstances • id (pk) • FrameSources.id (fk) - Atsaucas uz kadru avotu tabulu. • kadra numurs • kadra laiks - Kadru avotu tabula. Lauks “type” norāda uz to, kādā veidā iegūts kadrs. Lauks “name” maina savu jēgu, atkarībā no lauka “type” vērtības. Iespējami vairāki varianti: • Kadrs iegūts no video. Lauks “name” satur faila nosaukumu. • Kadrs iegūts no atsevišķa faila(png,

jpg, ...). Lauks “name” satur faila nosaukumu. • Kadrs iegūts no IP kameras. Lauks “name” satur kameras URL. • Kadrs iegūts no MITS iekšējās kadru izsūtīšanas sistēmas. Nosūtošās programmas nosaukums. FrameSources • id (pk) • type • name - Objektu tabula. Koordinātes: top,left,right un bottom apraksta tieši pašu objektu. Objects • id (pk) • left • top • right • bottom - Transformāciju tabula. Lauks Nodes.id norādīs uz grafa mezglu, kurš veicis apstrādi. Transformations • id (pk) • Nodes.id (fk) - mezgls, kurš veicis šo transformāciju. • FrameInstance.id (fk) - kadra eksemplārs, ar kuru veikta transformācija. - Objekts/Transformācija tabula Saista transformāciju ar objektiem, kas tajā piedalījušies, kā ieejas objekti, vai kā rezultāts. TransformationsToObjects • Transformations.id (fk) - Atsauce uz transformāciju. • Objects.id (fk) - Atsauce uz vienu no objektiem, kas piedalījušies šajā transformācijā. • object role - Norāda uz objekta lomu transformācijā (ieejas objekts /rezultāta objekts). - Mezglu tabula. Lauks “type” apraksta mezgla tipu. Lauks “settings.id” atsaucas uz ierakstu vienā no parametru tabulām. Katram mezgla tipam ir sava parametru tabula, tā kā katram apstrādes algoritmam ir savi parametri. Ir pieņemts, ka darbības laikā mezgla parametri nemainīsies. Nodes • id (pk) • type • settings.id (fk)

5.2 Kameras novietojuma 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 AVRN plāksnes izmērs visos attēlos ir aptuveni vienāds. Programmatūras izstrādes un pilnveidošanas gaitā, lai kvalitatīvi analizētu tās darbības rādītājus dažādos diennakts laikos un vides laika apstākļos, tika nokomplektēta un uzstādīta automašīnu numuru noteikšanas sistēma. Šim nolūkam tika sagatavota āra apstākļiem piemērota video kamera un infrasarkano (IR) staru prožektors, bet dators un pārējās komponentes ievietotas atbilstošā āra izpildījuma skapī. Šim nolūkam izgatavoti visi nepieciešamie stiprinājuma elementi. Par uzstādīšanas vietu izvēlēta auto maģistrāle ar samērā intensīvu satiksmi 5.17.

Testos izmantotā kamera tika aprīkota ar 850nm gaismas filtru, lai, izmantojot papildus IR apgaismojumu, iegūtu, no numuru atpazīšanas viedokļa kvalitatīvus attēlus jebkurā diennakts laikā. Šāds risinājums ļauj samazināt laika apstākļu ietekmi uz numura noteikšanu. Kamera un IR prožektors stiprināts virs auto maģistrāles uz tur jau esošās konsoles. Kamera vērsta pret brauktuvi aptuveni 30 grādu leņķī 5.18. Tas ir sava veida kompromiss. No vienas puses, lai iegūtu attēlus ar iespējami maziem ģeometriskiem kropļojumiem, bet no otras puses, lai automašīna atrastos ne pārāk tālu no kameras un nodrošinātu numura zīmes kvalitatīvu izgaismošanu un redzamību, piemēram, stipra lietus laikā.

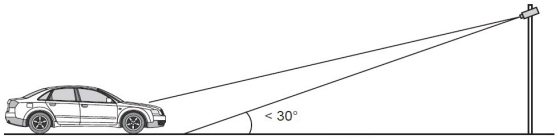
IR prožektors novietots zem kameras. Tā raksturlielumi izvēlēti tā, lai nodrošinātu pietiekošu numura zīmes izgaismošanu ne tikai krēslas stundās vai tumsā, bet arī samazinātu uz numura zīmi mestās ēnas kontrastu saulainā laikā. Testa laikā izmantota kamera ar izšķirtspēju 1280x800 un slēdža ātrumu ne mazāku par 1/1000 s, ātrumam līdz 90 km/h un 1/2000 ātrumam līdz 180 km/h. Nepieciešams atslēgt kamerā integrētās attēlu kontroles un apstrādes funkcijas, tādas, kā Dynamic Contrast un Backlight compensation. IR prožektora nepārtrauktai izmantošanai, jāatslēdz kamerā iebūvētais IR bloķēšanas filtrs.

5.3 Datu sūtīšana izmantojot GSM tīklu

Izstrādāts datu sūtīšanas modulis. Tā uzdevums nodrošināt MITS sistēmas kodolam iespēju nosūtīt informāciju īsziņas formā. Lēmumu par īsziņas sūtīšanu pieņem MITS centrālā vadības programma, bet izstrādātā programma komunicē ar GSM modemu 5.19, kas pieslēgts datoram, izmantojot USB->COM adapteri.

Risinājumā izmantotas AT komandas no ETSI (European Telecommunications Standards Institute) „GSM 07.05 for SMS” dokumenta: AT+CMGF=1 OK , īsziņas nosūtīšanai tiek ieslēgts teksta režīms; AT+CMGS=”+37199999999” > , ievadam adresāta mobilās iekārtas numuru; > message ctrl-Z +CMGS: 162 OK , ievadam nosūtāmo tekstu un nosūtam īsziņu.

Att. 5.17: Kameras novietojums virs brauktuves



Att. 5.18: Kameras novietojuma shēma

Att. 5.19: GSM datu sūtīšanas modulis