

Darbības programma: “Uzņēmējdarbība un inovācijas”
Prioritātes numurs un nosaukums: 2.1. “Zinātne un inovācijas”
Pasākuma numurs un nosaukums: 2.1.1. “Zinātne, pētniecība un attīstība”
Aktivitātes numurs un nosaukums: “Atbalsts zinātnei un pētniecībai”

Projekta Nr. 2010/0250/2DP/2.1.1.1.0/10/APIA/VIAA/086
Projekta nosaukums: ”Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija” (MITS)

Līguma noslēšanas datums: 10.12.2010.g
Projekta sākuma datums: 01.01.2011.g.
Projekta beigu datums: 1.12.2013.g.

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

Attēlu algoritmu izpēte Multifunkcionālai inteliģentai transporta sistēmas punkta tehnoloģijas izstrādei

Projekta zinātniskais vadītājs: Leo Seļavo, Ph.D. vad. Pētnieks

Projekta izpildītāji:
Vad.pētn. Dr.sc.comp. Kārlis Freivalds
Vad.pētn. Dr.sc.comp. Mihails Broitmans
Zin. Asist. Mārtiņš Liepiņš
Zin. Asist. Aivars Ševerdaks
Zin. Asist. Roberts Kadiķis
Prog. Inž. Normunds Šilinskis

Pētnieks Uldis Grunde /Profesionāla rakstura dabrs bez atlīdzības/
Zin.Asist. Artis mednis /Profesionāla rakstura dabrs bez atlīdzības/
Zin.Asist. Andris Gordjušins /Profesionāla rakstura dabrs bez atlīdzības/

Saturs

| | |
|---|----|
| Ievads | 3 |
| 1. Attēlu apstrādes metodes automašīnu atklāšanai | 4 |
| 1.1 Automašīnu atklāšana pēc to parametriem | 4 |
| 1.2 Kustības detektēšana | 7 |
| 1.3 Automašīnas atklāšana izmantojot fona uzkrāšanu | 7 |
| 1.4 Automašīnas izsekošana kadrā | 9 |
| 1.5 Automašīnu detektēšana naktī | 11 |
| 1.6 Mūsu piedāvātā metode automašīnu atklāšanai | 12 |
| 2. Attēlu apstrādes metodes nummura zīmju detektēšanai | 15 |
| 2.1 Attēla saspiešana ar speciālu masku pielietošanu | 15 |
| 2.2 Attēla binarizācija ar aprēķinātu binarizācijas sliekšni | 16 |
| 2.3 Speciālas attēla attīrīšanas metodes | 17 |
| 2.4 Attēlu apstrāde ar speciālām maskām, lai uzlabotu AVRN attēla kvalitāti | 17 |
| 2.5 Saisfīto pikseļu kopumu meklēšana | 18 |
| 3. Literatūras saraksts | 19 |

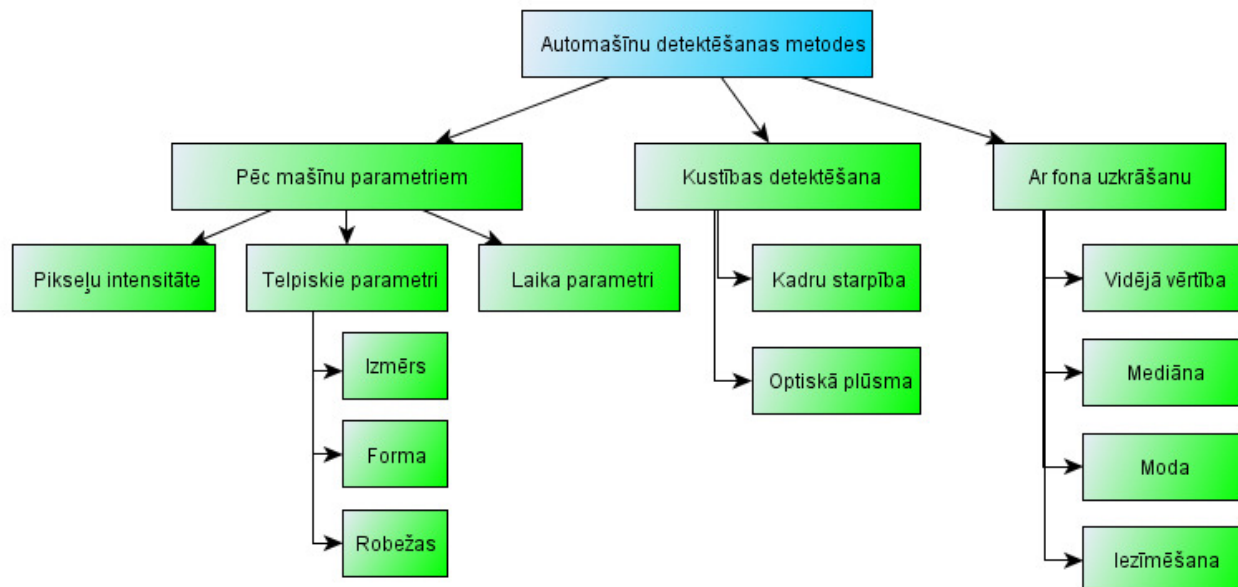
Ievads

Metodes apraksts “Attēlu algoritmu izpēte Multifunkcionālai inteliģentai transporta sistēmas punkta tehnoloģijas izstrādei” ir ERAF 2.1.1.1 aktivitātes ietvaros atbalstītā projekta “Multifunkcionāla inteliģenta transporta sistēmas punkta tehnoloģija” (MITS) aktivitātes rūpnieciskais pētījums rezultāts.

Svarīgu lomu transporta līdzekļu atklāšanā veic video datu apstrāde. Izmantojot attēlu apstrādes tehnoloģiju sniegtās iespējas, iespējams iegūt visplašāko un pilnīgāko informāciju par brauktuves noslogojumu un transporta līdzekļu izvietojumu uz ceļa. Ir izveidoti gatavi risinājumi, kur brauktuveju krustojumu regulēšanu veic balstoties tikai uz video apstrādes datiem, neizmantojot informāciju no citiem sensoriem. MITS projektā attēlu apstrādes tehnoloģijas tiek izmantotas gan transporta līdzekļu reģistrēšanai uz brauktuves, gan arī transporta līdzekļu nummura zīmju atpazīšanai. Metodes aprakstā tiek apskatītas dažādas metodes braucoša transporta līdzekļa atpazīšanai, kā arī izvēlētais numura zīmes izdalīšanas un atpazīšanas algoritms. Tiek aprakstīta arī attēlu apstrādes grupas piedāvātā metode transporta līdzekļu atpazīšanai.

1. Attēlu apstrādes metodes automašīnu atklāšanai

Automašīnu atklāšana un izsekošana ar video kameru palīdzību pasaulē jau tiek pētīta desmitiem gadu. Projekta pētījuma fāzē tika apskatītas dažādas esošas metodes, kuras sadalītas 1.1. attēlā redzamajās klasēs.



1.1 att. Automašīnu atklāšana izmantojot video apstrādi

1.1. Automašīnu atklāšana pēc to parametriem

Vienkārša un arī neefektīva metode ir nogriešanas līmeņa (sliekšņa) izmantošana automašīnu atrašanai. Metode balstās uz priekšstatu, ka mašīnas ir kompakti objekti, kuru pikseļu intensitātes atšķiras no fona pikseļu intensitātes. Tādējādi, pielietojot nogriešanas līmeni nelielos apgabalos – pielīdzinot nullei par uzdotu vērtību lielākus vai mazākus pikselus, iespējams izdalīt automašīnas. Metodes efektivitāte stipri atkarīga no izmantotā sliekšņa vērtības. Lai mazinātu apgaismojuma maiņas ietekmi uz atklāšanas precizitāti, tiek lietoti adaptīvi sliekšņi. Metodes trūkums ir ēnu atpazīšana par objektiem, kā arī nespēja atpazīt mašīnas, kuru pikseļu intensitātes ir tuvas ceļa pikseļu intensitātei. Lai uzlabotu metodes efektivitāti, iegūtajiem kadriem tiek pielietoti morfoloģiskie operatori.

Cita pieeja ir izmantot vairākus režģus interesējošo apgabalu identificēšanai (*multigrid identification*). No ieejas bildes tiek izveidotas vairākas jaunas bildes versijas ar dažādām izšķirtspējām. Tad attēlā ar vismazāko izšķirtspēju tiek meklēti interesējošie apgabali. Šo apgabalu pikselus var meklēt, vadoties pēc to intensitātes līdzības ar blakus pikseļu intensitāti, līdzīgi kā iepriekšējā nogriešanas līmeņa metodē, vai arī vadoties pēc to pašu pikseļu intensitātes sekojošos kadros, kā arī pēc pikseļu veidoto gradientu lielumiem. Meklējot objektus zemas izšķirtspējas attēlā, tiek ignorēts troksnis un par automašīnām krietni mazāki apgabali, kuri no apkārtējiem pikseliem atšķiras ar intensitāti līdzīgi kā automašīnas [1].

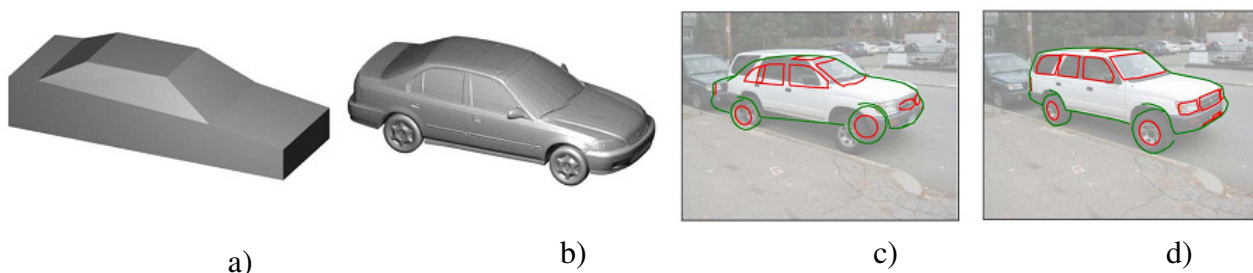
Mašīnas var meklēt pēc to telpiskajām īpašībām (formas, izmēra). Attiecīgās īpašības var meklēt ar iepriekš izveidotu automašīnu modeļu lietošanu, kuros, piemēram, iekļauti meklējamie automašīnu izmēri dažādā attālumā un dažādos skata

leņķos pret kameru.

Meklējot automašīnas ar iepriekš izveidotu modeļu jeb veidņu palīdzību, nepieciešams izveidot šablonu katram automašīnas veidam, kas ir problemātiski transportlīdzekļu dažādības dēļ. Metode pieprasa arī automašīnas intensitātes nemainīgumu, kas reālos apstākļos neizpildās. Šeit var tikt lietoti neironu tīkli, dēļ to spējas interpolēt starp dažādām zināmām (veidnēs iestrādātām) mašīnu formām un intensitātēm [1].

Pilnīgāks telpisks priekšstats par detektēto objektu tiek iegūts, ja tiek lietoti automašīnu trīsdimensiju modeļi. Šādi modeļi ietver visu informāciju par automašīnas izskatu dažādos attālumos un leņķos pret kameru, un tos var salīdzināt, piemēram, ar kadrā izdalītajiem objektu kontūriem, tādējādi nosakot, vai objekts ir automašīna. 3D modeļi ir efektīvi arī tālākos satiksmes plūsmas video apstrādes posmos – gan izsekošanā, gan klasificēšanā. Var tikt izmantoti gan vispārīgi ģeometriski objektu modeļi, gan detalizētu modeļu kolekcija (attēls 1.2.). Lai risinātu problēmu ar lielo 3D modeļu daudzumu, kas nepieciešami visu uz ceļa sastopamo transportlīdzekļu aprakstam, tiek veidoti deformējami modeļi [2].

Pretēji 3D modeļu pielāgošanai 2D attēlam, automašīnu telpiskie parametri var tikt iegūti arī no video materiāla bez modeļu lietošanas. Avotā [3] šādam nolūkam tiek lietotas divas kameras – tādējādi iegūstot stereo attēlus. No viena un tā paša punkta koordinātu neatbilstības šajos attēlos iespējams noteikt reālo punkta attālumu no kamerām 3D telpā.



1.2.att. Automašīnu 3D modeļi

a) vispārīgs modelis; b) detalizēts modelis; c) deformējama modeļa pielāgošana automašīnas robežām [2]

Mašīnu kontūri jeb robežas ir raksturīgs objektu parametrs, kas kadrā veidojas, ja blakus pikseļu vērtības krasi atšķiras. Avotos [4], [5], [6] kadrā tiek meklētas visas horizontālās un vertikālās robežas, izmantojot dažādas metodes gradienta izdalīšanai attēlā.

Automašīnas parasti sastāv no vairākām šādām robežām, turpretī uz ceļa parasti robežu ir maz. Tādējādi iespējams par automašīnām uzskatīt apgabalus ar lielu detektēto robežu blīvumu, ko var iegūt analizējot kadra histogrammu.

Cits variants ir sagrupēt atrastās robežas, tādējādi iegūstot automašīnu un citu objektu kontūrus. Piešķirot katra noteiktā kontūra pikseļiem savu vērtību, notiek kontūru apzīmēšana jeb segmentēšana pa objektiem. Robežu apvienošanas metodes var iedalīt divās grupās lokālajā un globālajā [4].

Lokālā robežu apvienošana savieno punktus, ņemot vērā to tuvā apkārtnē esošus pikseļus. Ja blakus esošie punkti apmierina kādu kritēriju, kas attiecas uz gradienta lielumu un virzienu, tie tiek apvienoti vienā kontūrā. Piemēram, kontūram var tikt pievienots punkts, kurš atrodas ne tālāk kā trīs pikseļu attālumā no punkta, kas jau iepriekš ticis pasludināts par kontūram piederīgu pikseli. Tādējādi trīs pikseļu plati

pārrāvumi starp iepriekšējā solī noteiktajām robežām tiks aizpildīti ar līniju.

Globālie robežu savienotāji ir sarežģītāki un patērē vairāk skaitļošanas resursus, toties tie ļauj iegūt precīzākus rezultātus, kā arī ļauj izvairīties no dažādu kontūru saplūšanas vienā objektā. Globālajā apvienošanā ņem vērā visus atrastos robežas punktus uzreiz un novērtē to atbilstību kādam līdžības noteikumam.

Uzlabojumus kontūru formēšanā var panākt ar Hofa transformāciju, kas avotā [7] lietota, lai automašīnas robežas apvienotu vienotā taisnu līniju kontūrā, kas nepieciešams gadījumos, kad robežu pikseļi, kas atbilst vienai automašīnai, nav savstarpēji savienotas pēc detektēšanas. Šīs pašas problēmas risināšanai tiek izmantotas arī attēlu apstrādes morfoloģiskās operācijas. Piemēram, paplašināšanas operators paplašina par uzdotu pikseļu skaitu detektētās robežas, tādējādi palielinot iespējamību, ka kontūra detektētie robežu fragmenti saskarsies un tiks tālāk atpazīti kā viena robeža.

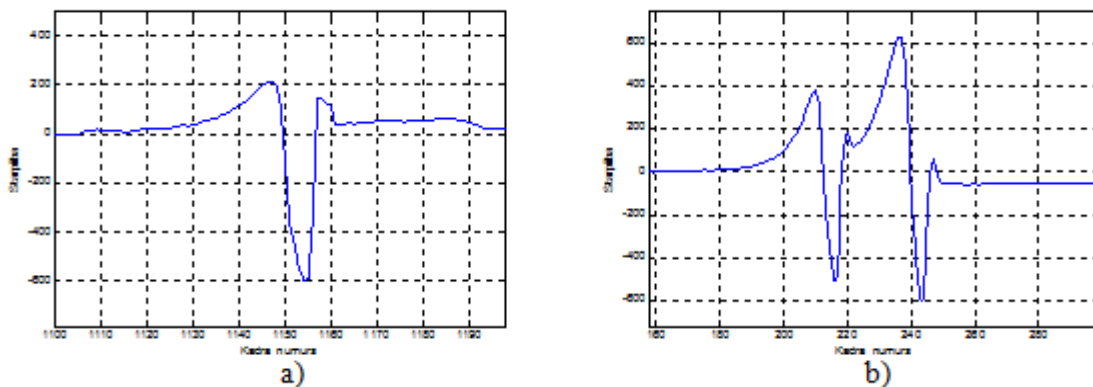
Robežu atrašanas metodes pluss ir tāds, ka robežas ir parametrs, kas saglabājas labi saskatāms daudzu ārēju apstākļu gadījumā [8].

Līdztekus apskatītajiem paņēmieniem automašīnu atklāšanai pēc to intensitātes vai telpiskiem parametriem, pastāv iespēja tās detektēt arī pēc laika pazīmēm jeb laika paraksta (*signature*), pētot pikseļu intensitātes vērtības izmaiņas laikā. Kad pikseļus šķērso braucoša automašīna, to intensitātes vērtība mainās pēc šai mašīnai atbilstoša profila – laika paraksta, kuru var izmantot par indikatoru mašīnas esamībai kadrā.

Sekojošais algoritms [9] automašīnu detektēšanai apstrādā nelielu ceļa daļu – apstrādes logu un nosaka, vai šo logu šķērso automašīnas. Logā esošo pikseļu intensitātes vērtības tiek summētas. Gadījumā, kad automašīna logu nešķērso, iegūtā summa ir atskaites vērtība. Ja, salīdzinot tekošo loga pikseļu summu ar atskaites vērtību, novērojama būtiska starpība, kas pārsniedz kādu iepriekš noteiktu sliekšņa vērtību, var pieņemt lēmumu, ka logu šķērsojusi automašīna.

Šāda algoritma realizēšanā būtiski ir iegūt precīzu atskaites vērtību. Mainoties apgaismojumam vai laika apstākļiem, izmainās arī tukša ceļa bilde. Tātad, atskaites vērtībai jābūt laikā mainīgai.

Gadījumos, kad ceļš pēc intensitātes ir tumšāks, nekā automašīna, algoritms ļauj noteikt, kad mašīna iebrauc novērošanas logā un kad izbrauc. Iebraukšanas gadījumā starpība starp pašreizējo pikseļu summu un vidējo vērtību pieaugs, bet izbraukšanas gadījumā kritīsies līdz sasniegs nullei tuvu vērtību. Attēlā 1.3. parādīti tipiski automašīnu laika paraksti, pētot detektēšanas loga pikseļu intensitātes summas izmaiņu laikā. 1.3.a. iezīmējas, ka logā vispirms aizvien vairāk parādās automašīnas priekšējā daļa, palielinot intensitātes summu. Sekojošais summas kritums norāda uz automašīnas tumšo vējstiklu, kuram savukārt seko automašīnas jumts, kas parādās kā intensitātes



pieaugums.

1.3.att. Starpība starp vidējo loga pikseļu summas vērtību un šī brīža summu

a) logu šķērso viena automašīna; b) divu tuvu braucošu automašīnu gadījumā

Ieviešot kadrā vairākus interešu logus, piemēram, divus uz vienas joslas tuvu esošus apgabalus, var aptuveni noteikt automašīnu ātrumu, ja pieņem, ka šķērsojot logus automašīnas ātrums nemainās. Jāņem vērā, ka mašīna var arī apstāties, kamēr tā ir novērošanas logā.

Metodes pluss ir tāds, ka netiek apstrādāts viss kadrs, bet tikai tā daļa – tātad metode nepieprasa lielu skaitļošanas jaudu. Šādi iegūti dati par automašīnu ātrumiem un izmēriem ir atkarīgi no video materiāla kadru skaita sekundē – jo ātrākas izmantotās kameras, jo precīzākus rezultātus iespējams iegūt. Transporta līdzekļu pikseļu intensitāte arī ietekmēs rezultātu – no ceļa stipri atšķirīgas krāsas automašīnas novedīs pie lielākām starpības vērtībām.

1.2. Kustības detektēšana

Stacionāro kameru gadījumā ceļš kadrā ir nekustīgs, savukārt automašīnām, kaut vai lai tikai nokļūtu kadrā, ir jākustas. Tādēļ automašīnas mēdz atklāt arī izmantojot kustību detektēšanu. Kustības noteikšanai kadrā var izmantot starpību starp viens otram sekojošiem kadriem, kas rezultātā atstāj tikai apgabalus, kuros pikseļiem ir izmainījusies intensitāte. Ar šādu pieeju apstājušās mašīnas nevar nodetektēt, tāpat problēmas rada automašīnas, kuru pikseļu intensitātes ir līdzīgas ceļa pikseļu intensitātei.

Vēl viena kustības noteikšanas metode balstās uz video kameras matricas atsevišķu zonu pētīšanu. Katrs kadrs ar izmēru $M \times N$ tiek sadalīts atsevišķos nelielos apgabalos ar izmēru $m \times n$. Ja vidējā pikseļu intensitāte nelielajā apgabalā pārsniedz noteiktu sliekšni, tiek uzskatīts, ka apgabals nodetektējis kustību. Visi kustību detektējošie apgabali tiek analizēti un sašķiroti balstoties uz to atrašanās vietu attiecībā vienam pret otru, kā arī uz informāciju no iepriekšējiem kadriem. Šādā veidā algoritms spēj noteikt kustīga objekta atrašanās vietu katrā kadrā. Šī metode ir pietiekami efektīva objektu izsekošanai, tomēr nav precīza to ātruma noteikšanai [4].

Automašīnu detektēšanai tiek pielietota arī optiskās plūsmas lauku pētīšana. Kustības lauks (*motion field*) ir objekta reālā kustība trīs dimensiju telpā, kas projicēta uz 2D kadra plakni. Savukārt optiskā plūsma (*optical flow*) ir pelēko toņu kustība kadra plaknē. Šie lauki ne vienmēr sakrīt [10]. Ja divos sekojošos kadros tiek detektēta optiskā plūsma, tā parādās kā pārvietojuma vektors, kas parāda pelēko toņu pārvietojuma attālumu un virzienu starp šiem kadriem.

Optiskās plūsmas metode ļauj izdalīt neatkarīgi pārvietojošos objektus pat kustībā esošas kameras gadījumā, kas varētu būt risinājums vēja brāzmu radītām kameras svārstībām. Tomēr kustības lauka atrašanas metodes bieži pieprasa lielus skaitļošanas resursus un tās ir jūtīgas pret troksni [11].

1.3. Automašīnu atklāšana izmantojot fona uzkrāšanu

Izplatīts paņēmiens objektu atpazīšanai satiksmes video ir izmantojot šī brīža kadra un tukša ceļa bildes starpību [4], [12], [13]. Šādas starpības rezultātā tiek iegūts attēls, kurā pikseļi ar neizmainītu intensitāti – tātad tukša ceļa pikseļi ir tuvu nullei. Ja uz ceļa ir automašīna, vai cits tukšam ceļam neatbilstošs objekts, tad šī objekta un tukša ceļa pikseļu intensitātes starpība visbiežāk būs atšķirīga no nulles. Fona un objekta pikseļus pēc starpības iegūšanas var atdalīt izmantojot intensitātes sliekšni, pēc formulas (1.1).

$$I(x, y) = \begin{cases} \text{Objekta pikselis, ja } |I(x, y) - F(x, y)| \geq T \\ \text{Fona pikselis, ja } |I(x, y) - F(x, y)| < T \end{cases}, \quad (1.1)$$

kur $I(x, y)$ - pikseļa intensitāte kadrā;
 $F(x, y)$ - fona pikseļa intensitāte;
 $T(x, y)$ - sliekšņa vērtība.

Fona izdalīšanu iespējams veikt ar vairākām metodēm. Vienkāršākā metode ir salīdzināt šī brīža kadru ar iepriekš manuāli uzņemtu tukša ceļa kadru. Būtisks metodes trūkums ir fona attēla neadaptēšanās apgaismojuma izmaiņām, kas āra apstākļos var notikt bieži. Tādēļ uzmanība tiek vērsta uz adaptīvām fona izdalīšanas metodēm, kas uzkrāj informāciju par fonu kādā laika periodā, līdz ir iegūts novērtējums par ceļu veidojošajiem pikseļiem.

Vienkāršākā un ātrākā no adaptīvajām metodēm ir vidējās vērtības metode. Par ceļa pikseļa vērtību tiek pieņemta vidējā pikseļa vērtība kādā laika periodā (formula (1.2)). Šī metode paredz, ka visbiežāk kādā ceļa posmā būs redzams tieši ceļš, un tie brīži, kad šajā posmā ir mašīna, pietiekami ilgā novērošanas laikā dos mazu ieguldījumu vidējās vērtības aprēķinā. Tādējādi šī metode labi darbojas mazas satiksmes intensitātes apstākļos, un avotā [14] ieteicams šādu fona ieguves paņēmieni lietot tikai šādos apstākļos. Jo vairāk kadru tiks lietots vidējās vērtības aprēķinā, jo mazāk fona attēla vērtību ietekmēs automašīnas, bet tajā pašā laikā pieaug prasības pret iekārtām – kadru saglabāšanai nepieciešama aizvien lielāka atmiņa. Lielāka izmantojamu kadru gadījumā fons arī lēnāk adaptējas.

$$F(x, y) = \sum_{i=1}^K I(x, y), \quad (1.2)$$

kur K – aprēķinā lietoto kadru skaits.

Fona pikselim var piešķirt nevis vidējo vērtību laikā, bet gan mediānu. Tādā gadījumā fona pikseļa vērtības aprēķins prasa vairāk laika [15]. Lietojot mediānas metodi, fona bildes atrašanai tiek pieņemts, ka katrs ceļa jeb fona pikselis kadrā ir redzams vairāk kā 50% laika. Tātad metode nav precīza ilgas, intensīvas satiksmes apstākļos, kā arī problēmas var radīt uz ilgu laiku apstājušās mašīnas.

Ja aprēķinā izmanto kadru skaits ir pietiekami liels, un visbiežāk kadrā parādās tieši ceļam atbilstošā pikseļa vērtība, tad visprecīzāko fona bildi iegūst nosakot par fonu tieši šo visbiežāko pikseļa vērtību, jeb modu. Avotā [7] trīs tikko apskatītās metodes pēc precizitātes sakārtotas sekojoši – moda, mediāna un vismazāk precīzā ir vidējās vērtības atrašana. Tomēr gadījumos, kad uzkrāto kadru skaits ir neliels, pikseļa intensitātes sadalījumā ceļam atbilstošā vērtība var neatbilst maksimumam jeb modai. Šādā gadījumā, kā precīzāka metode konkrētajā avotā tiek ieteikta mediānas atrašana.

Vēl viena iespēja fona attēla izveidei ir iezīmēšanas (*inpainting*) metodes pielietošana. Apskatot kadrus dažādos laika momentos, tiek nodalīti dinamiskie un statistiskie apgabali kadrā. Statiskajos apgabalos novērošanas laikā pikseļu intensitātē izmaiņu ir maz, kamēr dinamiskie apgabali ir tie, kuros notiek transporta līdzekļu plūsma. Statiskie apgabali tiek saglabāti fona attēlā, tikmēr dinamiskajiem apgabaliem fona bildē tiek uzlikta maska, kas parāda, ka konkrētie fona apgabali vēl nav iegūti. Tad fona attēlā dinamiskie apgabali tiek iezīmēti, vadoties pēc zināmajiem tuvumā esošajiem statistiskajiem apgabaliem. Uz robežas starp dinamiskajiem un statistiskajiem apgabaliem nemitīgi tiek zīmēti nelieli pikseļu bloki, kas tad tiek pieskaitīti pie statistiskā apgabala. Process turpinās, līdz fona bildē vairāk nav palikuši dinamiskie apgabali. Otra iespēja, kā aizpildīt dinamiskos apgabalus, ir iezīmējot šī paša apgabala pikseļus, kas iegūti citā laikā, kad konkrētā apgabala pikseļi bija statistiski.

Pirms iezīmēšanas veicamā dinamisko apgabalu noteikšana arī var notikt dažādi, piemēram, avotā [14] tiek pielietota kadra dispersijas atrašana pēc formulas:

$$\sigma_{kadra}^2 = q \times \frac{1}{K} \sum_{i=1}^K |I_K - \bar{I}_i|, \quad (1.3)$$

kur $|I - \bar{I}|$ - ieejas lielumu deviācija;
 q - empīriski iegūts dispersijas sliekšnis;
 σ_{kadra}^2 - kadra dispersija;
 K - aprēķinā lietoto kadru skaits.

Tiek iegūtas arī katra pikseļa dispersijas diskrēta laika logā K , veicot avotā [1] piedāvāto rekursīvo aprēķinu:

$$E(x, y, k) = \frac{K-1}{K} \cdot E(x, y, k-1) + \left(1 - \frac{K-1}{K}\right) \cdot I(x, y, k), \quad (1.4)$$

$$E_2(x, y, k) = \frac{K-1}{K} \cdot E_2(x, y, k-1) + \left(1 - \frac{K-1}{K}\right) \cdot I^2(x, y, k), \quad (1.5)$$

$$\sigma^2(x, y, k) = E_2(x, y, k) - E^2(x, y, k),$$

kur $I(x, y, k)$ - pikseļa intensitāte kadrā k ;
 $E(x, y, k)$ - pikseļa intensitātes matemātiskā cerība;
 $E_2(x, y, k)$ - pikseļa intensitātes otrās kārtas sākuma momenta

funkcija;

$\sigma^2(x, y, k)$ - pikseļa intensitātes dispersija.

Tad pikseļu šī brīža dispersijas vērtība tiek salīdzināta ar kadra dispersiju, ja $\sigma^2(x, y, k) \geq \sigma_{kadra}^2$, tad reģions tiek uzskatīts par dinamisku, ja $\sigma^2(x, y, k) < \sigma_{kadra}^2$, apgabals ir statisks.

Intensitātes dispersija tiek lietota, jo katram pikselim tā ir atkarīga gan no intensitātes izmaiņas amplitūdas, gan no izmaiņas ilguma, tādējādi tiek mazināta īslaicīga trokšņa ietekme uz lēmumu vai dotais pikselis ir iekļaujams statistiskajā vai dinamiskajā kadra apgabalā.

1.4. Automašīnu izsekošana kadrā

Lai izsekotu automašīnas pārvietojumu kadrā, nepietiek ar detektēto savienoto elementu analīzi. Detektēšanas posmā atrasto objektu pikseļiem var piešķirt tos identificējošas pikseļu vērtības, piemēram, pirmā objekta pikseļi var būt ar intensitāti viens, otrā - divi utt. Tomēr, jau nākamajā kadrā, kad automašīnu savstarpējais izvietojums var būt izmainījies, detektēšanas algoritms var piešķirt iepriekš otrajam objektam vieninieka vērtības utt.

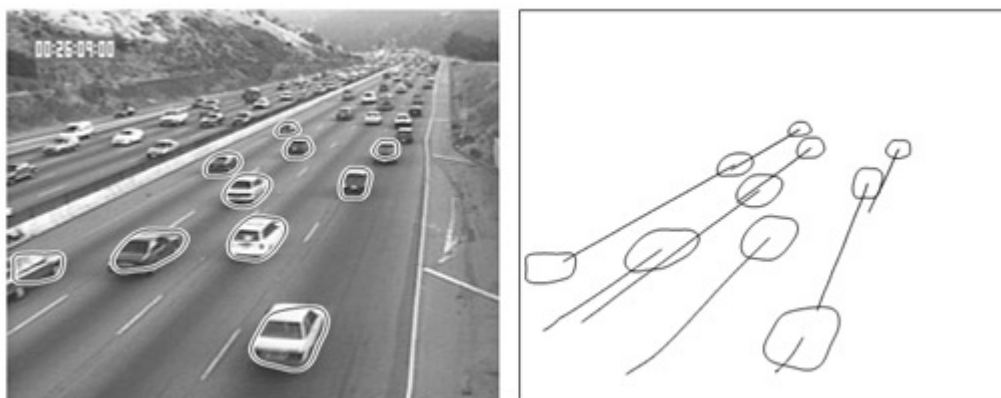
Turklāt, lai iegūtu pēc iespējas vairāk informācijas par satiksmi, kameru jānovieto tā, lai būtu saskatāms pēc iespējas garāks ceļa posms. Tādēļ kamera nevar būt vērsta perpendikulāri pret ceļu, ja vien tā nav uzstādīta lielā augstumā. Kamerai skatoties uz ceļu ar leņķi, rodas situācijas, kad tuvu braucošas automašīnas daļēji aizklāj skatam tālāk esošās, līdz ar ko vienkārša savienotu elementu noteikšana uzrādīs šajā apgabalā vienu objektu, pat ja dažus kadrus iepriekš abas mašīnas tika detektētas kā atsevišķi objekti.

Analizējot satiksmi pie krustojuma, šāda situācija ir bieža un neizbēgama. Ja kādā kamerai tālākā apgabalā izdodas detektēt atsevišķu automašīnu, tad informāciju

par šo transporta līdzekli iespējams izmantot tālākai mašīnas izsekošanai, lai atdalītu to no citām tuvumā esošām mašīnām. Piemēram, turpmākai izsekošanai iespējams izmantot sākumā atrastās automašīnas bildi, ko nākamajos kadros izmantot kā veidni jeb šablonu (*template matching*). Katrā sekojošā kadrā veidnei tiek piemeklēta pozīcija, kurā tai ir visvairāk kopīgu pikseļu ar kadru. Tā kā laikā mainās automašīnu izmērs un orientācija attiecībā pret kameru, veidnei jābūt adaptīvai – to var panākt, piemēram, ņemot par nākamajā kadrā meklējamo veidni, iepriekšējā kadrā atrasto veidnei atbilstošo kadra apgabalu.

Izsekot var arī nevis dažādas intensitātes pikseļu automašīnas bildi, bet kustīgu baltu apgabalu, kas atbilst mašīnai binārā bildē. Apgabala izsekošanā iespējams izmantot jau iepriekš zināmu informāciju par iespējamo kustību - pārvietošanās modeli, kas, piemēram, nosaka, ka, ja blakus kadros starp apgabaliem ir pārāk liels attālums, tad tā nevar būt tā pati automašīna.

Apgabala vietā mēdz arī iegūt un laikā izsekot interesējošā objekta kontūru. Atšķirībā no vesela reģiona izsekošanas, šī metode pieprasa mazāku skaitļošanas jaudu. Metodes trūkums ir tas pats, kas reģionu izdalīšanas metodei – viena otrai tuvu braucošas automašīnas var tikt izsekotas kā viens kontūrs. Atsevišķas automašīnas iespējams tomēr atklāt arī daļējas robežu pārklāšanās gadījumā, ja izdodas pirms tam katrai automašīnai piesaistīt konkrēto kontūru veidojošās robežas. Piemēram, rakstā [6], no kurienes arī attēls 1.4., katras automašīnas kontūrs tiek aprakstīts ar noslēgtām brīvformas līknēm jeb splainiem, sauktiem arī par čūskām, kuru kustība un forma tiek izsekotas laikā.



1.4.att. Automašīnu aprakstīšana ar splainiem un to izsekotais pārvietojums

Kontūru izsekošanu var uzlūkot kā speciālu gadījumu plašākam izsekošanas paņēmieni klāstam – raksturīgo parametru un pazīmju izsekošanai. Šajā pieejā video plūsmā netiek izsekoti veseli objekti (automašīnas), bet gan kādas to raksturīgas daļas – skaidri izdalāmi punkti, līnijas, stūri. Šīs metodes ļauj darboties arī objektu daļējas pārklāšanās apstākļos, ja kādas no raksturīgām pazīmēm paliek redzamas kadrā. Vēl viens metodes pluss ir tāds, ka izdalot raksturīgās pazīmes, tiek samazināts datu apjoms, kas apraksta automašīnu, un kas jāmeklē tālākas izsekošanas laikā.

Kad objekta pazīmes ir detektētas, tās var izsekot. Var izdalīt vairākas pieejas. Skaitliskās signatūras izsekošanā katram detektētajam objektam tiek izteikta uz intensitātes un ģeometrijas balstīta pazīmju kopa. Pazīmes tiek korelētas nākamajā kadrā, lai precizētu to jauno atrašanās vietu. Tad pazīmes tiek atjaunotas pielāgojot tās attāluma, perspektīvas, objekta redzamības izmaiņām.

Savukārt avotā [14], pēc fonam nepiederošu objektu detektēšanas, šo objektu raksturīgās iezīmes tiek salīdzinātas ar iepriekš izveidotu modeli, kuru veido no agrāk noteiktām, automašīnām piedēvētām iezīmēm. Tātad tiek izmantots uz modeļiem

balstīts princips, kurā iezīmes tiek izmantotas atklāto objektu klasificēšanas nolūkā, ļaujot atnest nepareizi detektētos objektus.

Gadījumos, kad atklātā objekta trajektorija norāda, ka objekts ir automašīna, bet modelī uzkrātās īpašības nosaka, ka tā nav, tad pazīmju modelis tiek automātiski papildināts ar jauniem automašīnu attēliem. Tādā gadījumā jauna objekta bilde tiek pārveidota par iezīmēm, kas turpmāk arī tiks izmantotas papildinātajā modelī.

Atrasto automašīnu izsekošanā, kā efektīva metode trokšņainu mērījumu datu un mašīnu daļējas savstarpējās aizklāšanās gadījumos, bieži tiek ieteikta Kalmana filtra lietošana. Piemēram, publikācijā [16] Kalmana filtrs tiek lietots, lai kadrā izsektu automašīnai atbilstošo objektu, avotā [17] ar Kalmana filtru tiek paredzētas automašīnu iezīmju (šajā gadījumā to stūru) koordinātes un ātrums nākamajā kadrā. Rakstā [18] dienas laikā filtrs tiek lietots, lai izsektu vējstikla centra pozīciju, un izmēru tā izmēru, savukārt naktī tiek izsekots viduspunkts starp lukturiem, kā arī attālums starp lukturiem. Avotā [19] ar Kalmana filtru, ņemot vērā objekta ierobežojošā četrstūra centrālā punkta trajektoriju un ātrumu, tiek paredzēta automašīnas trajektorija. Ņemot vērā paredzamo trajektoriju, tiek samazināts apgabals nākamajā kadrā, kurā automašīnu meklēt.

Bez izsekošanas, Kalmana filtrs tiek pielietots arī adaptīvas fona bildes ieguvei, kā to dara publikācijā [20].

1.5. Automašīnu detektēšana naktī

Projektā tiek apskatītas metodes satiksmes novērošanai ar redzamā spektra kamerām. Tādēļ nakts laikā, zemā apgaismojuma dēļ, mainās redzami automašīnu parametri, un tādas metodes kā automašīnas korpusa un robežu nodalīšana no ceļa kļūst problemātiska. Reālai detektēšanas sistēmai, kas paredzēta darbībai gan dienā, gan naktī, būtu jāspēj automātiski noteikt ceļa apgaismoju, un atkarībā no tā mainīt atklāšanas metodes. Šeit var analizēt kadra kopējo intensitāti, tomēr, blīvas satiksmes apstākļos, automašīnu lukturi var kopējo kadra intensitāti padarīt gaišu arī nakts laikā. Alternatīva būtu pētīt cik gaišs ir kāds kadra apgabals ārpus ceļa.

Patentā [21] tumšo apstākļu detektēšanas nolūkam tiek pētīts $m \times n$ pikseļu liels apgabals uz ceļa, un tiek aprēķināta gan šī apgabala vidējā intensitātes vērtība, gan šīs intensitātes dispersija:

$$\bar{I}_{m,n} = \frac{1}{m \cdot n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} I(x, y), \quad (1.6)$$

$$\sigma_{m,n} = \frac{1}{m \cdot n} \sum_{(x=0)}^{m-1} \sum_{y=0}^{n-1} (I_{m,n} - I(x, y))^2. \quad (1.7)$$

Lēmums par nakts iestāšanos tiek pieņemts tad, kad kopējā intensitāte $\bar{I}_{m,n}$ ir maza, vai arī liela ir tās dispersija. Nakts laikā uz ceļa ir tumši apgabali, kas kontrastējas ar lielas intensitātes lukturu gaismām, no kurienes arī rodas liela dispersija.

Kad ir noteikts, ka jāstrādā nakts apstākļos, var sākt automašīnu detektēšanu. Visuzskatāmāk nakts laikā par automašīnu esamību kadrā liecina priekšējo lukturu gaisma, kā arī tās atspīdums uz ceļa, tādēļ izplatīta metode satiksmes plūsmas novērtēšanai naktī ir lukturu atklāšana [14]. Automašīna var tikt definēta, kā lukturu pāris, kas atrodas viens no otra noteiktā attālumā un leņķī. Metodes trūkums ir nespēja identificēt skatam daļēji aizsegtus transporta līdzekļus, no kuriem kadrā ir saskatāms tikai viens lukturnis.

Lietojot divas kameras, iespējams automašīnas identificēt reizē gan pēc priekšējiem, gan aizmugurējiem lukturiem. Kad viena kamera detektē priekšējos

lukturus, un otra kamera tajā pašā laikā detektējusi attiecīgos aizmugurējos, mašīna tiek atklāta ar lielu ticamību, kā arī tiek iegūta papildus informācija par tās garumu un ātrumu. Tomēr arī šāda sistēma ir nenoturīga pret aizsegtu automašīnu radītām detektēšanas kļūdām, kā arī kameru skaita pieaugums un specifiskais izvietojums sarežģī sistēmas izvietojumu uz ceļa.

Naktīs tiek lietoti algoritmi, kas vispirms tiek galā ar lukturu atspīdumu uz ceļa, atstājot tikai lukturus, kā vienīgos izteiktos objektus kadrā. Pēc šādas apstrādes var lietot iepriekšējās nodaļās aprakstītos algoritmus raksturīgo parametru un kustīgu objektu atrašanai un izsekošanai, pielāgojot tos redzamo lukturu īpašībām.

1.6. Mūsu piedāvātā metode automašīnu atklāšanai

Apskatītajām metodēm piemīt dažādas priekšrocības un trūkumi. Mūsu piedāvātajā automašīnu atklāšanas algoritmā apvienotas vairākas priekšrocības. Pirmkārt, tiek apstrādāta tikai kadra daļa – ceļam perpendikulāra līnija, kas nozīmē, ka nav vajadzīgi lieli skaitļošanas resursi. Algoritms vispirms atrod automašīnas, izmantojot blakus kadru starpību – tāpat kustību detektēšanu. Lai risinātu problēmu, ka blakus kadru starpības gadījumā netiek detektētas apstājušās automašīnas, algoritmā iestrādāta automašīnu detektēšana arī ar fona uzkrāšanu, kas uzrauga mašīnas esamību uz pētāmās līnijas pēc tam, kad kustību detektēšana ir noteikusi, ka uz līnijas uzbraukusi automašīna.

Mūsu algoritms:

1. Ieejā pienāk video kadri. Atkarībā no video materiāla, tiek ievadītas konstantes. Vairākas nepieciešamās vērtības gala algoritmam būs jāievada uzstādot iekārtu – veicot kalibrēšanu (kadra izmērs, pētāmās līnijas koordinātes u.c.). Citas vērtības algoritms varētu noteikt pats – līdz ar to, tās būtu arī adaptīvas (slietņi, mašīnām atbilstošo intervālu vidējie garumi u.c.).

2. Kadrs tiek konvertēts no krāsaina uz melnbaltu, ko bez automašīnu skaitīšanas izmanto arī kameras noraustīšanās detektēšanai.

2.1. Melnbaltais kadrs tiek saglabāts 2 videokadru garumā.

2.2. Saglabātais kadrs tiek atņemts no šī brīža kadra.

2.3. Saglabātais kadrs tiek atņemts no šī brīža kadra.

2.4. Starpībai tiek ņemta absolūtā vērtība.

2.5. Tiek iegūta visu starpības pikseļu summa.

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

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

3.1. Līniju matrica tiek saglabāta viena vai vairāku sekojošu kadru garumā. Saglabāšanas ilguma koeficients varētu būt atkarīgs no video kadru skaita sekundē un mašīnu vidējā ātruma (no tā, cik kadru laikā mašīna pārbrauc pāri līnijai).

3.2. Saglabātās līnijas tiek atņemtas no šī brīža līniju vērtībām.

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

3.4. No rezultātā iegūtajām līnijām tiek izveidota viena līnija. Katram izejas līnijas pikselim piešķir maksimālo vērtību no atbilstošajiem L līniju pikseļiem.

4. No starpību līnijas tiek izgriezts pētāmais ceļam atbilstošais intervāls.

5. Interesējošam līnijas apgabalam tiek pielietots mediānas filtrs, lai samazinātu liekus pārrāvumus starp mašīnai atbilstošajiem intervāliem.
6. Līnija tiek sliekšņota. Tiek izmantotas divas sliekšņa vērtības $S11$ un $S12$, kur $S12 < S11$.
- 6.1. Šī brīža līnijai tiek pielietots lielais sliekšnis $S11$.
- 6.2. Iegūtajā binārajā līnijā tiek noteiktas visu balto pikseļu veidoto intervālu sākuma un beigu koordinātes.
- 6.3. Ja uz iepriekšējā kadra atbilstošās līnijas tiek atklāts intervāls, kam izpildās nosacījumi, ka tas ir lielāks par iepriekš definētu minimālo mašīnai atbilstošo intervālu, un mazāks par noteikto maksimālo vērtību, tad šī brīža kadrā intervāla apņemtajā apgabalā un tā apkārtnē (plus 10 pikseļi uz abām pusēm no intervāla gala koordinātēm) tiek lietots mazais sliekšnis $S12$.
- 6.4. Tiek pārbaudīts, vai starp kādiem diviem intervāliem reizē izpildās sekojošas sakarības: intervāli ir līdzīgi pēc garuma, intervālus savienojot, tie veidos intervālu, kas būs pietiekami liels automašīnai, bet ne pārāk liels.
- 6.5. Ja izpildās visas sakarības, tad abu intervālu apvienotajā intervālā (kā arī + 10 pikseļos uz katru pusi no izveidotā kopīgā intervāla) uz līnijas vēlreiz tiek pārskatīti pikseļi, šoreiz lietojot mazāku sliekšni $S12$.
7. Uz sliekšņotās līnijas tiek noteikti un saglabāti intervāli, kas ir garāki par iestādāmu lielumu.
8. Tālāk notiek iegūto intervālu apstrāde:
- 8.1. Ja šī brīža kadrā kāds intervāls daļēji pārklājas ar iepriekš saglabātajiem, tie tiek apvienoti kopējā intervālā. Tiek panākts, ka laikā intervāli tiek apvienoti. Piemēram, vairāki mazāki intervāli, kas sākotnēji atbilda atsevišķām vienas automašīnas detaļām, šādā veidā dod ieguldījumu nākamajos kadrus veidojot kopīgo automašīnas intervālu.
- 8.2. Ja kāds no iegūtajiem intervāliem atbilst punkta 6.3. prasībām, tas tiek uzskatīts par iespējamu mašīnai atbilstošo intervālu.
- 8.3. Ja kāds no intervāliem ir pārāk garš (pārsniedz vidējo automašīnām atbilstošo intervālu pareizinātu ar iestādāmu koeficientu), tas jāsadala sīkākos intervālos:
- 8.3.1. Garais intervāls tiek salīdzināts ar intervāliem, kas bija pirms garā intervāla parādīšanās.
- 8.3.2. Tas intervāls, kurš iepriekšējā kadrā bijis lielā intervāla robežās, pie tam, ir pārsniedzis minimālo mašīnai atbilstošo intervāla garumu, un atrodas vistuvāk kreisajai lielā intervāla malai, tiek ņemts par atskaites intervālu dalīšanai.
- 8.3.3. Atskaites intervāla gala koordināte (labajā galā) tiek ņemta par atdalīšanas koordināti, viena intervāla vietā tiek izveidoti divi.
- 9.1. Uz pētāmās līnijas paralēli blakus kadru starpības noteikšanai, tiek noteiktas arī robežas.
- 9.1. Līnijai tiek pielietots nogludinošais gausa filtrs.
- 9.2. Tiek iegūta katru blakus pikseļu starpība, šī starpība tagad 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 fona uzkrāšanas buferī 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 šobrīd eksistē mašīnas izmēram atbilstoši intervāli, šo intervālu atstāj nemainīgu robežu līnijā – pārējo, intervālam nepiederīgo apgabalu pielīdzina nullēm.
- 9.7. Robežu līniju saglabā 3 kadru garumā.
- 9.8. No izdalītās, 3 kadrus glabātās robežu līnijas atņem uzkrāto fona līniju.
- 9.9. Starpības pikseļiem tiek pielietots sliekšnis.
- 9.10. Tiek izskaitīts, cik pikseļu ir pārvarējuši sliekšni. Šie pikseļi tiek pierakstīti katram

intervālam, kā to raksturojošs parametrs.

9.11. Ja sliekšni pārvarējuši mazāk kā noteikts skaitsskaits pikseļu, tas nozīmē, ka intervāla apgabalā nav pietiekami daudz ceļam neatbilstošu robežu, un konkrētais intervāls iespējams jāslēdz.

10. Ja mašīna aizbraukusi – intervāls jāslēdz, slēgšana notiek ja reizē izpildās sekojoši nosacījumi:

10.1. Intervāla koordinātes nav paplašinājušās 3 kadru laikā.

10.2. Intervāla apgabalā šobrīd nav atrasti jauni zīmīga izmēra intervāli.

10.3. Intervālam pierakstītie robežu pikseļi (punkts 9.10) nepārsniedz noteiktu lielumu.

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

11.1. Intervāls nav nedz par platu, nedz par šauru, lai atbilstu automašīnai.

11.2. Intervāls ir pastāvējis vismaz 3 kadrus.

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

13. Tiek noskaidrots aptuvenš pikseļu skaitu, ko automašīna nobrauc pētāmās līnijas apkārtnē viena kadra laikā.

13.1. Intervālam, kas pēc izmēriem atbilst mašīnai, tiek saglabātas pikseļu vērtības.

13.2. Nākamajā kadrā aizturētais intervāls tiek salīdzināts ar tāda paša izmēra intervāliem, kas atrodas zem pētāmās līnijas.

13.3. Tiek noteikts, cik tālu (cik rindu uz leju) no pētāmās līnijas ir vislīdzīgākais intervāls. Attālums arī nosaka nobraukto pikseļu skaitu. Šī informācija norāda uz automašīnas ātrumu un ļauj detektēt apstāšanās gadījumus.

Piedāvātais algoritms var tikt papildināts, piemēram, līdz ar esošo robežu salīdzināšanu ar fona robežām, iestrādājot arī citu parametru (intensitāte, krāsa) fonu iegūšanu. Algoritmā paredzēta ne tikai adaptīva fona iegūšana (robežām un citiem parametriem), bet arī adaptīva sliekšņu vērtību iegūšana, kas nosaka, kuri pikseļi tad atbilst fonam, kuri atklātajam objektam.

Aprakstītais algoritms izmanto vienu līnijas detektoru kadrā (pētītas tiek tikai dažas blakus rindas kadrā). Šādu detektoru skaits var tikt palielināts, izvietojot kadrā vairākas šādas līnijas. Tas padarītu algoritmu precīzāku (apvienojot datus no dažādajām līnijām), kā arī ļautu ne tikai atklāt automašīnas kadrā, bet arī izsekot to kustībai.

2. Attēlu apstrādes metodes nummura zīmju detektēšanai

Atskaites periodā tika veikta iepriekš piedāvātās automobiļu valsts reģistrācijas numuru (AVRN) noteikšanas tehnoloģijas izstrāde, analīze un aprobācija.

Attēlu apstrādē tika izmantotas sekojošas metodes:

- attēla saspiešana ar speciālu masku pielietošanu;
- attēla binarizācija ar aprēķinātu binarizācijas sliekšni, kas tiek noteikts katram attēlam atsevišķi;
- speciālas attēla attīrīšanas metodes;
- attēlu apstrāde ar speciālām maskām, lai uzlabotu AVRN attēla kvalitāti;
- saistīto pikseļu kopumu meklēšana.

2.1 Attēla saspiešana ar speciālu masku pielietošanu

Pirmā etapā notiek attēla saspiešana ar filtrāciju. To izpilda, lai samazinātu turpmāko attēla analīzes laiku. Lai saglabātu mūs interesējošās attēla īpašības, tiek izmantots speciāls filtrs. Sakumā bilde ar mašīnu un AVRN izskatās sekojoši (2.1.att.):



2.1.attēls

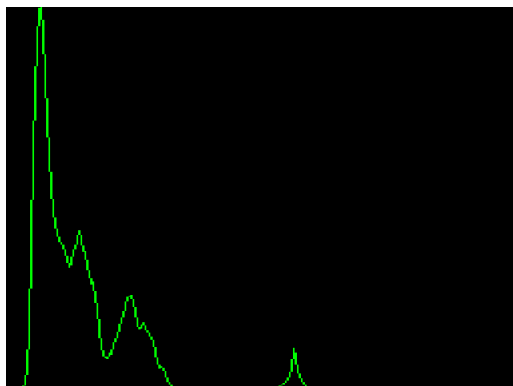
Attēla saspiešana notiek sekojoši: vispirms tiek noteikts saspiežamās zonas izmērs, kurš tiks nomainīts ar vienu pikseli, tad tiek analizētas secīgi visas attēla zonas. Attēla saspiešanas rezultāts parādīts 2.2.att.



2.2.attēls

2.2 Attēla binarizācija ar aprēķinātu binarizācijas sliekšni

Lai turpinātu AVRN plātes meklēšanu nepieciešams noteikt attēla binarizācijas sliekšni. Analizējot attēlu histogrammas (2.3.att.) secināts, ka binarizācijas sliekšnis atkarīgs ne tikai no vidējās pikseļu intensitātes, bet arī no dispersijas.



2.3.attēls

Ja ar S apzīmē binarizācijas sliekšni, tad

$$S = F(M,D),$$

kur M - attēla pikseļu vidēja intensitāte, bet D - dispersija.

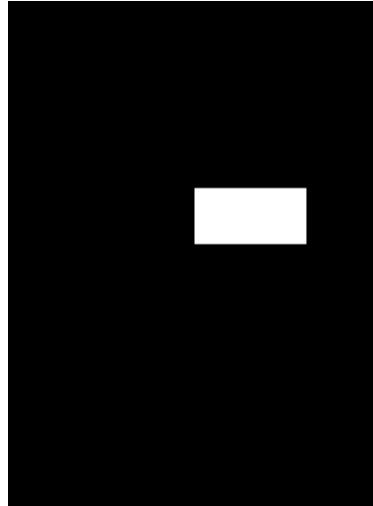
Izmantojot aprēķināto binarizācijas sliekšni, izpildām attēla binarizāciju. Rezultātā iegūstam melnbaltu attēlu (bez pelēkiem toņiem), kurā uz melna fona izvietotas viena vai parasti vairākas baltas zonas (2.4.att.). Viena no tām atbilst numura platei, pārējās – artefakti.



2.4.Attēls

2.3 Speciālas attēla attīrīšanas metodes

Artefaktu 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 (2.5.att.).



2.5.Attēls

Ja paliks vairākas, zonas tad visi nākošie soļi jāpiemēro katrai atrastajai zonai.

2.4 Attēlu apstrāde ar speciālām maskām, lai uzlabotu AVRN attēla kvalitāti

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

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

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

2.5 Saistīto pikseļu kopumu meklēšana

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

Ja $R_{min} < R_c \leq R_{max}$, tad izdarām secinājumu, ka tas ir simbols. Pretējā gadījumā pieņemam, ka tas ir artefakts un attiecīgais pikseļu kopums tiek pārkrāsots baltā krāsā (fona krāsā). Kā R_{min} tiek izmantots pikseļu skaits attēlā „1”, bet kā R_{max} – simbola „W” pikseļu skaits. Pēdējā etapā tiek startēta OCR apakšprogramma, kurai, kā ieejas dati, tiek secīgi padoti visi atrastie 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.

3. Literatūras saraksts

- 1.Kastrinaki V., Zervakis M., Kalaitzakis. A survey of video processing techniques for traffic applications// Image and Vision Computing. – 2003. – Nr.21. – 359.-381. p.
- 2.Leotta M. J., Mundy J. L. Vehicle Surveillance with a Generic, Adaptive, 3D Vehicle Model// IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2011. – Nr.7. – 1457.-1469. p.
- 3.Houben Q., Diaz J. C. T., Warzée N., Debeir O., Czyz J. Multi-feature Stereo Vision System for Road Traffic Analysis// Proceedings of the Fourth International Conference on Computer Vision Theory and Applications. – 2009. – 129.-134. p.
- 4.Atkočiūnas E., Blake R., Juzapavičius A., Kaziminianec M. Image Processing in Road Traffic Analysis// Nonlinear Analysis: Modelling and Control. – 2005. – Nr.4. – 315.-332. p.
- 5.Bertozzi M., Broggi A. GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection// IEEE Transactions on Image Processing. – 1998. – Nr.1. – 62.-81. p.
- 6.Koller D., Weber J., Malik J. Robust Multiple Car Tracking with Occlusion Reasoning// Lecture Notes in Computer Science. – 1993. – Nr.800. – 189.-196. p.
- 7.Li X., Liu Z.-Q., Leung K.-M. Detection of vehicles from traffic scenes using fuzzy integrals// Pattern Recognition. – 2002. – Nr.35. – 967.-980. p.
- 8.Jia Y., Zhang C. Front-view vehicle detection by Markov chain Monte Carlo method// Pattern Recognition. – 2009. – Nr.42. – 313.-321. p.
- 9.US 5,296,852 (ASV). Method and apparatus for monitoring traffic flow/ Rathi R. P. – Iesniegts 27.02.1991. Nr. 661,297; Publicēts 22.03.1994.
- 10.Jähne B. Digital Image Processing. 6th revised and extended edition. – Berlin: Springer, 2005. – 639 p.
- 11.Hu W., Tan T., Wang L., Maybank S. A Survey on Visual Surveillance of Object Motion and Behaviors// IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. – 2004. – Nr.3. – 334.-352. p.
- 12.Gupte S., Masoud O., Martin R. F. K., Papanikolopoulos N. P. Detection and Classification of Vehicles// IEEE Transactions on Intelligent Transportation Systems. – 2002. – Nr.1. – 37.-47. p.
- 13.Rodríguez T., García N. An adaptive, real-time, traffic monitoring system// Machine Vision and Applications. – 2010. – Nr.21. – 555.-576. p.
- 14.Thi H. T. A robust traffic surveillance system for detecting and tracking vehicles at nighttime. – Sydney: University of Technology, 2007. – 100 p.
- 15.Rabie T., Auda G., El-Rabbany A., Shalaby A., Abdulhai B. Active-Vision-based Traffic Surveillance and Control// Proceedings of the Fourteenth Canadian Vision Interface Conference. – 2001. – 87.-93. p.
- 16.Meffert B., Blaschek R., Knauer U., Reulke R., Winkler F., Schischmanow A. Monitoring traffic by optical sensors// Proc. of 2nd International Conference on Intelligent Computing and Information Systems. – 2005. – 9.-14. p.
- 17.Coifman B., Beymer D., McLauchlan P., Malik J. A real-time computer vision system for vehicle tracking and traffic surveillance// Transportation Research Part C. – 1998. – Nr.6. – 271.-288. p.
- 18.Robert K. Video-based traffic monitoring at day and night: vehicle features detection tracking// Intelligent Transportation Systems. – 2009. – 1.-6. p.
- 19.Rad R., Jamzad M. Real time classification and tracking of multiple vehicles in highways// Pattern Recognition Letters. – 2005. – Nr.26. – 1597.-1607. p.

20. Cheung S. S., Kamath C. Robust Background Subtraction With Foreground Validation For Urban Traffic Video// EURASIP Journal of Applied Signal Processing. – 2005. – Nr.14. – 2330.-2340. p.

21. US 7,460,691 B2 (ASV). Image processing techniques for a video based traffic monitoring system and methods therefor/ Ng Y. L., Ang K. S., Chong C. C., Gu M. K. – Iesniegts 15.08.2006. Nr. 11/504,276; Publicēts 02.12.2008.