

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**VIRTUĀLĀ REALITĀTE:
CAURSPĪDĪGI OBJEKTI**

MAGISTRA DARBS

Autors: **Jānis Judvaitis**

Stud. apl. Nr. jj08005

Darba vadītājs: profesors Dr. dat. Leo Seļāvo

RĪGA 2014

ANOTĀCIJA

Mūsdienās virtuālā realitāte tiek izmantota arvien biežāk, viens no virzieniem, kas daudziem varētu likties interesants ir kāda reāla objekta caurspīdīguma nodrošināšana virtuālajā realitātē. Piemēram, ja pa ceļu viena aiz otras pārvietojas divas automašīnas, otrās automašīnas vadītājam varētu būt lietderīgi redzēt „cauri” priekšā braucošajai automašīnai, lai iegūtu labāku priekšstatu par notiekošo uz ceļa. Protams, neko tādu nav iespējams nodrošināt, ja nav pieejams attēls, kas attēlotu to, kas objektam, kuru gribam padarīt caurspīdīgu, atrodas otrā pusē. Šī kursa darba mērķis ir izpētīt esošos risinājumus šāda caurspīdīguma nodrošināšanai un izveidot savu risinājumu, kurš būtu piemēros augstāk aprakstītajam scenārijam, t.i. spētu darboties reālā laikā un sniegt saprotamu priekšstatu par to, kas notiek caurspīdīgajam objektam otrā pusē.

Atslēgvārdi: Virtuālā realitāte, caurspīdīgi reāli objekti, caurspīdīga automašīna.

ABSTRACT

Nowadays virtual reality is used more often, one of directions of usage, which many could find interesting is transparency of some real object in virtual reality. For example, if on the road two cars drive in the same direction one after another, then the driver of second car could use ability to see „through” the car which is driving in the front of him, to get better understanding about what is happening on the road ahead. Of course none of this is possible is there are no images describing what is located on the other side of the object which we want to make transparent. The target of this course work is to analyze existing solutions to provide such transparency as well as create a new solution which would be suitable for earlier described scenario, e.g. could work in real time and give understandable insight about what is happening on the other side of transparent object.

Key words: virtual reality, transparent real objects, transparent car.

AUTOREFERĀTS

Autors maģistra darba ietvaros ir veicis literatūras izpēti par attēlu ar caurspīdīgiem objektiem iegūšanu un iepazīties attēlu apstrādes pamatiem. Literatūras izpētes nodālā tikai viena no tēmām ir balstīta uz nerecenzētu interneta avotu, tāpēc, ka informācija par konkrēto tēmu nav pieejama zinātniskajos rakstos vai žurnālos.

Darbā izmantotās metodes autors ir aprakstījis izklāstot to būtību, kā arī sniedzot atsauces zinātniskām publikācijām vai materiāliem internetā, kur padziļināti aprakstītas konkrētās metodes.

Darba pamatā ir divi autora veidoti prototipi, to apjoms koda rindiņās ir salīdzinoši neliels, bet ieguldītais darbs, lai tos izveidotu pēc autora domām sastāda aptuveni pusi no šī maģistra darbā paveiktā, jo autoram pirms maģistra darba izstrādes sākuma bija ļoti vispārinātas zināšanas par šo tēmu un to visu nācās apgūt maģistra darba ietvaros, kā arī izveidotos prototipus nācās pielāgot atsevišķiem eksperimentiem.

Autors uzskata, ka maģistra darba noformējums atbilst maģistra darba izstrādes un aizstāvēšanas metodiskajiem norādījumiem. Autors darbu ir pārlasījis vairākas reizes, lai pārliecinātos, ka darbā nav ievades kļūdu, kā arī autors darbu vairākas dienas pirms nodošanas gala versijā ir nodevis darba vadītājam caurskatīšanai.

Autors uzskata, ka maģistra darbā nepastāv plaģiātisma iespēja, jo autors pēc vislabākās sirdsapziņas visus izmantotos avotus ir pieminējis literatūras sarakstā ar atsaucēm tekstā, kur tie izmantoti. Visiem attēliem, kuri nav autora veidoti to nosaukumā ir atsauce uz literatūras avotu no kura paņemts konkrētais attēls.

SATURS

| | |
|--|----|
| Ievads..... | 7 |
| 1. Literatūras analīze..... | 9 |
| 1.1. Roku un rīku caurspīdīguma nodrošināšana..... | 9 |
| 1.2. Divpusējas skārienjūtīgas iekārtas..... | 11 |
| 1.3. Caurspīdīgais motora pārsegs..... | 12 |
| 1.4. Caur-redzes sistēma..... | 13 |
| 2. Pirmais prototips..... | 15 |
| 2.1. OpenCV attēlu apstrādes bibliotēka..... | 15 |
| 2.2. Metodes..... | 15 |
| 2.2.1. Kontūru noteikšanas metode..... | 16 |
| 2.2.2. Objektu atpazīšanas metode..... | 18 |
| 2.2.3. Attēlu sapludināšanas metode..... | 19 |
| 2.2.4. Secinājumi par metodēm..... | 25 |
| 2.3. Eksperimenti un novērtējums..... | 26 |
| 2.3.1. Pirmā prototipa pārbaudei izmantotie attēli..... | 26 |
| 2.3.2. Vairāku secīgu attēlu sapludināšana..... | 27 |
| 2.4. Secinājumi par pirmo prototipu..... | 28 |
| 3. Otrais Prototips..... | 29 |
| 3.1. Kvalitatīvu ieejas datu iegūšana..... | 29 |
| 3.1.1. Testa datu iegūšana no trīs automašīnām..... | 29 |
| 3.1.2. Testa datu iegūšana no imitētām automašīnām..... | 30 |
| 3.1.3. Testa datu iegūšana no vienādiem viedtālruniem..... | 31 |
| 3.2. Implementācijas uzlabošana..... | 32 |
| 3.2.1. Ātrdarbības novērtējums..... | 32 |
| 3.2.2. Ātrdarbības uzlabošana..... | 35 |
| 3.2.3. Ātrdarbības novērtējums ar ORB metodi..... | 37 |

| | |
|---|----|
| 3.3. Pareizo attēlu atrašana priekš sapludināšanas | 38 |
| 3.3.1. Pareizo attēlu priekš sapludināšanas īpašības..... | 39 |
| 3.3.2. Video ar GPS koordinātām..... | 44 |
| 3.4. Automātiska sapludinātā attēla korektuma pārbaude | 45 |
| 3.4.1. Aizvietotā laukuma forma | 46 |
| 3.4.2. Aizvietotā laukuma izmērs | 47 |
| 3.4.3. Histogrammas | 47 |
| 3.4.4. Sakrītošo punktu skaits..... | 48 |
| 3.4.5. Korektuma pārbaudes algoritms..... | 48 |
| 3.5. Eksperimenti un novērtējums | 49 |
| 3.5.1. Eksperiments ar attēliem no trīs pēc kārtas braucošām automašīnām..... | 49 |
| 3.5.2. Attēlu sapludināšana izmantojot attēlus no vienādām ierīcēm..... | 50 |
| 3.5.3. Video ar caurspīdīgu automašīnu iegūšana | 53 |
| 3.6. Diskusija | 55 |
| 3.7. Secinājumi par otro prototipu | 56 |
| 4. Nobeigums un secinājumi | 58 |
| 4.1. Pirmā prototipa rezultātu apskats | 58 |
| 4.2. Otrā prototipa rezultātu apskats | 59 |
| 4.3. Kopsavilkums | 59 |
| 5. Pateicības..... | 61 |
| 6. Izmantotā literatūra un avoti..... | 62 |

IEVADS

Mūsdienās virtuālā realitāte ar vien manāmāk ienāk mūsu ikdienas dzīvēs sākot ar sociālajiem tīkliem[1] līdz pat ikdienā lietojamiem apģērbiem un aksesuāriem, kā piemēram Google Glass[2] u.c.[3]. Autora skatījumā viens no virtuālās realitātes labākajiem pielietojumiem būtu papildus informācijas sniegšana par apkārtējo vidi cilvēkam draudzīgā un viegli uztveramā veidā.

Par pamatu virtuālās vides un caurspīdīgu objektu pētīšanai autors ir izvēlējis sekojošu scenāriju: pa ceļu vienā virzienā viena aiz otras pārvietojas divas vai vairāk automašīnas, kas aprīkotas ar attēlu iegūšanas iekārtām. Izmantojot virtuālās realitātes tehnoloģiju iespējas autors vēlas aizmugurē braucošās automašīnas vadītājam sniegt informāciju par to ceļa posmu, kurš atrodas aiz priekšā braucošās automašīnas un nav redzams aizmugurē braucošās automašīnas vadītājam.

Darba autors par galveno mērķi uzstāda virtuālās realitātes efekta iegūšanu reālā laikā padarot priekšā braucošo automašīnu caurspīdīgu, tādējādi paplašinot aizmugurē braucošās automašīnas vadītājam pieejamo informāciju par satiksmes stāvokli priekšā un cerams, uzlabojot ceļu satiksmes drošību.

Darba galvenais uzdevums ir algoritma izstrāde caurspīdīga objekta iegūšanai, kā arī strādājoša prototipa izveide, kas ļautu iegūt attēlus ar caurspīdīgu automašīnu laboratorijas apstākļos. Autors uzskata, ka šī maģistra darba ietvaros labs rezultātus būtu, ja izveidotais algoritms spētu apstrādāt vismaz 1 kadru sekundē, tādējādi ik sekundi atjaunojot informāciju par to, kas atrodas caurspīdīgajam objektam otrā pusē.

Darba gaitā autors sniedz atbildes uz šādiem jautājumiem:

- Kādas pieejas līdz šim izmantotas reālu objektu caurspīdīguma iegūšanai?
- Kā padarīt reālus objektus caurspīdīgus virtuālajā realitātē?
- Kādi attēli ir jāizmanto lai objektus padarītu caurspīdīgus?

Darbā izmantotās metodes:

- Literatūras analīze;

Darbs strukturēts sekojoši: 1. nodaļā autors veic literatūras apskatu analizējot citu autoru veiktos pētījumus par caurspīdīgu objektu iegūšanu. 2. nodaļā autors apraksta pirmā prototipa izveidi, kas sevī ietver vairāku metožu izvērtējamu caurspīdīga objekta iegūšanai, kā

arī sniedz secinājumus par pirmā prototipa izstrādes laikā sasniegtajiem rezultātiem. 3. nodaļā autors apraksta otrā prototipa izveidi, kas balstās uz pirmā prototipa izveides laikā gūtajām atziņām, kā arī autors apraksta vairākus eksperimentus, kas veikti ar mērķi uzlabot izstrādātā prototipa veiktspēju un iegūto attēlu kvalitāti. 4. nodaļā autors sniedz darba kopējo apskatu un konstatē šī maģistra darba laikā sasniegtos rezultātus.

1. LITERATŪRAS ANALĪZE

Virtuālās realitātes kontekstā caurspīdīgi objekti nav nekas jauns, bet parasti šie caurspīdīgie objekti nāk no virtuālās pasaules nevis reālās pasaules. Piemēram pētījumā [4] rakstīts par to, kā padarīt daļu no objekta caurspīdīgu vairāku lietotāju virtuālā vidē, lai visi lietotāji spētu sekot līdzī kādā procesam katrs no sava skatu punkta. Padarīt caurspīdīgus virtuālās pasaules objektus ir diezgan vienkārši, jo objekts jau pēc savas būtības ir uzzīmēts, tātad ir pieejami visi dati par to un izmantojot virtuālas 3D pasaules priekšrocības ir iespējams no jebkura skatu punkta uzzināt kas tad tieši atrodas aiz šī objekta. Bet, ja mēs vēlamies padarīt caurspīdīgu kādu reālās pasaules objektu, tad arī informācija par to, kas atrodas objektam otrā pusē ir jāiegūst no reālās pasaules.

1.1. Roku un rīku caurspīdīguma nodrošināšana

[5] autori ir mēģinājuši padarīt caurspīdīgas cilvēku rokas vai izmantotos rīkus, lai atvieglotu darbu veikšanu un sniegtu priekšstatu par to, ko rokas vai rīki aizsedz, kas ir īpaši svarīgi smalkos un precīzos darbos, kā piemēram operāciju gadījumā, kad ķirurga rokas vai kāds rīks aizsedz skatu. Šis pētījums un arī rezultāts diezgan lielā mērā sakrīt ar autora iecerēm, bet [5] autori izmanto tikai vienu kameru, tātad no tās ir jāiegūst informācija gan par rokām vai rīkiem, ko vajag padarīt caurspīdīgus, gan par objektiem, kas atrodas aiz rokām vai rīkiem.

Piedāvātais risinājums ir sekojošs: pirms darba uzsākšanas darba vieta tiek nofotografēta un darba laikā šī sākotnējā bilde tiek apvienota ar tekošo bildi izmantojot dažādus sapludināšanas līmeņus, kas arī sniedz roku vai rīku caurspīdīguma efektu. 1.1. att. parāda šīs metodes darbību, (a) gadījums ir sākotnējā bilde, (f) gadījums ir tekošā bilde un (b), (c), (d) un (e) ir dažādi starprezultāti, kas iegūti sākotnējai un tekošajai bildei pielietojot dažādus sapludināšanas līmeņus.



(a) Alpha level 0



(b) Alpha level 0.4



(c) Alpha level 0.5



(d) Alpha level 0.6



(e) Alpha level 0.8



(f) Alpha level 1

1.1. att. Objekta paņemšana ar dažādiem rokas caurspīdīguma līmeņiem, attēls no avota [5]

Tā kā caurspīdīgums tiek nodrošināts objektiem, ar ko cilvēkiem reāli nepieciešams mijiedarboties, tad ir dažas situācijas, kad šāds caurspīdīgums traucē un neļauj veikt precīzas darbības, kā piemēram, ja nepieciešams novietot lego klucīti uz cita lego klucīša, kā redzams 1.2. att. [5] autori to risina attiecīgo rīku atzīmējot ar speciālu iezīmi, lai to darba gaitā varētu atpazīt, kā arī noteikt tā pozīciju un pozu. Katram šādam rīkam ar iezīmi apakšējo malu padarot mazāk caurspīdīgu tas palīdzētu precīzākai mijiedarbībai starp šo rīku un fonu. 1.2. att. (a) redzams lego klucītis ar speciālo marķieri, (b) redzams kā izskatās lego klucītis kurš padarīts caurspīdīgs neņemot vērā marķieri un (c) redzams kā izskatās lego klucītis, pēc tam, kad tā apakšējā mala padarīta mazāk caurspīdīga.



(a) Lego brick and hand are rendered opaquely. An AR Toolkit marker is attached to the brick so that its position and pose can be tracked.



(b) Lego brick and hand are rendered at a uniform alpha value of 0.6. At this level, it is difficult for the user to determine the exact position of the brick relative to the target brick.



(c) The bottom edge of the brick is rendered with an alpha value of 0.9. Now it is much easier for the user to place the brick.

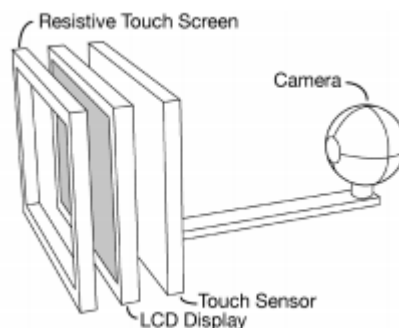
1.2. att. Svarīgo sastāvdaļu atzīmēšana un kontūras caurspīdīguma līmeņa mainīšana, attēls no avota [5]

Jāatzīmē, ka izmantotajai metodei ir divi ļoti lieli trūkumi. Abi šie trūkumi izriet no tā, ka tiek izmantota tikai viena kamera, tāpēc kameras novietojumu darbības laikā mainīt nav

iespējams, jo tad nepieciešams veidot jaunu sākuma attēlu, kā arī tas, kas redzams aiz caurspīdīgā objekta darbības laikā nevar tikt mainīts, jo šī viena kamera neredz to, kas notiek aiz rokas vai rīka, ir pieejama tikai informācijas par to, kas darba sākumā atradās aiz rokas vai rīka. Kā arī piedāvātā caurspīdīguma samazināšanas metode pēc autora domām nesniedz pietiekoši labu priekšstatu par objekta stāvokli un nepadara darbības starp objektu un fonu pietiekoši vienkāršas, labāks risinājums būtu bijis šī rīka kontūras apvilksana ar šauru, bet labi redzamu līniju. [5] autori gan apgalvo, ka turpinās darbu pie šīs tēmas un mēģinās abus augstāk minētos ierobežojumus apiet, bet diemžēl publikācijas par šādu tēmu no viņiem vairs nav bijušas.

1.2. Divpusējas skārienjūtīgas iekārtas

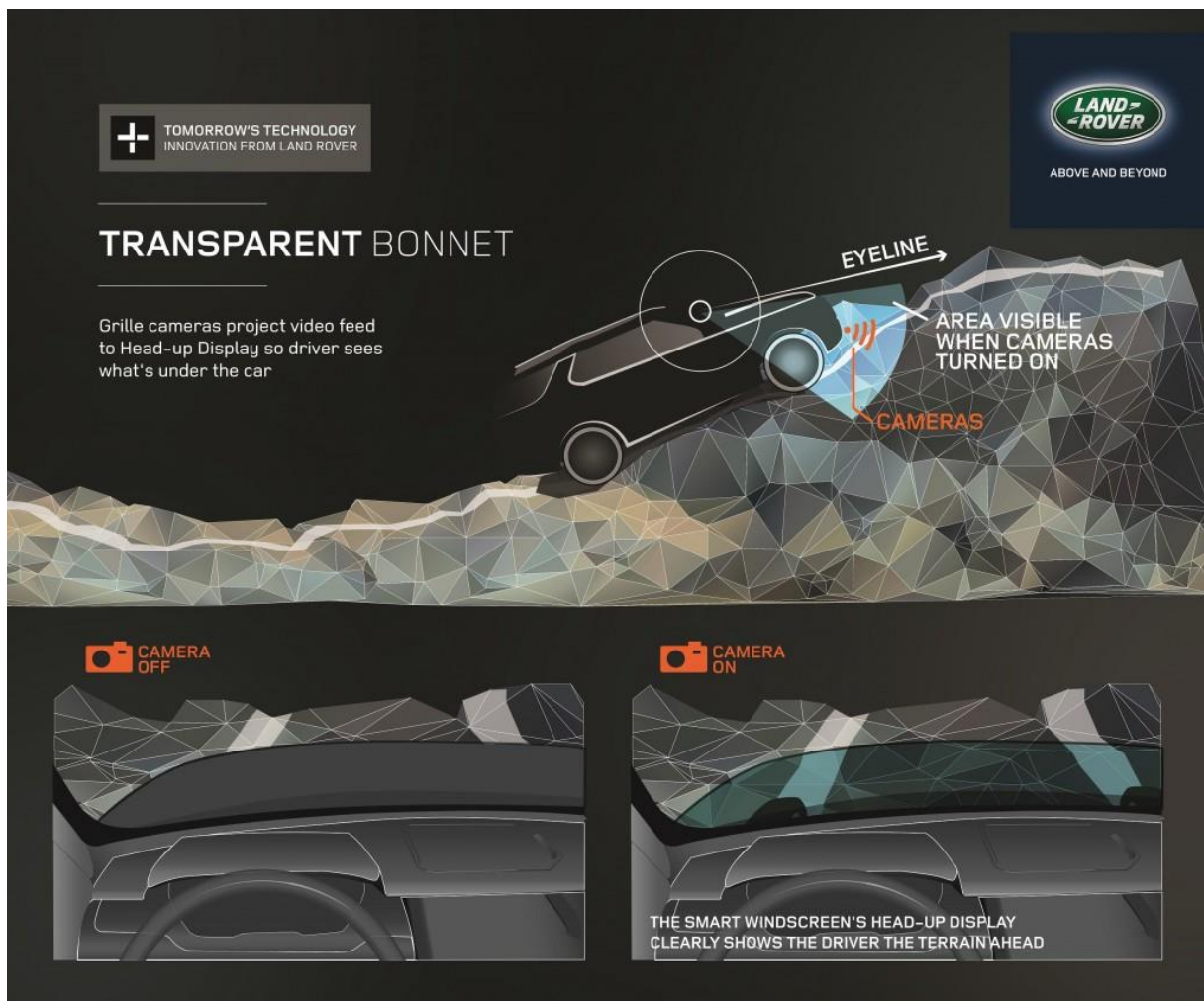
[6] autori piedāvā skārienjūtīgu iekārtu, kura piedāvā lietotājiem skārienjūtīgu virsmu abās ierīces ekrāna pusēs. Lai lietotāji redzētu savus pirkstus ekrāna otrā pusē tiek izmantota kamera un algoritms, kas šos pirkstus, kas atrodas otrā pusē attēlo uz ekrāna, tādējādi padarot pašu ekrānu daļēji caurspīdīgu. Šajā gadījumā ideja ir ļoti līdzīga autora iecerei, padarīt kādu objektu caurspīdīgu, bet izmantota tiek tikai viena kamera, jo attēlu kurā redzami caurspīdīgie objekti attēlo skārienjūtīgās ierīces ekrāns un šis attēls ir ņemts no virtuālās pasaules, attiecīgi uzzīmēts. Kā redzams 1.3. att. sākotnējā prototipā tiek izmantota kamera, kas novietota pret ekrāna otru pusi un kalpo kā informācijas devējs par to, kur atrodas lietotāja pirksti, tālākā realizācijā [6] autori plāno izstrādāt līdzīgu iekārtu, kas iztiks bez kameras un lietotāju pirkstu pozīcijas noteiks izmantojot speciālu sensoru, kas uzliks pa virsu skārienjūtīgajam sensoram ekrāna otrā pusē.



1.3. att. Divpusējas skārienjūtīgas iekārtas prototipa shēma, attēls no avota [6]

1.3. Caurspīdīgais motora pārsegs

Avotā [7] minēts, ka Land Rover plāno 2014. gada Ņūjorkas starptautiskajā auto izstādē prezentēt inovatīvu tehnoloģiju paplašinātās realitātes integrēšanai automašīnās. Ideja ir sekojoša, zem automašīnas priekšējās daļas ir novietotas vairākas kameras, kas vēro virsmu zem automašīnas, iegūtie attēli tiek attēloti uz automašīnas priekšējā vējstikla tā, lai tie pārklātu automašīnas motora pārsegu, tādējādi radot vadītājam iespaidu, ka automašīnas motora pārsegs ir caurspīdīgs. Papildus tam uz kameru iegūtā attēla tiek uzņemti virtuāli automašīnas priekšējie riteņi, kas attēlo pašreizējo automašīnas riteņu pagrieziena leņķi, tādējādi sniedzot vadītājam vēl vairāk informācijas par tekošo situāciju. 1.4. att. redzams vizuāls attēlojums par caurspīdīgā motora pārsega tehnoloģiju.



1.4. att. Caurspīdīgā motora pārsega vizuāls attēlojums, attēls no avota[7]

Protams nekādas detaļas par tehnisko realizāciju nav atklātas, kā arī nav skaidrs vai attēls uz priekšējā vējstikla tiek veidots ņemot vērā tekošo vadītāja acu novietojumu vai arī ir konstants, jo pēc autora domām skatoties no cita skatupunkta iegūtais attēls neradītu iespaidu

par caurspīdīgu motora pārsegu un iespējams varētu pat traucēt koncentrēties. Tomēr šis risinājums lielā mērā atbilst tam, ko autors mēģina darīt šajā maģistra darbā, jo reālās pasaules objekts – motora pārsegs, tiek padarīts caurspīdīgs un tas tiek panākts izmantojot kameras, kas novietotas šim objektam otrā pusē. Kā vēl vienu svarīgu atziņu autors vēlas uzsvērt šīs problēmas aktualitāti, jo pašlaik masu ražošanā šādas Land Rover automašīnas vēl nav un Land Rover izstrādātais caurspīdīgais motora pārsegs pašlaik ir tikai prototips.

1.4. Caur-redzes sistēma

[8] autori ir veidojuši algoritmu, izveidojuši prototipu un veikuši praktisku eksperimentu par tēmu caur-redzes sistēma. Galvenā ideja ir izmantot automašīna-automāšīna saziņas kanālu video straumēšanai un iegūto video rādīt aizmugurē braucošās automašīnas vadītājam izmantojot caurspīdīgu ekrānu, kas uzstādīts pie automašīnas priekšējā vējstikla. Uz šī ekrāna vietā, kur atrodas priekšā braucošā automašīna tiek rādīts saņemtais video, tādējādi radot iespaidu, ka priekšā braucošajai automašīnai ir iespējams redzēt cauri.

Lai atpazītu priekšā braucošo automašīnu tiek izmantots liels specifiskas krāsas marķieris, kas tiek uzstādīts aizmugurē priekšā braucošajai automašīnai. Atrodot šo marķieri aizmugurējās automašīnas uzņemtā video kadros tiek noteikta priekšā braucošās automašīnas atrašanās vieta. Vietā kur atrodas priekšā braucošā automašīna uz caurspīdīgā ekrāna tiek veidots virtuāls tunelis, kura izmēri tiek aprēķināti izmantojot priekšā braucošās automašīnas izmērus, tie tiek iegūti izmantojot automašīna-automāšīna saziņas kanālu. Izveidotā tuneļa “otrā galā” tiek rādīts saņemtais video, nepieciešamības gadījumā tas tiek apgriezts, lai iekļautos tuneļa izmēros, 1.5. att. redzams [8] autoru izstrādātais prototips darbībā.



1.5. att. Caur-redzes sistēmas prototips darbībā, attēls no avota [8]

[8] autori apgalvo, ka video, kas tiek attēlots uz priekšā braucošās automašīnas aizmugurē braucošās automašīnas vadītājam tiek rādīts ar aptuveni 200ms aizturi, kas ir ļoti labs rezultāts, jo pat pie ātruma 180km/h šī aizture rada tikai 10m nobīdi, t.i. tiek rādīts video it kā priekšā braucošā automašīna atrastos 10m uz aizmuguri no tās pašreizējās atrašanās vietas.

[8] autori arī uzsver, ka šis prototips pašreizējā stāvoklī var saturēt aklo zonu, kurā atrodas pretī braucošā automašīna un tā nav redzama ne saņemtajā video, ne no aizmugurē braucošās automašīnas vadītāja skata punkta. Vēl viens šī prototipa ierobežojums ir caurspīdīgā ekrāna nespēja bloķēt caurspīdīguma efektu nepieciešamajās vietās, rezultātā, kā redzams 1.5. att., attēlotais video ir ar izkropļotām krāsām, piemēram redzams rozā iekrāsots video kadra fragments, kur atrodas automašīnas atpazīšanai izmantotais marķieris. Autors uzskata, ka šis prototips ir ļoti labs sasniegums caurspīdīguma efekta iegūšanai, kā arī tas izmanto autora paredzēto mērķa vides izvietojumu, ar divām automašīnām, kas pārvietojas pa ceļu viena aiz otras un uzņem attēlus ar priekšā notiekošo. Tomēr, autors uzskata, ka ir iespējams iegūt labāku caurspīdīguma efektu, padarot caurspīdīgu visu priekšā braucošo automašīnu nevis tikai daļu no tās.

2. PIRMAIS PROTOTIPS

Autora izvēlētajā scenārijā pa ceļu vienā virzienā viena aiz otras pārvietojas divas automašīnas, kas aprīkotas ar attēlu iegūšanas iekārtām un datu pārraides iekārtām, lai iegūtos attēlus spētu nosūtīt uz serveri un apstrādātos attēlus saņemt no servera. Šādam mērķim ļoti labi kalpo viedtālrunis, kurš piedāvā gan kameru attēlu uzņemšanai, gan mobilā interneta savienojumu datu pārsūtīšanai starp viedtālruni un serveri.

Datu apstrāde uz servera izvēlēta tāpēc, lai risinājums nebūtu specifisks kādai no viedtālrunu platformām un arī tāpēc, ka serveris piedāvā lielāku skaitļošanas jaudu, kas attēlu apstrādes gadījumā, īpaši veidojot prototipus, ir ļoti svarīgi.

Darba gaitā autors izmēģināja vairākas pieejas caurspīdīga objekta iegūšanai, tika pārbaudīta iespēja padarīt automašīnu caurspīdīgu to atpazīstot, lai atpazītu automašīnu tika izmantotas kontūru noteikšanas metodes un objektu atpazīšanas metodes, kā arī tika pārbaudīta metode, kas spēj padarīt automašīnu caurspīdīgu nenosakot precīzu tās atrašanās vietu vai kontūru, tam tika izmantota attēlu sapludināšanas metode.

2.1. OpenCV attēlu apstrādes bibliotēka

Lai visas attēlu apstrādes funkcijas nebūtu jāimplementē pašam, autors izvēles lietot populāru attēlu apstrādes bibliotēku OpenCV[9], kurai pieejamas implementācijas vairākās populārās programmēšanas valodās, kā piemēram C++, Python, Java, eksistē arī vairāki ietvari, kas paredzēti OpenCV funkciju izsaukšanai no citām programmēšanas valodām, piemēram EmguCV[10] ir ietvars, kuš ļauj piekļūt OpenCV attēlu apstrādes funkcijām izmantojot .NET savietojamas programmēšanas valodas. Autors izvēlas pirmo prototipu veidot Python programmēšanas valodā, jo tā piedāvā augsta līmeņa programmēšanas valodas priekšrocības un autoram ir salīdzinoši liela pieredze darbā ar šo programmēšanas valodu.

2.2. Metodes

Pēc autora domām eksistē vairākas pieejas kā padarīt priekšā braucošo automašīnu caurspīdīgu, viena no pieejām ir atpazīt to bildē, kas uzņemta no aizmugurē braucošās

automašīnas un atpazīto reģionu aizstāt ar attiecīgo daļu no bildes, ko uzņēmusi priekšā braucošā automašīna. Šāds variants būtu ļoti labs, jo sniegtu precīzu informāciju par to, kas atrodas aiz priekšā braucošās automašīnas, nemainot to informāciju, kas aizmugurē braucošās automašīnas vadītājam jau pieejama no viņa skata punkta. Šādai pieejai autors ir atradis divas metodes – kontūru noteikšanas metodi un objektu atpazīšanas metodi, kas aprakstītas attiecīgi nodaļās 2.2.1. un 2.2.2.

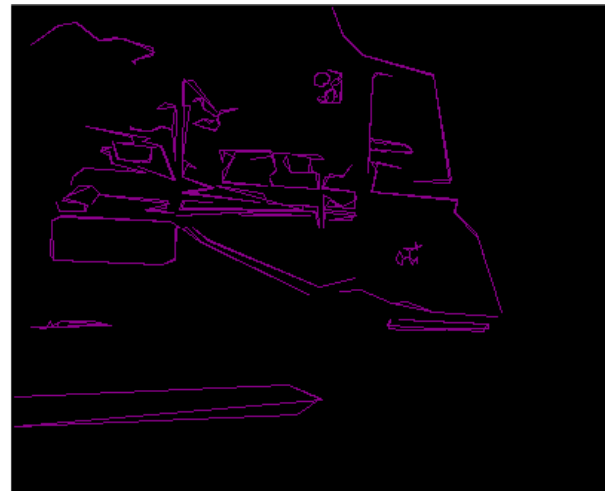
Otra pieeja ir padarīt automašīnu caurspīdīgu nenosakot precīzu tās atrašanās vietu attēlā, to iespējams izdarīt apvienojot abus iegūtos attēlus vienā. Šādai pieejai autors ir atradis vienu metodi – attēlu sapludināšanas metode, kas aprakstīta nodaļā 2.2.3.

2.2.1. Kontūru noteikšanas metode

Lai noteiktu automašīnas atrašanās vietu attēlā kā viena no metodēm tika pārbaudīta kontūru noteikšanas metode. Metode sastāv no diviem etapiem, pirmais ir stūru noteikšana, tā angļu valodā nosaukums ir „Canny edge detector”[11], tā pamatā ir 5 soļi, kas jāveic, lai samazinātu attēlā redzamo datu apjomu saglabājot attēla struktūras informāciju, ko var izmantot tālākai datu apstrādei. Pēc attēla apstrādes ar šo algoritmu tiek iegūts binārs attēls, kurā atzīmēti visi stūri ko šis algoritms ir atradis, 2.1. att. (a) redzams 2.10. att. (b) attēls pēc apstrādes ar Canny algoritmu. Canny algoritmam ir divi parametri, kas maina tā darbību, autors šos parametrus pieregulēja attiecīgi tā, lai iegūtu pēc iespējas detalizētāku informāciju par automašīnas kontūrām, neņemot vērā pārējo iegūto informāciju, kas neattiecas uz automašīnu. Kā redzams attēlā ir saglabājušās automašīnas kontūras, bet vēl jau projām ir arī ļoti daudz informācijas par apkārtējo vidi, kas traucē viennozīmīgi noteikt automašīnas atrašanās vietu bildē, tāpēc tiek lietots otrais etaps, kas binārā attēlā meklē kontūras, tiek izmantots algoritms, kas aprakstīts [12], tas atgriež masīvu ar punktiem, kas atbilst attēlā atrastajām kontūrām. 2.1. att. (b) redzams kādas kontūras tiek atrastas izmantojot šo algoritmu. Kā redzams kontūru atlasīšana ir samazinājusi informācijas apjomu, bet tajā vēl jau projām nav iespējams skaidri noteikt automašīnas atrašanās vietu vai iezīmēt izteiktu tās kontūru.



(a) Canny algoritma sniegtais rezultāts



(b) Attēlā atrastās kontūras

2.1. att. Kontūru noteikšanas metodes etapu rezultāti

Aplūkojot 2.1. att. (a) tika secināts, ka tajā nemaz nav viennozīmīgi iespējams izšķirt automašīnu, jo pilna tās kontūra nav saskatāma, tās sānos iztrūkst vairākas līnijas, lai pārliecinātos, ka šo problēmu neizraisa Canny algoritma darbināšanai izmantotie parametri, autors apstrādāja šo attēlu ar pašiem zemākajiem parametriem, lai iegūtu pilnīgi visas kontūras ko vien Canny algoritms spēj atrast, kā redzams 2.2. att. tomēr nav saskatāma pilna automašīnas kontūra, kas liek secināt, ka lai izmantotu stūru noteikšanas metodi automašīnas atpazīšanai nāktos implementēt kādu interpolācijas algoritmu, kas pilnveidotu mašīnas kontūru.

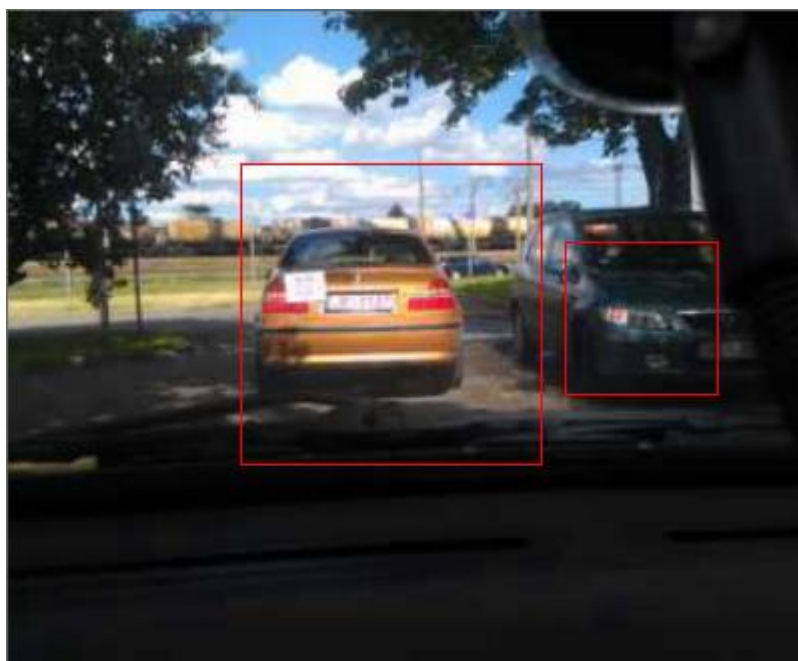


2.2. att. Canny algoritma sniegtais rezultāts ar zemākajiem parametriem

2.2.2. Objektu atpazīšanas metode

Lai noteiktu automašīnas vietu attēlā, kā viena no metodēm tika pārbaudīta objektu atpazīšanas metode izmantojot Haar kaskādes[13], kas tiek plaši izmantotas sejas atpazīšanā[14]. Metodes būtība ir kaskādes datnes izveide, kas apraksta meklēto objektu, šādu datni jāveido no vairākiem attēliem, kas satur meklēto objektu un vairākiem attēliem kas to nesatur. Kad datne ir izveidota, to iespējams izmantot ātrai meklējamā objekta atrašanai attēlos, bet pati datnes veidošana ir ļoti laikietilpīgs process.

Lai pārbaudītu vai šāda metode ir pielietojama priekšā braucošās automašīnas atpazīšanai, autors izvēlējās pārbaudīt vai ar šo metodi ir iespējams atpazīt vienu konkrētu automašīnu, kas redzama 2.10. att. (b). Vienas konkrētas automašīnas atpazīšana ir salīdzinoši vienkāršāks process, tāpēc nav nepieciešams ļoti liels daudzums ieejas attēlu. Tika izveidota kaskādes datne izmantojot 209 attēlus, kuros redzama izvēlētā automašīna un vairāk nekā 500 attēli, kuros šī automašīna nav redzama.



2.3. att. **Automašīnas atpazīšana izmantojot kaskādes datni**

Pēc datnes iegūšanas tomēr izrādījās, ka automašīnas atpazīšana izmantojot šo kaskādes datni nav precīza un retos gadījumos atgriež pareizu automašīnas atrašanās vietu attēlā un ļoti daudzos gadījumos atgriež vairākas nepareizas automašīnas atrašanās vietas. 2.3. att. redzamas ar sarkanu iezīmētas vietas, kur atrasta automašīna, bet lielākais taisnstūris ietver daudz lielāku apgabalu nekā meklētā automašīna un mazākais taisnstūris vispār nav meklētās automašīnas tuvumā. Autors saskata divas iespējamās problēmas šādai neprecīzai

automašīnas atrašanās vietas noteikšanai. Viena no tām ir zemā attēlu kvalitāte, kas izmantota kaskādes datnes izveidē, bet šādas pašas kvalitātes attēlos arī tiek meklēta izvēlētā automašīna, tāpēc nav īsti skaidrs vai tā tiešām ir problēma. Kā otro problēmu autors redz salīdzinoši mazo izmantoto attēlu skaitu kaskādes datnes iegūšanai, jo pēc [15] nepieciešams 5000 attēlu, kas satur meklējamo objektu un 3000 attēlu, kas nesatur to, bet šādas testa datu kopas sagatavošana aizņemtu ļoti daudz laika.

2.2.3. Attēlu sapludināšanas metode

Metodes pamatā ir doma, ka abi attēli, ko uzņēmušas viena aiz otras braucošās automašīnas satur ļoti daudz kopīgās informācijas, tāpēc šos attēlus būtu iespējams apvienot vienā attēlā veicot attēlu sapludināšanu. Izmantojot šo metodi ir ļoti svarīgi kurā vietā abas mašīnas atradušās brīdī, kad attēls ticis uzņemts.

Ja no abām mašīnām tiktu saņemts attēls, kurš uzņemts vienā vietā dažādos laika posmos, ar vienādu izmēru un skata leņķi, šos attēlus būtu ļoti viegli sapludināt un iegūt caurspīdīgu priekšā braucošo automašīnu. Diemžēl šādu apstākļu sakritību ir grūti nodrošināt, lai gan tā sniegtu vieglu atbildi uz šajā maģistra darbā izvirzīto problēmu.

Pastāv iespējas apstrādāt abus attēlus, lai iegūtu tuvinājumu augstāk aprakstītajai situācijai un tad iegūt to sapludināto attēlu kurā būtu redzama caurspīdīga priekšā braucošā automašīna. Jāatzīmē, ka lai attēlu sapludināšana sniegtu korektus rezultātus par to, kas atrodas aiz priekšā braucošās automašīnas, nepieciešams attēlu sapludināšanai izmantot divus attēlus, kas uzņemti pēc iespējas tuvāk viens otram, ideāli vienā un tajā pašā vietā, pretējā gadījumā attēlu sapludināšana var sniegt daļēji vai pilnīgi nepatiesu informāciju par to, kas atrodas aiz priekšā braucošās automašīnas.

Lai sagatavotu attēlus sapludināšanai sākumā nepieciešams saprast kā šie dažāda izmēra un skata punkta attēli ir savā starpā saistīti, lai to izdarītu secīgi tiek izmantotas sekojošas darbības:

1. Iezīmju atrašana;
2. Iezīmju atbilstību noteikšana;
3. Transformācijas atrašana;
4. Transformācijas pielietošana.



(a) Attēls pirms apgriešanas



(b) Attēls pēc apgriešanas

2.4. att. Attēls pirms un pēc apgriešanas

Lai attēlu sapludināšanas process būtu efektīvāks nepieciešams attēlu, kas iegūts no priekšā braucošas automašīnas apgriezt tā, lai nebūtu redzama pati automašīna. Pašlaik autors nav izstrādājis universālu metodi automašīnas daļu izgriešanai no attēla, bet tā kā tiek izmantoti attēli tikai no vienas priekšā braucošas automašīnas, tad šī attēla apgriešana tiek veikta manuāli norādot cik daudz pikselus noņemt no attēla apakšas un labās puses. Pēc attēla apgriešanas attēla izmēri ir 160x280 pikseļi, attēls pirms un pēc apgriešanas redzams 2.4. att. Šāda attēla apgriešana atrisina vairākas potenciālas problēmas:

- Attēlu sapludināšanas rezultātā iegūtajā attēlā būs redzamas priekšā braucošas automašīnas daļas, kas nav vēlamas;
- Iezīmju meklēšanas rezultātā var tikt atrastas vairākas viltus sakritības, kas atrodas tieši uz automašīnas daļām, kā rezultātā var veidoties nepareiza transformācija un izkropļots rezultātā iegūtais attēls.

2.2.3.1 Iezīmju atrašana

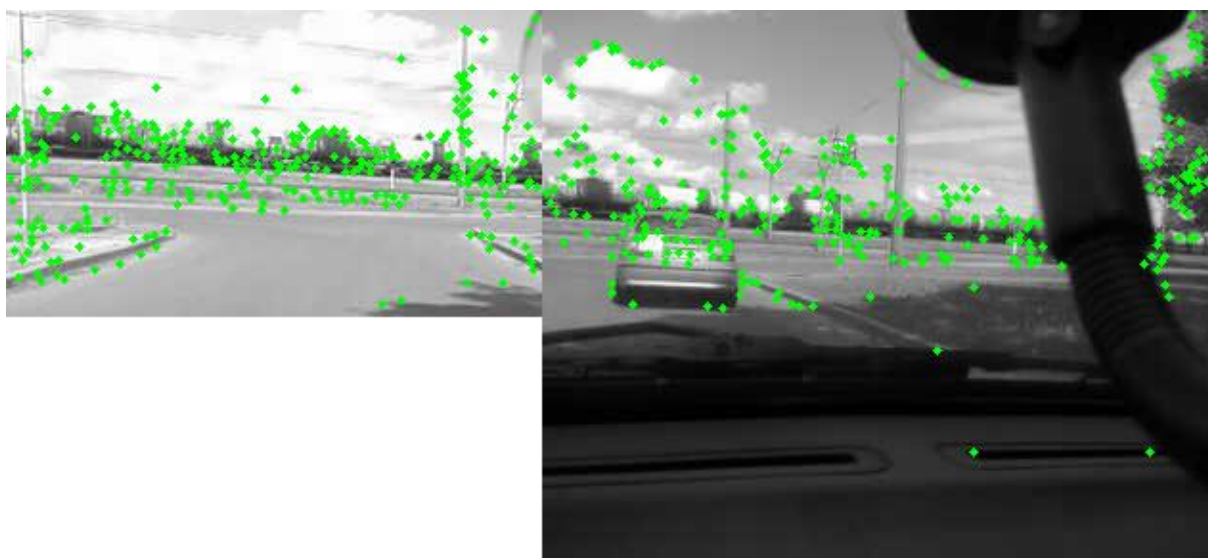
Iezīmju atrašanai tiek izmantotas pelēktoņu bildes, jo tajās novērojams mazāks apgaismojuma īpatsvars, kas ir svarīgi lai korektāk noteiktu iezīmes attēlos.

Eksistē vairāki algoritmi iezīmju atrašanai: SURF[16], SIFT[17], ORB[18], u.c. Pēc aprakstiem SIFT ir visefektīvākais, bet ORB visātrākais. Autors izvēlas lietot SIFT algoritmu, jo ātrdarbība šajā izstrādes fāzē nav tik svarīga kā precizitāte. Svarīgi ir tas, ka šie iezīmju atrašanas algoritmi spēj labi tikt galā ar troksni un nelielām perspektīvas maiņām, kas nozīmē, ka lielākā daļa attēlā atrastās iezīmes tiks atrastas arī tad, ja attēls tiks nedaudz transformēts,

t.i. tiks mainīts skata punkts, kas mūsu gadījumā ir ļoti svarīgi, jo pastāv ļoti liela varbūtība, ka uz abām automašīnām uzstādītajām kamerām skatu punkti nesakrīt.

Iezīmju atrašanas algoritms atgriež sarakstu ar iezīmēm un katrai iezīmei tās zīmīgo apgabalu, kas raksturo šo iezīmi. Katras iezīmes zīmīgā apgabala apraksts ir 128 dimensiju vektors. Šis vektors tiek veidots no 16 apgabaliem, katrs apgabals atbilst 4x4 pikseļus lielam laukumam iezīmes tuvumā un visi kopā tie noklāj laukumu ap iezīmi izmērā 16x16 pikseļi. Katram no šiem 16 apgabaliem atbilst 8 vektori, kur katrs vektors satur histogrammu no šī apgabala aprakstītā 4x4 pikseļus lielā laukuma, kas tiek aprēķināta ik pa 45 grādiem.

Šāds iezīmju saraksts tiek iegūts no abiem attēliem kuriem gribam noskaidrot to saistību. 2.5. att. redzamas visas iezīmes, ko SIFT algoritms atradis abos attēlos, 2.5. att. (a) atrastas 344 iezīmes, bet 2.5. att. (b) 451 iezīme. Kā redzams tiek atrastas ļoti daudz iezīmes un lielākā daļa no tām ir tikai vienā no attēliem. Tas skaidrojams ar to, ka abiem attēliem ir salīdzinoši maza kopīgā daļa, kurā būtu redzami vieni un tie paši objekti.



(a) Attēls no priekšā braucošās automašīnas

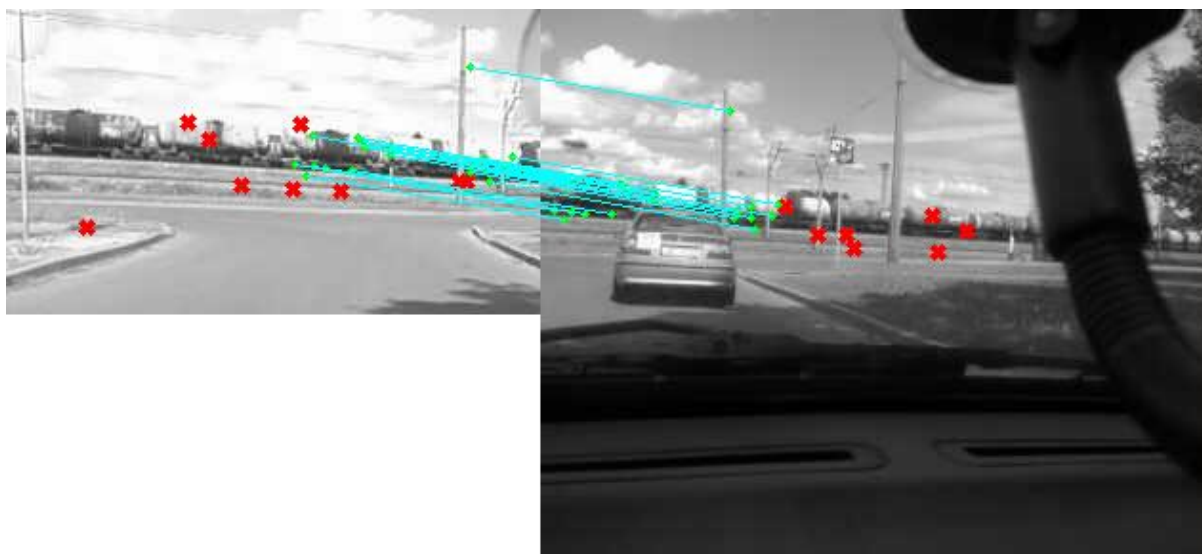
(b) Attēls no aizmugurē braucošās automašīnas

2.5. att. Attēlos atrastās iezīmes

2.2.3.2 Iezīmju atbilstību noteikšana

Kad diviem attēliem ir noteiktas to raksturīgās iezīmes ir iespējams šīs iezīmes salīdzināt un atrast savā starpā līdzīgās, tādējādi atrodot līdzīgus punktus un to apkārtnes abos attēlos. Iezīmju atbilstību noteikšanu SIFT iezīmēm iespējams veikt izmantojot algoritmu, kas katram punktam identificē tā tuvākās sakritības otrā attēlā izmantojot Eiklīda distanci 128

dimensiju vektoru telpā. Konkrētajā gadījumā sakritības pareizuma pārbaudei nepieciešams atrast divas tuvākās sakritības otrā attēlā katrai iezīmei.



(a) Attēls no priekšā braucošās automašīnas

(b) Attēls no aizmugurē braucošās automašīnas

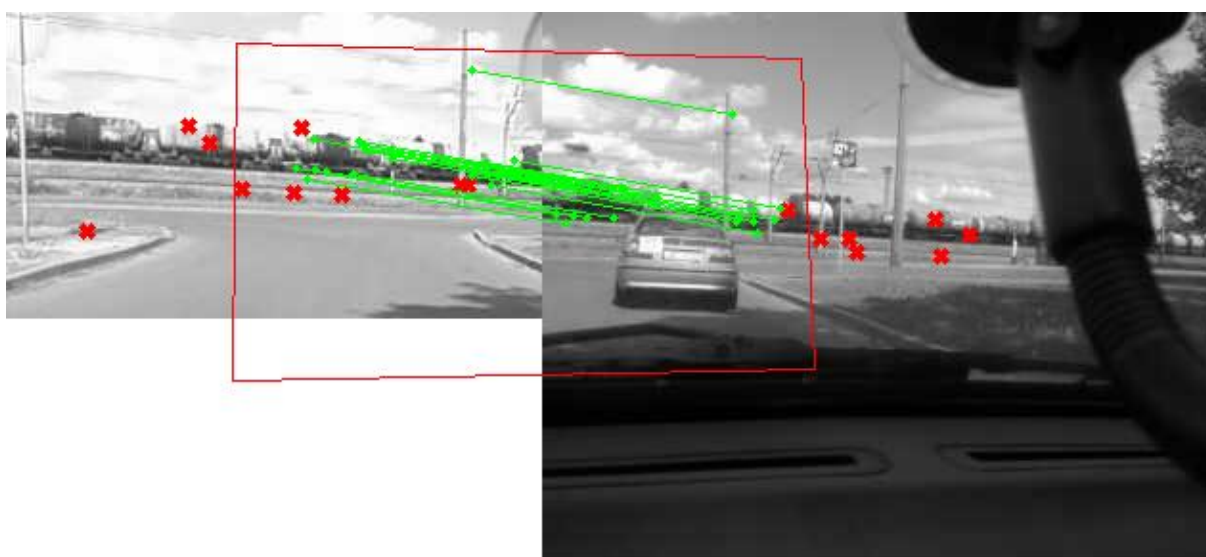
2.6. att. Atrastās iezīmes pēc nepatieso sakritību atlasīšanas

Tā kā iezīmju atrašanai tika izmantots algoritms, kurš spēj tikt galā ar nelielām perspektīvas nobīdēm, tad šajā etapā abās bildēs tiešām tiek atrasti atbilstoši punkti abās bildēs ar lielu varbūtību, tomēr parasti šāda meklēšana atgriez ļoti daudz nepatiesas sakritības, tāpēc ir nepieciešams atlasīt nost lielu daļu šīs nepatiesās sakritības. Lai atlasītu nost nepatiesās sakritības tiek izmantots nosacījums par divu tuvāko sakritību attālumu attiecību. Balstoties uz [17], ja Eiklīda distance starp konkrēto iezīmi un otru labāko sakritību ir lielāka kā 80% no Eiklīda distances starp konkrēto iezīmi un labāko sakritību, tad šī iezīme ir jāatmet kā nepareiza sakritība. [17] autors apgalvo, ka tādejādi iespējams atlasīt nost līdz pat 90% no nepatiesajām sakritībām zaudējot tikai 5% patieso sakritību.

2.6. att. iezīmētas visas 20 sakrītošās iezīmes abos attēlos, zaļā krāsā iekrāsotas tās iezīmes, kas sakrīt, bet sarkanā krāsā tās iezīmes, kurām atbilstošas iezīmes otrā attēlā nav atrastas. Gaiši zilas iezīmētas līnijas, kas savieno atbilstošos sakrītošos punktus abās bildēs. Salīdzinot 2.5. att. un 2.6. att. redzams, ka nepatieso sakritību atlasīšana ir veiksmīgi tikusi vaļā no lielākās daļas nepatieso iezīmju un ir palikušas tikai 9 nepatiesas sakritības 2.6. att. (a) attēlā un 7 nepatiesas sakritības 2.6. att. (b).

2.2.3.3 Transformācijas atrašana

Ja apgalvojums, ka attēlu, kas uzņemts no priekšā braucošās automašīnas var atrast attēlā, kas uzņemts no aizmugurē braucošās automašīnas ir patiess, tad visticamāk, lai šo atrasto attēlu varētu ievietot attēlā kurā tas tika atrasts, tam būs jāpielieto perspektīvā transformācija. Šajā gadījumā perspektīvo transformāciju jāpielieto attēlam, kas iegūts no priekšā braucošās automašīnas. Pēc perspektīvās transformācijas visas līnijas, kas attēlā bija paralēlas, arī tiks saglabātas paralēlas, bet attēla perspektīva tiks izmainīta.



(a) Attēls no priekšā braucošās automašīnas

(b) Attēls no aizmugurē braucošās automašīnas

2.7. att. Iezīmju sakrītības un perspektīvās transformācijas četrstūris

Lai veiktu perspektīvo transformāciju nepieciešams iegūt 3×3 matricu, kas apraksta šo transformāciju, šīs matricas iegūšanai nepieciešams atrast vismaz 4 sakrītošus punktus abos attēlos, t.i. nepieciešams atrast aizmugurē braucošās automašīnas uzņemtajā bildē vismaz 4 tādus punktus, kas sakrīt ar 4 punktiem no attēla, kas uzņemts no priekšā braucošās automašīnas. Jāatzīmē, ka šeit domāti 4 punkti kas sakrīt pēc būtības, t.i. pieder vienam un tam pašam objektam, un atrodas vienā un tajā pašā vietā attiecībā pret šo objektu abos attēlos. Vismaz 4 šādus punktus sniedz SIFT iezīmju meklēšanas algoritms, ja tam ir izdevies atrast vismaz 4 sakrītošas iezīmes, tāpēc kā atbilstošos punktus var izmantot SIFT algoritma rezultātu.

Perspektīvās transformācijas matricas iegūšanai tiek izmantots algoritms[19], kas no atbilstošo punktu koordinātēm izrēķina 3×3 perspektīvās transformācijas matricu, no kuras var iegūt informāciju par to, kurā vietā precīzi jānovieto transformētais attēls, kas iegūts no priekšā braucošās automašīnas, attēlā, kas iegūts no aizmugurē braucošās automašīnas. 2.7.

att. redzams ar sarkanu līniju iezīmēts četrstūris, kas atbilst pozīcijai, kur 2.7. att. (b) jāievieto 2.7. att. (a) pēc veiktās perspektīvās transformācijas, jāatzīmē, ka iegūtais četrstūris nav taisnstūris, tātad, ka šis četrstūris ir iegūts pielietojot iegūto perspektīvo transformāciju taisnstūrim, kas atbilst 2.7. att. (a) malām.

2.2.3.4 Transformācijas pielietošana

Kad iegūta perspektīvās transformācijas matrica, atliek vien to pielietot priekšā braucošās automašīnas uzņemtajam attēlam, lai iegūtu attēlu, ko ievietot atbilstošajā vietā aizmugurē braucošās automašīnas uzņemtajam attēlam. 2.8. att. redzams iegūtais rezultāts izmantojot SIFT algoritmu.

Jāatzīst, ka iegūtais rezultāts nav ideāls, ir acīmredzamas nesakritības starp attēliem, īpaši tas ir manāms ceļa labajā malā vērojot apmali. Vēl viena problēma ir redzamā melnā pārtrauktā līnijā, kas iezīmē ievietotā attēla kontūru, tas ir blakusefekts, ko autoram pagaidām vēl nav izdevies novērst. Kā arī priekšā braucošās automašīnas uzņemtais attēls ir daudz gaišāks nekā aizmugurējās automašīnas uzņemtais attēls, tātad ievietotais fragments izskatās pārgaismots. Bet, neskatoties uz vairākām nepilnībām, iegūtais attēls ir ļoti labs tuvinājums attēlam bez priekšā braucošās automašīnas, kas principā nodrošina iespēju redzēt kas atrodas objektam – priekšā braucošajai automašīnai otrā pusē.



2.8. att. Iegūtais attēls pēc transformācijas pielietošanas un attēlu apvienošanas

2.2.3.5 Caurspīdīga objekta iegūšana

Kad ir iegūts attēls bez priekšā braucošās automašīnas, ir iespējams to sapludināt ar attēlu, kur ir redzama priekšā braucošā automašīna, tādējādi iegūstot caurspīdīga objekta efektu. Attēlu sapludināšanu iespējams veikt izmantojot algoritmu, kas apvieno divus attēlus sasummējot atbilstošo pikseļu krāsu vērtības pielietojot svērtos koeficientus. Lai iegūtu attēlu ar caurspīdīgu mašīnu tiek apvienoti 2.10. att. (b) un 2.8. att., attiecīgi tiem pielietojot koeficientus 0.4 un 0.6, rezultātā tiek iegūts 2.9. att.



2.9. att. Caurspīdīga objekta iegūšanas rezultāts

Kā redzams 2.9. att. nospraustais mērķis ir panākts un attēlā ir redzama caurspīdīga priekšā braucošā automašīna. Pielietotā attēlu sapludināšanas metode ļauj iegūt dažādus priekšā braucošās automašīnas caurspīdīguma līmeņus, sākot ar pilnībā redzamu automašīnu, kā redzams 2.10. att. (b), beidzot ar pilnībā neredzamu automašīnu, kā redzams 2.8. att.

2.2.4. Secinājumi par metodēm

Gan stūru noteikšanas metode, gan Haar kaskādes ir paredzētas tieši objektu atpazīšanai attēlos, tomēr šajā konkrētajā gadījumā autoram neizdodas panākt automašīnas atrašanu, tāpēc darbs šajā virzienā netiek turpināts.

Attēlu sapludināšanas metode gan ir sniegusi rezultātu, kas ir pielīdzināms šā maģistra darba izvirzītajam mērķim, tāpēc autors izvēlas turpmākam darbam izmantot tikai šo metodi. Turklāt attēlu sapludināšanas metodei ir priekšrocība, ka tā nav ierobežota tikai ar

automašīnām, principā jebkas, kas ir redzams vienā attēlā un nav redzams otrā attēlā tiks padarīts caurspīdīgs izmantojot šo metodi.

2.3. Eksperimenti un novērtējums

Lai novērtētu cik labi ir izdevies izstrādāt šo prototipu caurspīdīgas automašīnas iegūšanai autors veic eksperimentu ar vairākiem pēc kārtas iegūtiem attēliem no divām automašīnām, kas pārvietojas pa ceļu viena aiz otras.

2.3.1. Pirmā prototipa pārbaudei izmantotie attēli

Lai pārbaudītu prototipa darbību nepieciešami attēli, kas uzņemti no divām automašīnām, kas pārvietojušās pa ceļu viena aiz otras. Autora rīcībā ir šādi attēli no iepriekš veiktiem eksperimentiem, kuros mērķis bija aizmugurē braucošās automašīnas vadītājam rādīt bildi no priekšā braucošās automašīnas bez jebkādas apstrādes. Šie iepriekšējie eksperimenti tika veikti kopā ar Sergeju Voronovu, kurš pilnībā izstrādāja programmatūru, kas darbojās uz viedtālruniem, autora pārziņā bija servera puses datu apstrādes nodrošināšana. Iegūto attēlu paraugi redzami 2.10. att. Jāatzīmē, ka tā kā attēli tika savākti tikai servera pusē, to kvalitāte ir salīdzinoši zema, jo datu pārraides apjoma taupīšanas dēļ attēli pirms pārsūtīšanas tika saspiesti.



(a) Bilde no priekšā braucošās automašīnas



(b) Bilde no aizmugurē braucošās automašīnas

2.10. att. Iepriekš veiktajā eksperimentā iegūto attēlu paraugi

Kā redzams starp 2.10. att. (a) un 2.10. att. (b) ir ļoti daudz būtisku atšķirību, attēli ir dažāda izmēra, 2.10. att. (a) izmēri ir 480x320 pikseļi, bet 2.10. att. (b) izmēri ir 352x288

pikseļi, izmērs gan nav tā būtiskākā atšķirība, jo to var viegli novērst apgriežot lielāko attēlu vai mērogojot mazāko. Lielākas problēmas sagādā tas, ka kameras uz abām automašīnām nav novietotas ar vienādu skata punktu, 2.10. att. (a) redzams, ka kamera vairāk pavērsta uz ceļu un mazāk attēlo to, kas notiek blakus ceļam pa labi no automašīnas, savukārt 2.10. att. (b) kamera pavērsta vairāk uz to, kas atrodas pa labi no ceļa un mazāk attēlo tieši to, kas notiek uz ceļa kreisajā pusē. Vēl viena nianse, kas nav tik labi redzama, bildēm nav vienāds gaišums, 2.10. att. (a) objekti izskatās gaišāki nekā 2.10. att. (b), tas skaidrojams ar to, ka tika izmantoti dažādi viedtālruni ar dažādām kamerām.

2.3.2. Vairāku secīgu attēlu sapludināšana

Lai novērtētu cik veiksmīgi izdevies izstrādāt algoritmu, kas attēlotu priekšā braucošo automašīnu caurspīdīgu, izveidotais attēlu sapludināšanas algoritms tika darbināts uz vairākām secīgām bildēm, kur divas automašīnas pārvietojas pa ceļu viena aiz otras. Visi šajā eksperimentā iegūtie attēli redzami tiešsaistē izvietotajā video <http://youtu.be/f-wg34euXkM>.



2.11. att. Vairāki secīgi caurspīdīgas automašīnas attēli

Lai iegūtu šādus vairākus secīgus attēlus tika izmantots modificēts attēlu sapludināšanas algoritms, kas pirms attēlu sapludināšanas izvēlējās vienu attēlu no aizmugurē braucošās automašīnas un mēģināja tam piemeklēt vislabāk atbilstošo attēlu no priekšā braucošās automašīnas, kā kritēriju labākajai atbilstībai izmantojot sakrītošo iezīmju skaitu. Diemžēl izrādījās, ka iezīmju meklēšana uz šādiem zemas kvalitātes attēliem nestrādā tik labi kā gribētos un priekšā braucošās automašīnas uzņemtais attēls, kam sakrīt visvairāk iezīmju ar aizmugurē braucošās automašīnas uzņemto attēlu ne vienmēr ir optimālais attēls ko izvēlēties attēlu sapludināšanai caurspīdīgas priekšā braucošās automašīnas iegūšanai. 2.11. att. labi redzams, ka, lai arī priekšā braucošā automašīna ir caurspīdīga, tālumā redzamo koku silueti nesakrīt, tas nozīmē, ka ievietotais attēls nav uzņemts tajā pašā vietā, kur oriģinālais attēls vai arī ievietotā attēla perspektīva ir nepareiza. No kā seko, ka informācija, kas attēlota aiz caurspīdīgās automašīnas var būt nepilnīga vai arī pilnīgi nekorekta.

2.4. Secinājumi par pirmo prototipu

Autors mēģināja iegūt caurspīdīgu automašīnu izmantojot trīs metodes – kontūru noteikšanu, objektu atrašanu un attēlu sapludināšanu. No šīm trīs metodēm pozitīvu rezultātu izdevās iegūt vienīgi izmantojot attēlu sapludināšanas metodi.

Veicot eksperimentus ar attēlu sapludināšanas metodi autors secināja, ka iegūtie attēli nav pietiekoši labi, jo tie satur nepilnīgu vai nepareizu informāciju par to, kas atrodas aiz caurspīdīgās automašīnas, tāpēc autors uzskata, ka tas nav pieņemams rezultāts autora izvēlētajā piemērā ar automašīnām un nav apmierinošs rezultāts šī maģistra darba ietvaros.

Prototipa pārbaudei izmantoto attēlu kvalitāte bija salīdzinoši zema, autors uzskata, ka tas var būt viens no iespējamajiem cēloņiem iegūtajam nepieņemamajam rezultātam.

Pēc pirmā prototipa implementācijas koda apskates autors izlemj, ka Python programmēšanas valoda nav piemērota šāda prototipa izveidei, jo kods ātri vien kļūst nelasāms tāpēc, ka OpenCV Python implementācijā viss tiek glabāts neaprstītās datu struktūrās, kas pirmā prototipa izstrādes laikā bieži nobremzēja prototipa izstrādi, jo nācās cīnīties ar kļūdām kā piemēram, ka funkcijai tiek padots nepareiza formāta masīvs un tamlīdzīgi.

3. OTRAIS PROTOTIPS

Ņemot vērā pirmā prototipa rezultātus tika nolemts, ka tālāka maģistra darba attīstība notiks sekojošos virzienos:

- Kvalitatīvu ieejas datu iegūšana;
- Implementācijas uzlabošana;
- Automātiska sapludinātā attēla korektuma pārbaude;
- Pareizo attēlu atrašana priekš sapludināšanas.

3.1. Kvalitatīvu ieejas datu iegūšana

Lai veiktu turpmākus eksperimentus ar attēlu sapludināšanas algoritmu bija nepieciešams iegūt labas kvalitātes attēlus ar divām vai vairāk automašīnām, kas pārvietojas pa ceļu viena aiz otras. Darba gaitā autors veica trīs testa datu iegūšanas piegājienu. Lai pārbaudītu vairākas hipotēzes, autors vēlas iegūt attēlus, kas uzņemti no 3 pēc kārtas braucošām automašīnām, attēlus, kuru uzņemšanas vieta fiksēta izmantojot GPS koordinātes un attēlus, kas uzņemti izmantojot vienādas attēlu uzņemšanas ierīces.

3.1.1. Testa datu iegūšana no trīs automašīnām

Ar nolūku iegūt testa datus tikai veikts brauciens ar 3 automašīnām, kurās bija uzstādīti viedtālruni, kas filmēja ceļu, kā arī reizi sekundē fiksēja automašīnas atrašanās vietu izmantojot GPS. Brauciens ilga aptuveni 8 minūtes, kurās automašīnas veica aptuveni 4 km garu distanci prom un atpakaļ, kopā veicot 8 km garu distanci. Aptuveni skaitļi minēti, jo video ilgums katrā no uzņemtajiem video nedaudz atšķiras, jo tie tika ieslēgti un izslēgti manuāli, kā arī veiktā ceļa distance ņemta no kartes. Automašīnās bija uzstādīti sekojoši viedtālruni:

- Pirmajā automašīnā atradās HTC ONE S viedtālrunis, kas uzņēma video ar izšķirtspēju 1920x1088 ar 29 kadriem sekundē.
- Otrajā automašīnā atradās HTC Desire X viedtālrunis, kas uzņēma video ar izšķirtspēju 352x288 ar 30 kadriem sekundē.

- Trešajā automašīnā atradās HTC EVO 3D viedtālrunis, kas uzņēma video ar izšķirtspēju 1280x720 ar 24 kadriem sekundē.



3.1 att. Trīs automašīnās novietoto viedtālrunu uzņemto attēlu paraugi

Attēlā redzami uzņemtā video paraugi no katras automašīnas, 3.1 att. (a) no priekšā braucošās automašīnas video, 3.1 att. (b) no otrās automašīnas video un 3.1 att. (c) no trešās automašīnas video. 3.1 att. (b) redzams, ka otrajā automašīnā uzstādītā viedtālruna video izšķirtspēja ir ievērojami zemāka kā pārējiem diviem, šajā gadījumā tam nevajadzētu būt šķērslim, jo autors plāno pārbaudīt, vai caurspīdīgu objektu ir iespējams iegūt no dažādu izšķirtspēju video. Kā arī redzams, ka 3.1 att. (c) ir daudz tumšāks par abiem pārējiem, diemžēl šis ir attiecīgā izmantotā telefona defekts, bet tas atkal ļaus pārbaudīt vai šādi video arī der caurspīdīga objekta iegūšanai.

3.1.2. Testa datu iegūšana no imitētām automašīnām

Ar nolūku iegūt testa datus, kuros abas attēlu uzņemšanas ierīces būtu vienādas autors izvēlas izmantot tīkla kameras Airlive POE-100CAM v2[20], jo autoram šī eksperimenta veikšanas brīdī nebija pieejami divi vienādi viedtālruņi, ko uzstādīt uz automašīnām. Attēlus no tīkla kameras iespējams iegūt tikai un vienīgi caur tai pievienoto tīkla vadu.

Tā kā tīkla kameru uzstādīšana automašīnās arī nav vienkārša, autors izvēlas automašīnas imitēt ar biroja krēsliem, uz kuriem uzstādītas tīkla kameras. Tika izmantoti divi krēsli. Uz abiem krēsliem tika uzstādīta tīkla kamera un UPS iekārta, kas spēj pietiekoši ilgu laiku darbināt šo tīkla kameru. Uz aizmugurējā krēsla tika uzstādīts arī maršrutētājs lokālā tīkla izveidošanai un portatīvais dators, kas saglabātu datus no tīkla kamerām. Abi krēsli tika sasieti kopā ar aptuveni 2m garu trosi, kā arī no abām tīkla kamerām tika novilkts tīkla vads līdz maršrutētājam.

Testa datu iegūšanas laikā abi ar tīkla kamerām aprīkotie biroja krēsli tika pārvietoti pa gaiteni viens aiz otra, testa datu iegūšana ilga 101 sekundi.



(a) Aizmugurē uzstādītās tīkla kameras iegūtais attēls



(b) Priekšā uzstādītās tīkla kameras iegūtais attēls

3.2 att. Tīkla kameras iegūto attēlu paraugi

Testa datu iegūšana tika veikta Elektronikas un Datorzinātņu institūta telpās. Tīkla kameras iegūto attēlu izšķirtspēja ir 640x480 pikseļi, tika fiksēti 25 kadri sekundē. Attēlā 3.2 att. redzami iegūtie attēli, 3.2 att. (a) redzams aizmugurē novietotās tīkla kameras iegūtais attēls un 3.2 att. (b) redzams priekšā novietotās tīkla kameras iegūtais attēls. Kā redzams, iegūto attēlu kvalitāte ir salīdzinoši slikta.

3.1.3. Testa datu iegūšana no vienādiem viedtālruniem

Laika gaitā autora paziņu lokā parādījās 2 cilvēki, kam piederēja vienādi viedtālruni, kurus autoram izdevās aizņemt testu datu iegūšanai no vienādām attēlu uzņemšanas ierīcēm, ko var arī ērti uzstādīt automašīnā. Izmantotais viedtālrunis bija HTC Desire 5000, kurš spēj ierakstīt video ar 29 kadriem sekundē izšķirtspējā 800x480 pikseļi. Izvēlētajā testa datu iegūšanas laikā laika apstākļi bija lietaini, bet tā kā izmantotie viedtālruni nav autora īpašums, eksperimentu uz citu laiku pārcelt nebija iespējams.

Iegūtajos attēlos uz automašīnas priekšējā vējstikla ir redzamas ūdens lāses un ik pa laikam tiek darbināts logu tīrīšanas mehānisms, kas daļēji aizsedz skatu uz ceļu. Autors uzskata, ka iegūtie testa dati tomēr būs derīgi, vismaz tam, lai noskaidrotu vai izstrādātais caurspīdīgās automašīnas iegūšanas algoritms spēj darboties šādos laika apstākļos.

Testa datu iegūšanas laikā aizmugurē braucošajā automašīnā uzstādītais viedtālrunis turētājs nokrita no stikla, tāpēc brauciens bija jāpārtrauc pēc 124 sekundēm.



(a) Attēls no aizmugurē braucošās automašīnas

(b) Attēls no priekšā braucošās automašīnas

3.3 att. Vienādu viedtālrunu iegūto attēlu paraugi

Iegūto attēlu paraugs no aizmugurē braucošās automašīnas 3.3 att. (a) un priekšā braucošās automašīnas 3.3 att. (b), kā redzams lietainā laika dēļ iegūtie attēli ir salīdzinoši tumši.

3.2. Implementācijas uzlabošana

Tā kā pirmā prototipa izstrādes laikā autoru neapmierināja OpenCV attēlu apstrādes bibliotēkas implementācija Python programmēšanas valodā, tad autors nolemj otro prototipu veidot C++ programmēšanas valodā izmantojot to pašu OpenCV attēlu apstrādes bibliotēku.

3.2.1. Ātrdarbības novērtējums

Pēc algoritma pārņemšanas uz C++ valodu tika nolemts pārbaudīt kāda ir izstrādātā algoritma ātrdarbība. Lai veiktu ātrdarbības testus tika ar roku izveidoti divi video, kurus apvienojot kadru pa kadram būtu iespējams iegūt sapludinātu attēlu ar caurspīdīgu priekšā braucošo automašīnu. Tika atlasīti 600 šādi kadru pāri, iepriekš pārlicinoties, ka katrs no šiem kadru pāriem sniedz apmierinošus rezultātus pielietojot attēlu sapludināšanas algoritmu. Tika izmantoti attēli no testa datu iegūšanas izmantojot trīs automašīnas, kadru pārus ņemot no pirmās un trešās automašīnas.

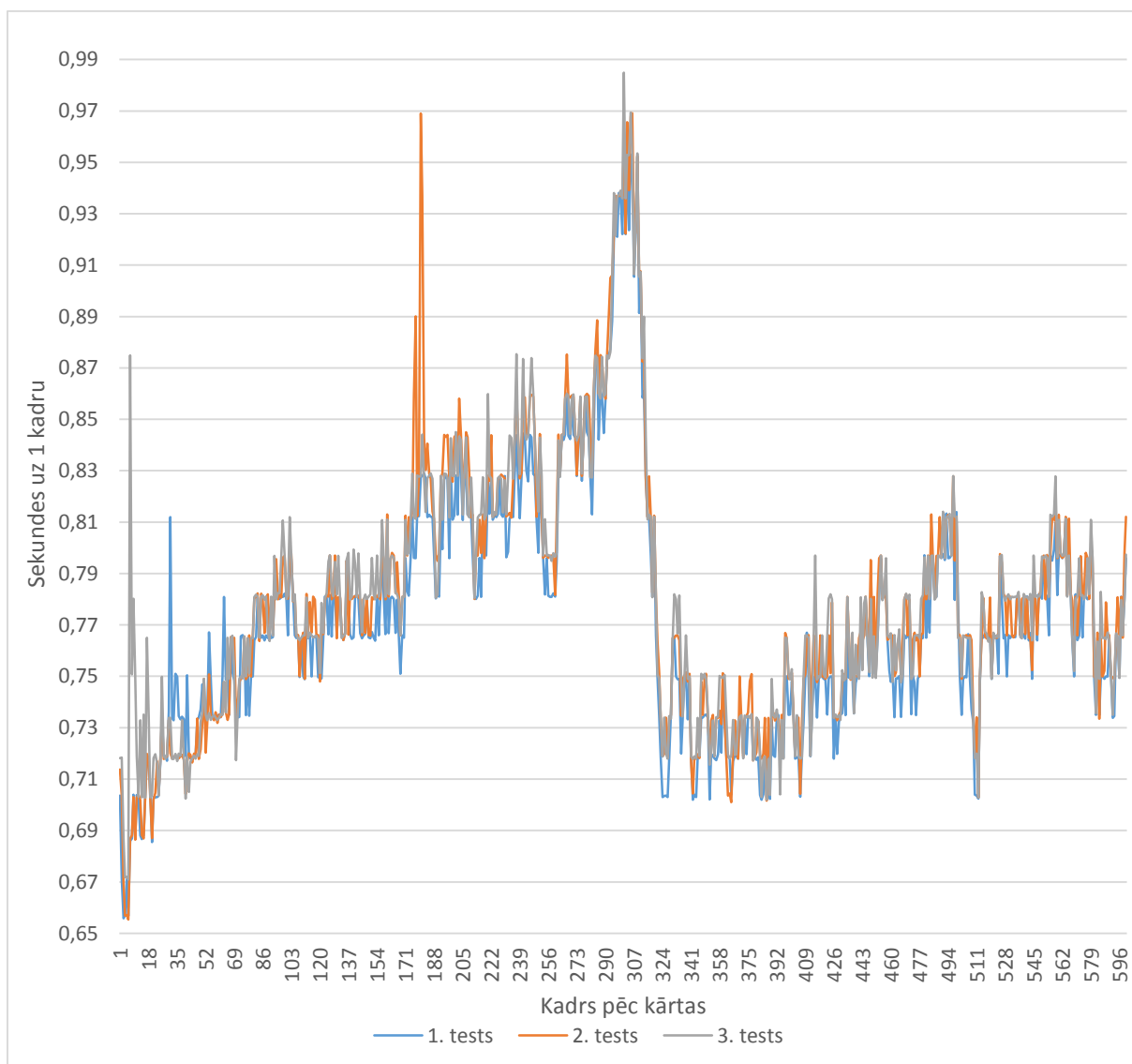
Lai veikspējas testi nebūtu ar pārāk lielu gadījuma raksturu tika veikti trīs testi, katrā testā fiksējot katra kadru pāra sapludināšanas laiku, kā arī informāciju par katrā kadrā atrastajiem zīmīgajiem punktiem un atsijātajiem derīgajiem atbilstošajiem punktiem abos kados. Jāpiebilst, ka testi tika veikti tikai attēlu sapludināšanas algoritmam uz jau skaidri

zināmiem diviem attēliem, kurus noteikti ir iespējams sapludināt, tāpēc nebija nepieciešams veikt pareizo attēlu atrašanu, ko veic pirms attēlu sapludināšanas.

Šī un arī visu pārējo testu veikšanai tika izmantots personālais dators ar sekojošu specifikāciju:

- Centrālais procesors: Intel i5-2500K, kurš darbojas ar takts frekvenci 4.621 GHz.
- Atmiņa: Crucial, apjoms 8GB, kopnes frekvence 1600MHz.
- Cietais disks: Kingston HyperX 3K SSD 120GB.

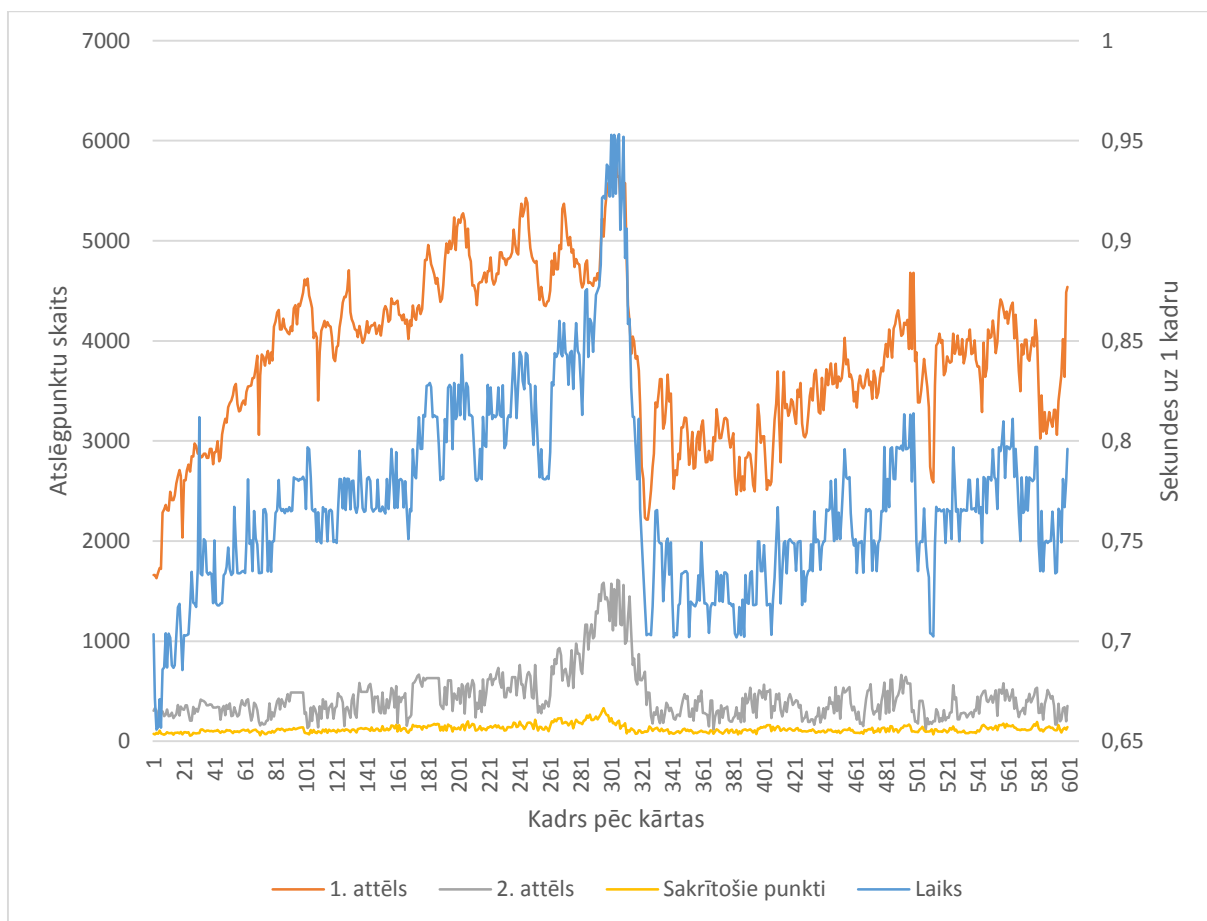
Visi testi tika darbināti izmantojot SSD disku, jo izmantotie video faili tika glabāti nesaspiesti, to izmēri bija ~1.6GB un ~3.6GB, līdz ar to, to nolasīšana no HDD aizņemtu pārāk daudz laika un iegūtie rezultāti nebūtu pietiekoši objektīvi attiecībā pret algoritma ātrdarbību.



3.4 att. Kadru iegūšanas laiks veiktajiem testiem

Veiktajos testos 600 kadru apstrādes ilgums bija attiecīgi 462.83s, 467.58s un 468.36s, kas dod vidējo rezultātu 466.26s, attiecīgi viena kadra apstrādes vidējais ilgums bija 0.777s. Viena kadra apstrādes minimālais ilgums bija 0.655s, maksimālais ilgums 0.985s. 3.4 att. redzams katra kadra iegūšanas ilgums trīs veiktajiem testiem, kā redzams neskaitot dažas nobīdes visas trīs līknes ir aptuveni vienādas, kas ļauj secināt, ka testu laikā nekāda liela ārēja efekta uz veiktspēju nav bijis. Jāpiebilst, ka izstrādātais algoritms strādā tikai vienā pavedienā, tāpēc šāds rezultāts arī bija gaidāms.

Pirmā testa rezultāti, salīdzinot kadra apstrādāšanas laiku ar atrasto nozīmīgo punktu skaitu abos attēlos un atrasto kopīgo punktu skaitu, redzami 3.5 att. Redzams, ka pastāv diezgan stingra korelācija starp šiem lielumiem, kas ļauj secināt, ka caurspīdīgā objekta iegūšanas algoritma veiktspēja ir tieši saistīta ar attēlos atrasto nozīmīgo punktu skaitu.

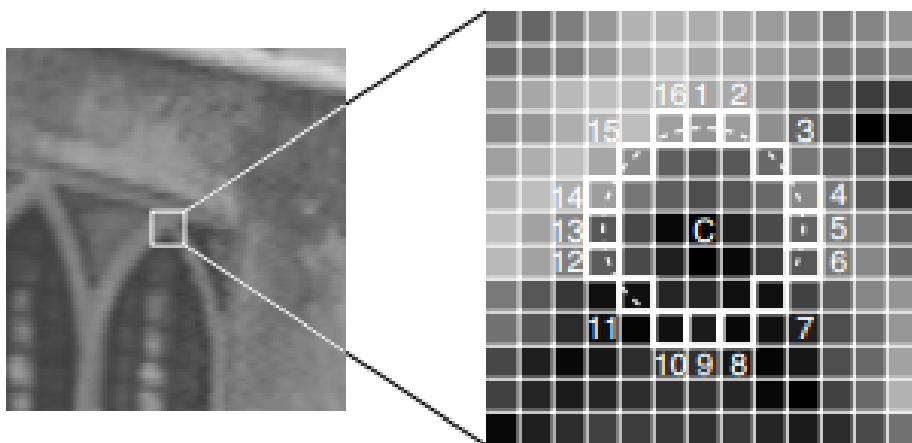


3.5 att. Kadra apstrādāšanas laiks salīdzinājumā ar punktu skaitu

Izdarot secinājumus par caurspīdīga objekta iegūšanu izmantojot tekošo C++ implementāciju autors secina, ka pašreizējā ātrdarbība, kas ir vidēji 0.777 sekundes uz vienu kadru nav apmierinoša, jo ar to nepietiek, lai spētu nodrošināt caurspīdīga objekta iegūšanu reālā laikā, turklāt šim laikam vēl jāliek klāt laiks, kas nepieciešams pareizo attēlu atrašanai šim sapludināšanas algoritmam.

3.2.2. Ātrdarbības uzlabošana

Pēc algoritma pārvešanas uz C++ programmēšanas valodu iegūtā ātrdarbība nebija apmierinoša, tāpēc tika nolemts SIFT algoritma nozīmīgo punktu iegūšanai vietā izmantot ORB algoritmu nozīmīgo punktu iegūšanai. ORB nozīmīgo punktu iegūšanas algoritmā zīmīgie punkti tiek aprakstīti ar bināriem skaitļiem, atšķirībā no SIFT algoritma, kurā tiek lietoti 128 dimensiju vektori. ORB algoritms principā sastāv no FAST(*Features from Accelerated Segment Test*)[21] stūru noteicēja un BRIEF(*Binary Robust Independent Elementary Features*)[22] punktu aprakstītāja, kuri ir stipri modificēti labākas ātrdarbības iegūšanai.

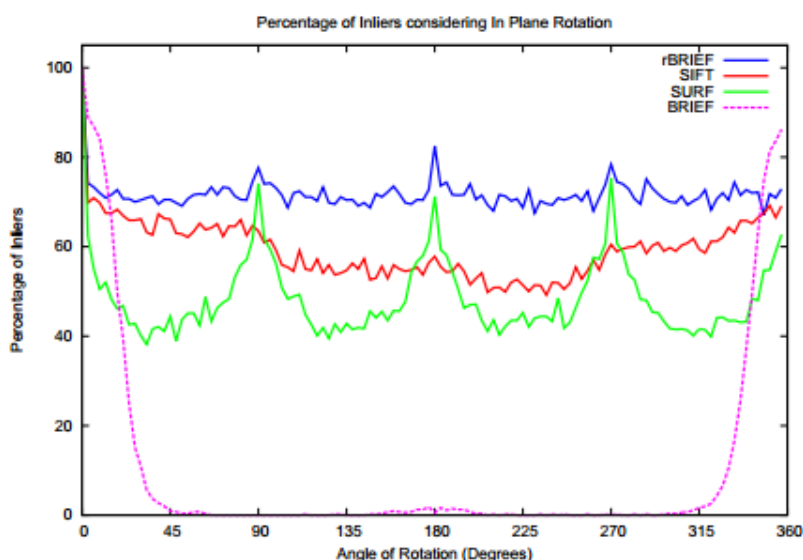


3.6 att. FAST algoritma darbības princips, attēls no avota [21]

FAST stūru noteicējs stūru noteikšanu veic pēc pilnās pārlasses principa pārbaudot katru pikseli attēlā. Kā redzams 3.6 att., katram pikselim tiek pārbaudīts, vai tas nav kāda stūra centra pikselis. Pārbaude notiek salīdzinot 3.6 att. norādītos 16 pikseļus, kas izvietoti aplī ap centra pikseli. 3.6 att. tiek izmantots Bresenham[23] aplis ar rādiusu 3, bet ORB implementācijā tiek izmantots Bresenham aplis ar rādiusu 9, jo tas sniedz labus rezultātus un veikspēju. 3.6 att. konkrētais centra pikselis tiek atzīts par stūra centra pikseli tad, ja vismaz 12 pēc kārtas esoši pikseļi aplī pēc intensitātes ir lielāki vai mazāki par centra pikseli par noteiktu sliekšni. Šāds piegājiens ļauj optimizēt stūru noteikšanu pārbaudot apla pikseļus ar numuriem 1, 9, 5 un 13, jo lai iepriekšminētais nosacījums izpildītos, vismaz 3 no šiem pikseļiem ir jāizpildās iepriekšminētajai sakarībai par punkta intensitāti attiecībā pret centra punktu.

FAST stūru noteicējs spēj darboties ļoti ātri, taču ORB vajadzībām tam ir viens liels trūkums, tas nesatur informāciju par stūra orientāciju, tāpēc ORB izstrādes gaitā tika izveidots FAST paplašinājums oFAST(*oriented* FAST), kas pievieno informāciju par stūra orientāciju izmantojot vienkāršu, bet efektīvu stūra orientācijas mērīšanas paņēmieni intensitātes centroīds[24], tā pamatideja ir pieņemt, ka stūra intensitāte ir nobīde no tā centra, tādējādi šis vektors var tikt izmantots intensitātes aprakstīšanai.

BRIEF punktu aprakstītājs ir attēla daļas aprakstīšana izmantojot bitu virknes, kas iegūtas no vairākiem bināriem intensitātes testiem. Šie testi tiek veikti katram testam izvēloties divus nejaušus pikselus un salīdzinot to intensitātes, salīdzināšanas rezultātu var aprakstīt ar vienu bitu, attiecīgi vai pirmā pikseļa intensitāte ir lielāka par otrā pikseļa intensitāti. ORB implementācija izmanto 256 šādus pikseļu pārus, attiecīgi katrs punkts tiek aprakstīts ar 256 bitiem. Protams, lai salīdzinātu dažādus punktus, testiem izvēlēto pikseļu koordinātas attiecībā pret centru visiem testiem ir vienādas.



3.7 att. Populārāko stūru aprakstīšanas algoritmu salīdzinājums atkarībā no attēla rotācijas leņķa, attēls no avota [18]

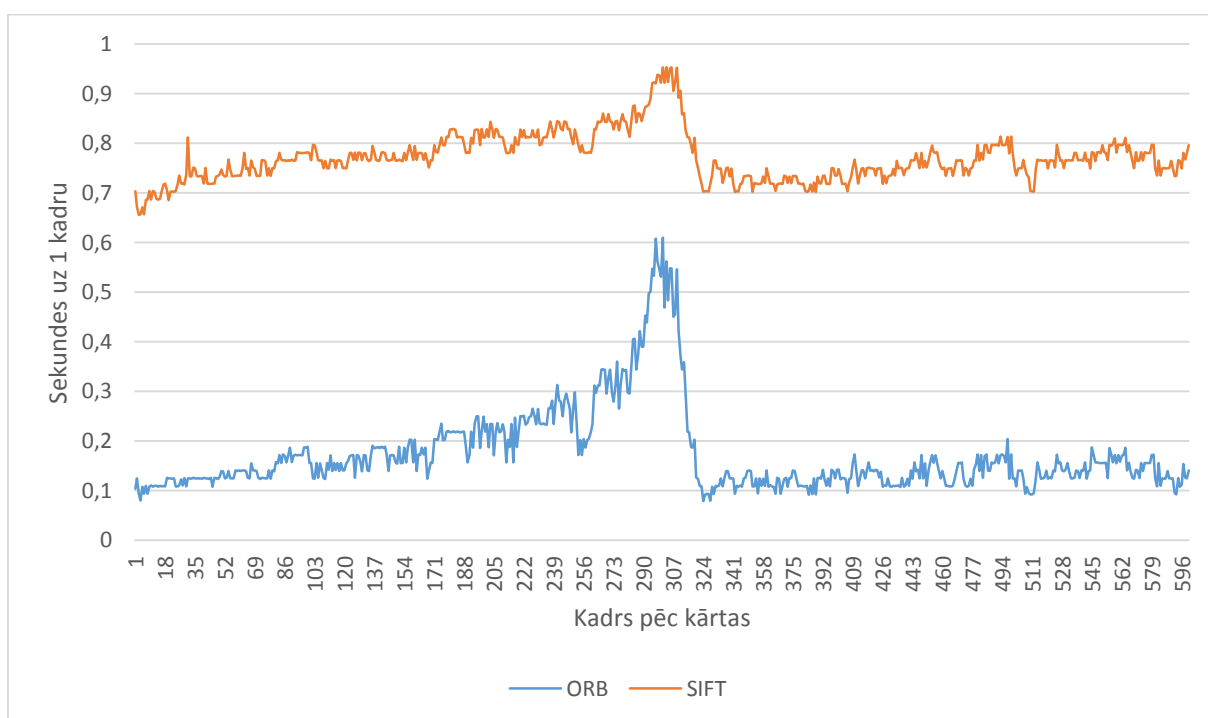
BRIEF stūru aprakstītājs spēj darboties ļoti ātri, taču ORB vajadzībām tam ir viens liels trūkums, ja attēls ir pagriezts par vairāk kā dažiem grādiem, tad attiecīgiem pikseļi, kas tiek izmantoti testiem vairs nav tie paši, tāpēc ļoti strauji krītas BRIEF algoritma precizitāte, tāpēc ORB izstrādes gaitā tika izveidots BRIEF paplašinājums rBRIEF(*rotation-aware* BRIEF), kas ir imūns pret attēla rotāciju. Galvenā rBRIEF ideja ir izvēlēties pikselus testiem attiecībā pret oFAST algoritma sniegto stūra orientāciju, tādējādi pat pēc rotācijas, visi testi tiks izpildīti uz tiem pašiem pikseļiem. 3.7 att. redzams salīdzinājums starp populārākajiem stūru aprakstīšanas algoritmiem un rBRIEF, kā redzams BRIEF algoritms vispār nav spējīgs

darboties pat pie nelielas rotācijas, bet rBRIEF algoritms darbojas stabili neatkarīgi no attēlam pielietotās rotācijas. Autora veidotajā caurspīdīgu objektu iegūšanas algoritmā noturība pret rotāciju ir būtiski svarīga, jo automašīnas pārvietojoties pa ceļu var svārstīties uz sāniem, attiecīgi iegūtie attēli var saturēt zināmu rotācijas līmeni.

3.2.3. Ātrdarbības novērtējums ar ORB metodi

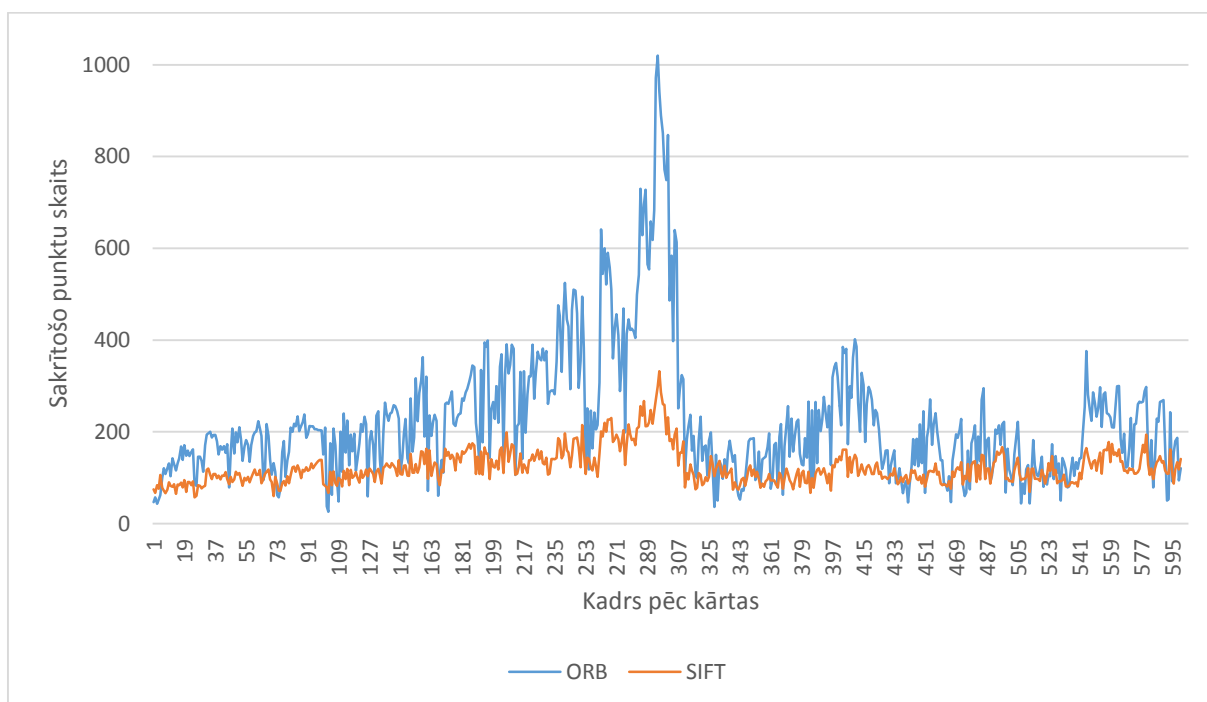
Pēc caurspīdīga objekta iegūšanas algoritma implementācijas uzlabošanas SIFT zīmīgo punktu noteicēja vietā izmantot ORB, tika veikts tieši tāds pats tests kā aprakstīts nodaļā 3.2.1., lai novērtētu vai pāreja no SIFT uz ORB ir devusi nepieciešamo veikspējas pieaugumu.

Veiktajos trīs testos ar ORB nozīmīgo punktu noteikšanas algoritmu 600 kadru apstrādes ilgums bija attiecīgi 105.52s, 105.84s un 104.54s, kas dod vidējo rezultātu 105.3s, attiecīgi viena kadra apstrādes vidējais ilgums bija 0.176s. Viena kadra apstrādes minimālais ilgums bija 0.077s, maksimālais ilgums 0.61s. ORB algoritma izmantošana caurspīdīga objekta iegūšanas algoritmā vidēji viena kadra apstrādē dod ātrdarbības pieaugumu par 4.41 reizi salīdzinot ar SIFT algoritma izmantošanu. 3.8 att. redzams katra kadra apstrādes laika salīdzinājums starp SIFT un ORB algoritmiem, tika izmantoti abu algoritmu veikspējas testu labākie rezultāti.



3.8 att. Kadru apstrādes laika salīdzinājums starp SIFT un ORB algoritmiem

Viens svarīgs aspekts, ko jāņem vērā SIFT vietā izmantojot ORB, vai nemainās iegūtā attēla ar caurspīdīgu objektu kvalitāte. Autora subjektīvais vērtējums salīdzinot ORB un SIFT algoritmus ir par labu ORB algoritmam, jo skatoties sapludināto video liekas, ka ORB izmantošana dod labākus rezultātus, bet lai objektīvi novērtētu vai šis veiktspējas ieguvums nav samazinājis iegūto sapludināto attēlu kvalitāti var izmantot atrasto sakrītošo punktu skaitu attēlos, jo vairāk sakrītošo punktu, jo lielāka varbūtība, ka sapludinātais attēls ir korekts. 3.9 att. redzams kadros atrasto sakrītošo punktu skaits SIFT un ORB algoritmiem. No 600 kadriem 530 kadros ORB algoritms dod vairāk sakrītošo punktu, kas ļauj secināt, ka arī iegūtajiem attēliem 530 no 600 kadriem jābūt labākiem izmantojot ORB algoritmu.



3.9 att. Sakrītošo punktu skaits kadros SIFT un ORB algoritmiem

3.3. Pareizo attēlu atrašana priekš sapludināšanas

Darba izstrādes gaitā autors pamanīja, ka attēlu sapludināšanas rezultāta korektums ir tieši saistīts ar to, cik labi ir sapludināšanai izmantotie attēli, šajā kontekstā labi attēli ir tādi attēli, kas uzņemti pēc iespējas tuvāk viens otram.

3.3.1. Pareizo attēlu priekš sapludināšanas īpašības

Lai noskaidrotu kādas īpašības saista tos divus attēlus, kuru sapludināšanas rezultātā tiek iegūts optimāls sapludinātais attēls ar caurspīdīgu objektu tika veikts sekojošs eksperiments. Sākumā tika atrasti divi attēli, kuru dotais sapludināšanas rezultāts ir tuvu optimumam, 3.10 att. redzams aizmugurē braucošās automašīnas uzņemtais kadrs, kuram eksperimenta laikā tiks mēģināts piemeklēt visatbilstošāko priekšā braucošās automašīnas uzņemto kadru. Tika izmantoti attēli no testa datu iegūšanas izmantojot trīs automašīnas, kadru pārus ņemot no pirmās un trešās automašīnas.



3.10 att. Aizmugurē braucošās automašīnas uzņemtais kadrs, kurš tiks izmantots pareizo attēlu priekš sapludināšanas meklēšanai

3.11 att. redzams vislabāk atbilstošais priekšā braucošās automašīnas uzņemtais kadrs 3.11 att. (a), kā arī sapludināšanas rezultāts 3.11 att. (b), kā redzams sapludināšana ir devusi ļoti labu rezultātu, jo attēla ģeometrijā nav iespējams saskatīt gandrīz nekādus defektus, protams krāsu gammu, kā arī apgaismojuma defekti eksistē, bet pašlaik autors fokusējas tieši uz attēla ģeometrijas pareizu attēlošanu. Lai arī redzamajā sapludināšanas rezultātā 3.11 att. (b) acīmredzamu ģeometrijas defektu nav, jāpiebilst, ka attēlu palielinot viss liekas nedaudz izplūdis, to visticamāk rada neliela ģeometrijas nobīde, visticamāk par dažiem pikseliem, kas ar aci nav pamanāma, bet veido izplūduša attēla efektu.



(a) Kadrs no priekšā braucošās automašīnas



(b) Sapludināšanas rezultāts

3.11 att. Vislabāk atbilstošais priekšā braucošās automašīnas uzņemtais kadrs un tā sapludināšanas rezultāts

Lai pierādītu, ka labāka kadra sapludināšanai ar konkrēto aizmugurē braucošās automašīnas uzņemto kadru nav, sapludināšana tika veikta ar kadriem ko uzņēmusi priekšā braucošā automašīna pirms un pēc 3.11 att. (a). Attiecīgi 3.12 att. redzams priekšā braucošās automašīnas iepriekšējais uzņemtais kadrs un tā sapludināšanas rezultāts un 3.13 att. redzams priekšā braucošās automašīnas nākošais uzņemtais kadrs un tā sapludināšanas rezultāts.



(a) Kadrs no priekšā braucošās automašīnas



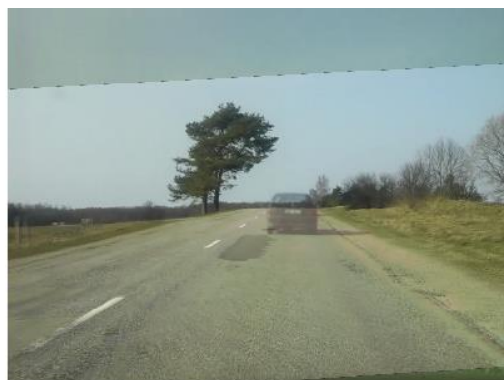
(b) Sapludināšanas rezultāts

3.12 att. Iepriekšējais kadrs pirms vislabāk atbilstošā priekšā braucošās automašīnas uzņemtā kadra un tā sapludināšanas rezultāts

Ja labi ieskatās 3.12 att. (b), redzams, ka uz ceļa redzamā tuvākā baltā līnija ir izkropļota, kā arī ceļa kreisajā pusē esošais stabs ir redzams dubultā. Šis arī nav slikts sapludināšanas rezultāts, bet tomēr sapludināšanas rezultāts 3.11 att. (b) ir labāks.



(a) Kadrs no priekšā braucošās automašīnas

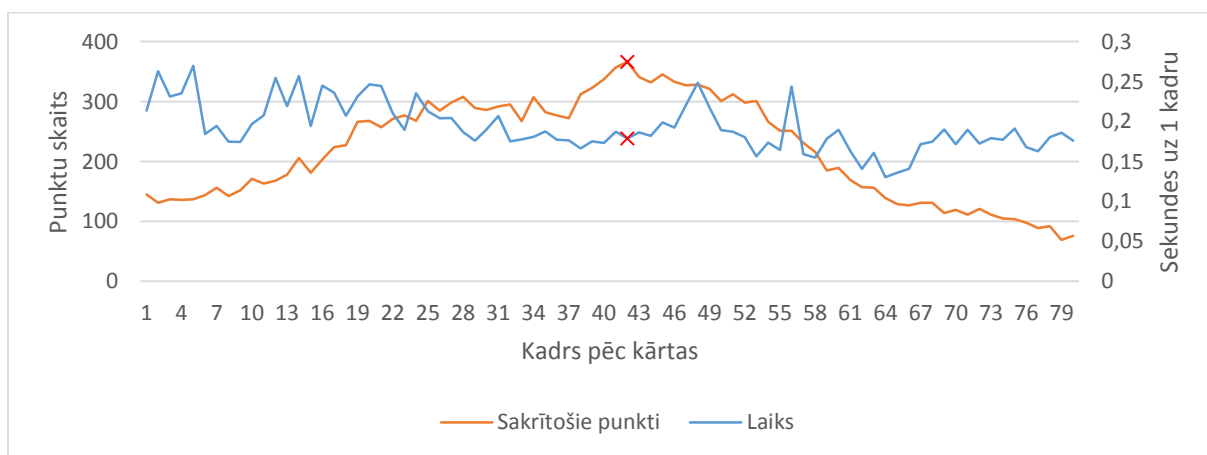


(b) Sapludināšanas rezultāts

3.13 att. Nākošais kadrs pēc vislabāk atbilstošā priekšā braucošās automašīnas uzņemtā kadra un tā sapludināšanas rezultāts.

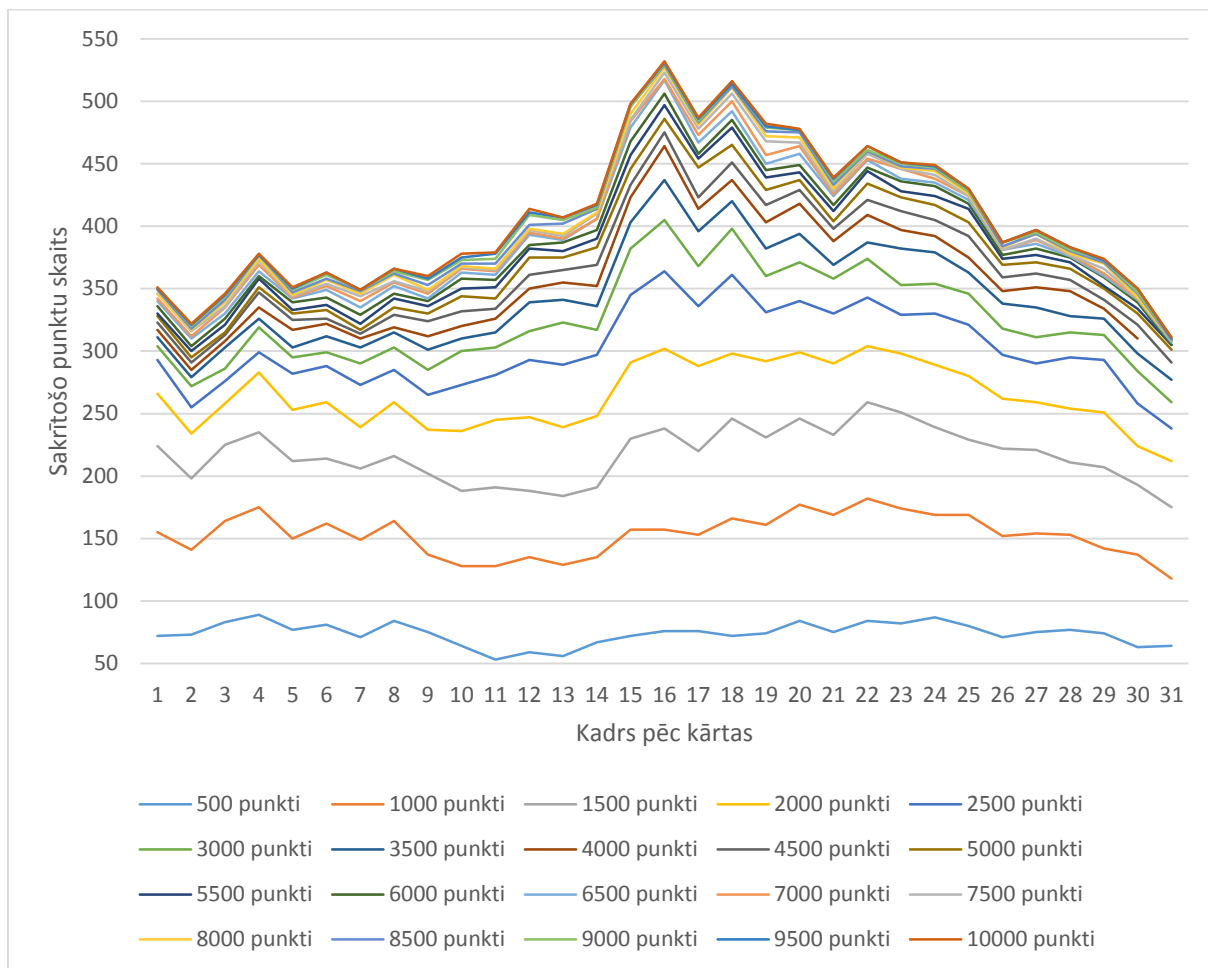
Ja labi ieskatās 3.13 att. (b), redzams, ka ceļa kreisajā pusē esošais stabs ir redzams dubultā. Šis arī nav slikts sapludināšanas rezultāts, bet tomēr sapludināšanas rezultāts 3.11 att. (b) ir labāks.

Tā kā ir atrasts kadrs no priekšā braucošās automašīnas, kurš sniedz vislabāko sapludināšanas rezultātu kadram no aizmugurē braucošās automašīnas, ir iespējams salīdzināt informāciju, kas iegūta attēlu sapludināšanas rezultātā, lai noskaidrotu vai kadru pārim, kas sniedz vislabāko sapludināšanas rezultātu ir kādas specifiskas īpašības, kas nav citiem kadru pāriem. Lai iegūtu šādu informāciju, 3.10 att. redzamais aizmugurē braucošās automašīnas uzņemtais kadrs tika sapludināts ar aptuveni 41 kadru, kas uzņemti pirms vislabāk atbilstošā priekšā braucošās automašīnas kadra, kā arī ar 38 kadriem pēc šī kadra. Iegūtais video aplūkojams tiešsaistē adresē http://youtu.be/a_sRnPR5Sjw. Katram kadru pārim sapludināšanas rezultātā tika pierakstīti sekojoši parametri: sakrītošo punktu skaits, kadra apstrādes laiks un informācija par aizmugurē uzņemtā kadra aizvietojamo apgabalu.



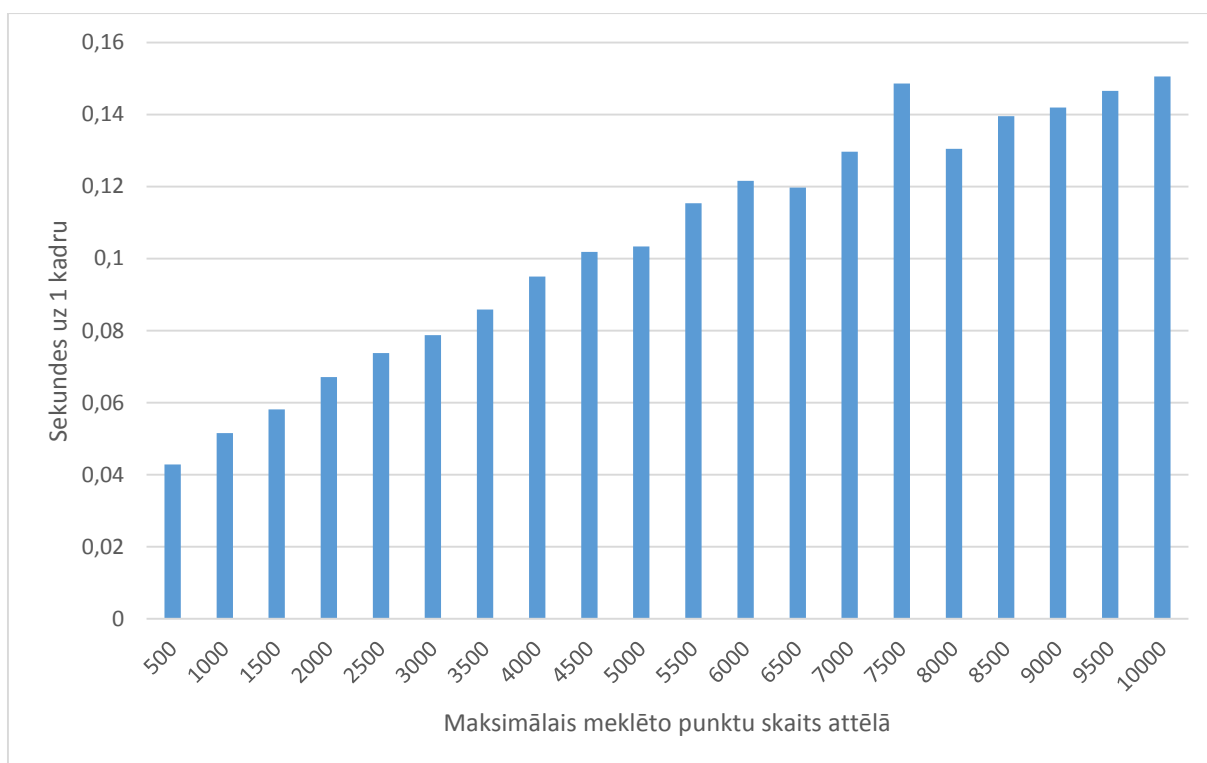
3.14 att. Sakrītošo punktu skaits un kadra apstrādes laiks secīgiem aizmugurē uzņemtajiem kadriem pret vienu priekšā uzņemto kadru. Ar sarkanu X atzīmēts tas kadrs, kurš ir uzskatāms par vislabāko sakrītību.

Analizējot iegūtos datus apstiprinājās autora aizdomas, ka sakrītošo punktu skaitam ir tieša sakarība ar iegūtā attēla kvalitāti, kā redzams 3.14 att., ar sarkanu x atzīmēts vislabāk atbilstošais kadrs un tas ir maksimuma punkts sakrītošo punktu skaitam. Arī redzams, ka kadra apstrādes laikam nav nekāda sakara ar to, cik labs ir iegūtais sapludinātais attēls.



3.15 att. Sakrītošo punktu skaits dažādiem maksimālo zīmīgo punktu skaitiem vienā attēlā

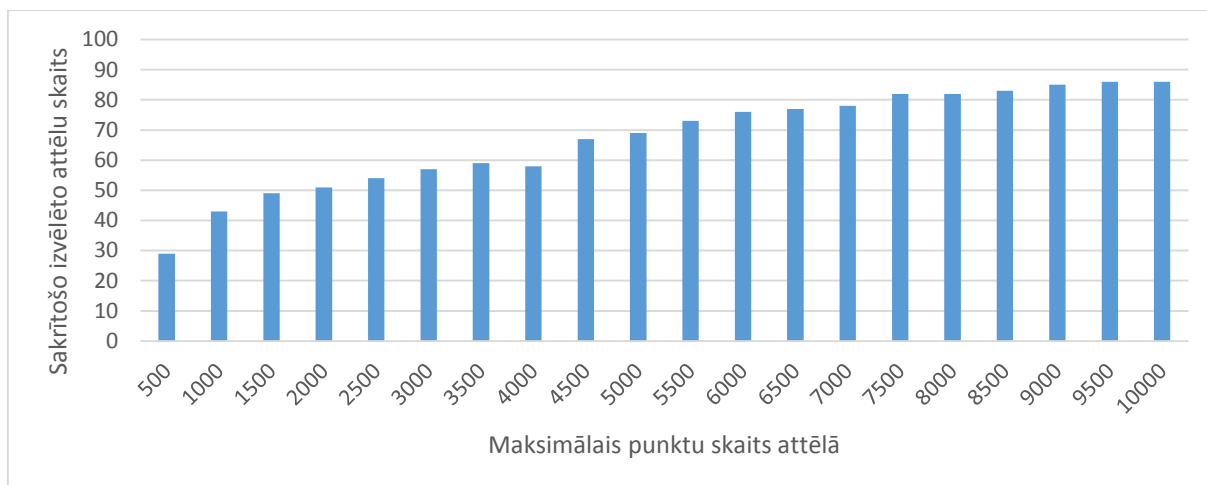
Tā kā 3.14 att. redzams, ka apstrādes laiks vienam kadram ir aptuveni 0.17 sekundes, tad autors vēlējās pārliecināties vai nav iespējams noteikt atbilstošo kadru izmantojot šo sakrītošo punktu skaitu, bet nemeklējot pilnīgi visus punktus abos attēlos. 3.15 att. redzams kā izmainās sakrītošo punktu skaits atkarībā no maksimālā punktu skaita kas tiek meklēts attēlos. 3.16 att. redzams viena kadra apstrādes laiks atkarībā no maksimālā zīmīgo punktu skaita kas tiek meklēts attēlos.



3.16 att. Viena kadra apstrādes laiks dažādiem maksimālo zīmīgo punktu skaitiem vienā attēlā

Salīdzinot 3.14 att. rezultātu labākajai atbilstībai, kur maksimālais punktu skaits attēlā nav ierobežots – 0.179s un 3.16 att. rezultātu labākajai atbilstībai ierobežojot maksimālo punktu, redzams, ka ja maksimālo meklēto punktu skaita tādus pašus rezultātus kā maksimālo punktu skaita neierobežošana, tad ietaupītais laiks būtu vērā ņemams.

Lai noskaidrotu vai ierobežojot maksimālo punktu skaitu ir iespējams iegūt tādus pašus labākos sakrītošos kadrus kā neierobežojot punktu skaitu autors veica eksperimentu, kurā tika noskaidrots vai ierobežojot maksimālo punktu skaitu ir iespējams iegūt tādus pašus rezultātus kā neierobežojot to attiecībā uz labākā attēlu pāra izvēli sapludināšanai. Tika izmantoti attēli no testa datu iegūšanas izmantojot trīs automašīnas, kadru pārus ņemot no pirmās un trešās automašīnas, pārbaudot izvirzīto hipotēzi uz 100 kadriem. Katram kadram kas uzņemts no aizmugurējās automašīnas tika mēģināts piemeklēt atbilstošo kadru no priekšā braucošās automašīnas uzņemtajiem kadriem.



3.17 att. Sakrītošo punktu skaits vislabāk atbilstošajam attēlam pie dažādiem maksimālo zīmīgo punktu skaitiem vienā attēlā

Eksperimentā iegūtie rezultāti redzami 3.17 att., kā redzams pat meklējot 10 000 punktus katrā attēlā, tikai 86 no 100 reizēm atrastais attēls ar visvairāk atbilstībām ir tas pats, kuram ir visvairāk atbilstības neierobežojot punktu skaitu attēlā. Vairāk kā 10 000 punktus attēlā meklēt vairs nav vērts, jo laiks, kas jāpatērē ar maksimālo punktu skaitu 10 000 ir tikai nedaudz mazāks nekā laiks, kas jāpatērē meklējot punktus bez skaita ierobežojuma – aptuveni 0.15s pret 0.179s jeb 16%. 16% ir slikts rezultāts, jo, ja uz katru pārbaudi tiek ietaupīti 16% laika ir nepieciešamas vismaz 7 pārbaudes, lai ietaupītais laiks pārsniegtu 100%, ja netiek pārsniegti 100% uz visām pārbaudēm, tad kopsummā laiks ir pazaudēts, jo ir jāveic vēl viena zīmīgo punktu atrašana izmantojot neierobežotu punktu skaitu, turklāt vērojot rezultātu tendenci diez vai apmierinošs rezultāts tiks sasniegts izmantojot 10 500 vai 11 000 punktus, toties patērētais laiks gan strauji tuvojas laikam, kas nepieciešams lai atrastu visus punktus. Autors bija cerējis, ka labus rezultātus izdosies sasniegt izmantojot 4000 – 5000 kā maksimālo punktu skaitu vienā attēlā, tad ātrdarbības uzlabojums būtu vērā ņemams – aptuveni 0.07s jeb 60% uz katru pārbaudi.

3.3.2. Video ar GPS koordinātām

Lai atvieglotu pareizo attēlu atrašanu priekš sapludināšanas veiktajā ieejas datu iegūšanas braucienā tika ievākti arī GPS dati. Katrs viedtālrunis piefiksēja savu atrašanās vietu reizi sekundē, sākot no video uzņemšanas sākuma. Lai arī ideja liekas ļoti laba, tomēr realitātē ar viedtālruņa GPS sistēmu nepietiek, jo tā nav pietiekoši precīza, lai spētu viennozīmīgi atjaunot katra kadra uzņemšanas vietu.



3.18 att. Viedtālrunu fiksētās GPS koordinātes testa datu iegūšanas laikā

Kā redzams 3.18 att. (a), pa lielam protams visi punkti atrodas vienuviet, izņemot dažus viedtālruna HTC EVO 3D punktus, kas GPS signāla trūkuma dēļ ņemti no GSM tīkla, bet 3.18 att. (b) skaidri redzams, ka atsevišķu punktu precizitāte ir ļoti slikta. Autors uzskata, ka no šādiem datiem nav iespējams atjaunot precīzu kadru uzņemšanas pozīciju. Lai iegūtu labākus datus nāktos izmantot precīzākas GPS iekārtas, kas diemžēl autoram pašlaik nav pieejamas.

3.4. Automātiska sapludinātā attēla korektuma pārbaude

Tā kā caurspīdīga objekta iegūšanas algoritms ne vienmēr spēj sniegt derīgu rezultātu, tad ir nepieciešams implementēt funkcionalitāti, kas spēj pārbaudīt to, cik labi ir veikta attēlu sapludināšana. Tā kā jāņem vērā vairāki faktori, kā piemēram apgaismojums, ievietotā attēla korektums, u.c., tad šis nav nemaz tik vienkāršs uzdevums. Autors izvirza 4 kritērijus, kas varētu kalpot par nosacījumiem sapludinātā attēla korektuma pārbaudei:

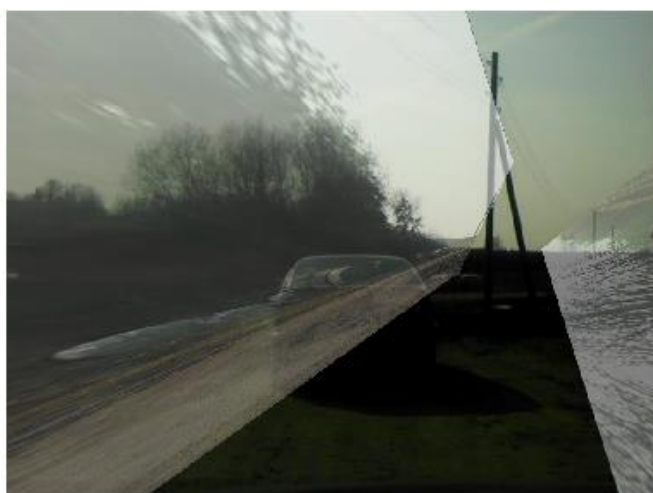
- Aizvietotā laukuma forma;
- Aizvietotā laukuma izmērs;
- Histogrammas;
- Sakrītošo punktu skaits.

3.4.1. Aizvietotā laukuma forma

Pēc perspektīvās transformācijas attēlam, ko novieto objekta vietā, lai iegūtu caurspīdīguma efektu, jābūt četrstūra formā. Šis nosacījums ļauj ātri un vienkārši atsijāt tos gadījumus, kad perspektīvās transformācijas rezultātā ir iegūts kropļots attēls, kas izkropļots tik tālu, ka nav vairs iespējams saprast, kas tajā attēlots. Šāda izkropļota attēla piemērs dots 3.19 att., 3.19 att. (a) ar sarkanu iekrāsots apgabals, kas netiek aizvietots, kā redzams aizvietotais apgabals sastāv no divām formām, attiecīgi arī iegūtais sapludināšanas rezultāts 3.19 att. (b) ir izkropļots, turpretim 3.20 att. (a) redzams, ka aizvietotais apgabals sastāv no vienas formas – četrstūra un kā redzams sapludināšanas rezultāts 3.20 att. (b) ir korekts attēls ar caurspīdīgu priekšā braucošo automašīnu.



(a) Aizvietotā laukuma forma



(b) Sapludināšanas rezultāts

3.19 att. Slihts aizvietotā laukuma formas piemērs un iegūtais sapludinātais attēls



(a) Aizvietotā laukuma forma



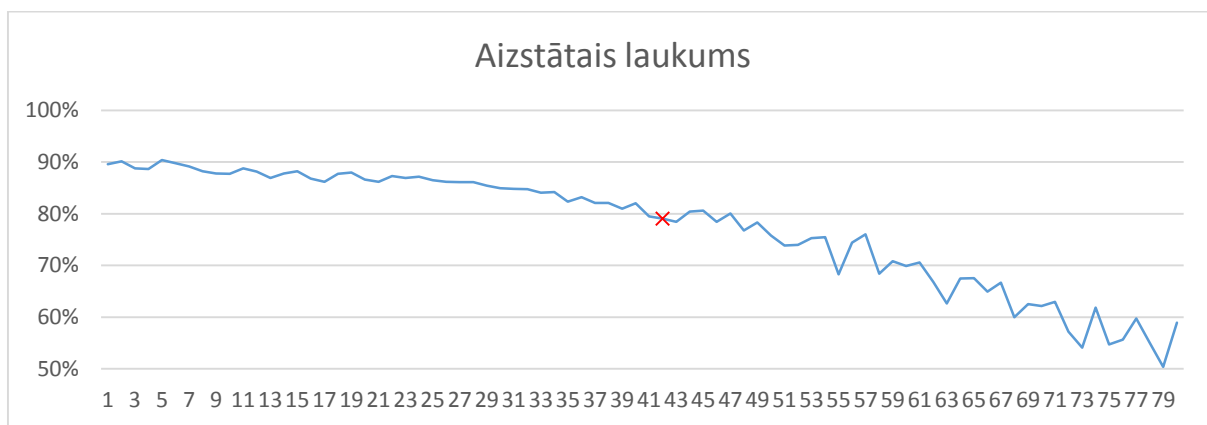
(b) Sapludināšanas rezultāts

3.20 att. Labs aizvietotā laukuma formas piemērs un iegūtais sapludinātais attēls

Lai pārliecinātos, ka aizvietotā laukuma forma ir četrstūris, attēlam ar aizvietotā laukuma formu tiek saskaitītas tajā atrastās kontūras, ja tajā eksistē tikai viena kontūra, tad tas visticamāk ir četrstūris, ja eksistē vairāk kā viena kontūra, tad iegūtā forma ir kaut kas cits.

3.4.2. Aizvietotā laukuma izmērs

Šis nosacījums nodrošina pret pārāk liela vai pārāk maza laukuma aizvietošanu. Aizvietojojam pārāk lielu vai pārāk mazu laukumu ir liela iespējamība, ka sapludinātais attēls ir nekorekts un parasti tas sniedz sliktus rezultātus. 3.21 att. redzami nodaļā 3.3.1. aprakstītā eksperimenta iegūtie dati par aizvietotā laukuma procentuālo izmēru attiecībā pret visu laukumu, kā redzams vislabāk atbilstošie kadri aizvietoja 79% no laukuma. Priekšējās automašīnas nākamajiem uzņemtajiem kadriem aizvietotais laukums samazinās, jo tajos ir redzams mazāks laukums nekā izmantotajā aizmugurējās automašīnas uzņemtajā kadrā.



3.21 att. Aizvietotā laukuma procentuālais izmērs attiecībā pret visu laukumi no 3.3.1. nodaļā aprakstītā eksperimenta. Ar sarkanu X atzīmēts tas kadrs, kurš ir uzskatāms par vislabāko sakrītību.

Šeit ir jāņem vērā abu kameru pozīcija, jo ja kameras nav pavērstas vienā virzienā, tad aizvietotā laukuma izmērs var variēt tieši šī iemesla dēļ, nevis tāpēc, ka konkrētais sapludinātais attēls būtu korekts vai tieši otrādi nekorekts, tāpēc autors uzskata, ka būtu saprātīgi noteikt aizvietotā laukuma izmēra atļautās robežas no 10% līdz 90%.

3.4.3. Histogrammas

Histogrammu salīdzināšana starp oriģinālo un sapludināto attēlu varētu sniegt kaut kādu priekšstatu par to cik labi ir notikusi attēlu sapludināšana. Šis paņēmieni balstās uz attēlu salīdzināšanu balstoties uz kādu noteiktu attēla īpašību, kā piemēram nokrāsa vai

piesātinājums. Histogrammas rēķināšanai tiek izmantots tikai tas attēla apgabals, kurš iegūts pēc perspektīvās transformācijas, tādējādi ignorējot to attēla daļu, kas abiem attēliem sakrīt, lai iegūtu lielāku precizitāti. Diemžēl šis paņēmiens nesniedz viennozīmīgus rezultātus, visticamāk tāpēc, ka histogrammā izmantotajiem parametriem nav nekāda sakara ar attēla ģeometrisko korektumu, šī metode varētu tikt izmantota, ja abu attēlu īpašības sakristu, piemēram, ja abiem attēliem nokrāsa un piesātinājums būtu vienādi, tad ar histogrammas palīdzību varētu noteikt par cik šīs īpašības ir mainījušās konkrētajā apgabalā, tādējādi iegūstot kaut kādu priekšstatu par to, cik korekta ir attēla ģeometrija.

3.4.4. Sakrītošo punktu skaits

Ņemot vērā nodaļā 3.3.1. aprakstītos rezultātus var diezgan droši apgalvot, ka attēlu sapludināšanas korektumu var novērtēt izmantojot sakrītošo punktu skaitu, jāatzīst gan, ka šis novērtējums var kalpot tikai kā relatīvs salīdzinājums starp sapludinātajiem attēliem, lai noteiktu kuram no tiem ir lielāka iespējamība būt korektam sapludinātam attēlam. Šo kritēriju var izmantot, lai noteiktu kuru no priekšā braucošās automašīnas uzņemtajiem kadriem izmantot sapludināšanai ar konkrēto aizmugurē braucošās automašīnas kadru, bet to nevar izmantot, lai salīdzinātu vairākus sapludinātos attēlus, kuros vismaz viens ieejas kadrs nesakrīt, jo sakrītošo punktu skaitam ir tieša saikne ar izmantotajiem kadriem.

3.4.5. Korektuma pārbaudes algoritms

Lai noteiktu vai attēls ir sapludināts korekti tiek izmantoti divi no iepriekš aprakstītajiem četriem nosacījumiem. Sākumā tiek noskaidrots vai aizvietotā laukuma forma atbilst četrstūrim un vai aizvietotais laukums procentuāli no visa attēla sastāda no 10% līdz 90%. Ja šie nosacījumi izpildās, tad sapludinātais attēls tiek atzīts par labu. Jāpiebilst, ka sakrītošo punktu skaits tiek izmantots lai noteiktu kurus attēlus izmantot priekš attēlu sapludināšanas, tāpēc nav vērts izmantot sakrītošo punktu skaitu kā pārbaudes kritēriju.

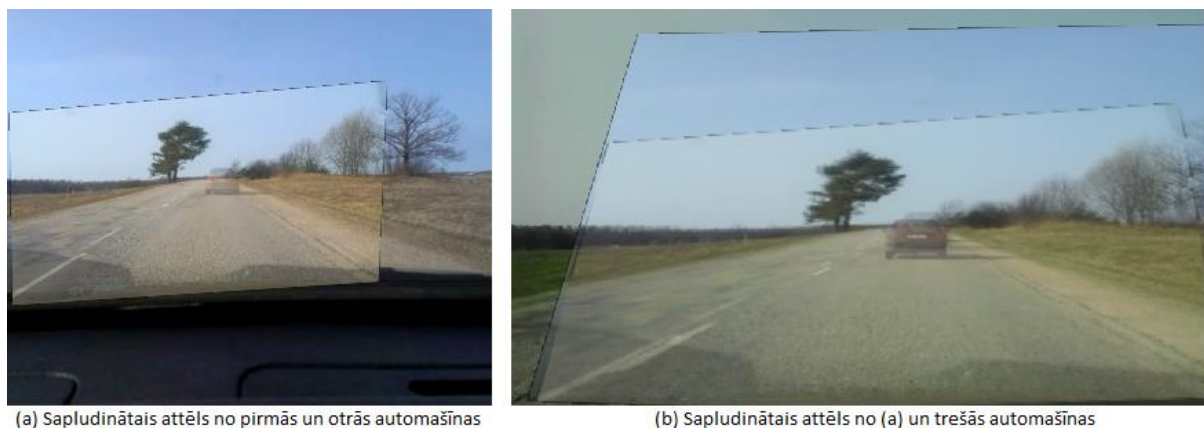
3.5. Eksperimenti un novērtējums

Lai novērtētu cik labi ir izdevies izstrādāt otro prototipu caurspīdīgas automašīnas iegūšanai autors veic vairākus eksperimentus. Diemžēl autoram nav izdevies panākt automātisku iegūtā sapludinātā attēla ģeometrijas korektuma pārbaudi, kas ļoti apgrūtina lielu eksperimentu veikšanu un padara iegūtos rezultātus savā ziņā neobjektīvus, jo katrs iegūtais sapludinātais attēls autoram ir jānovērtē subjektīvi.

3.5.1. Eksperiments ar attēliem no trīs pēc kārtas braucošām automašīnām

Autors vēlas pārbaudīt, vai sapludinātais attēls, kas iegūts izmantojot izstrādāto algoritmu ir lietojams kā attēls no priekšā braucošās automašīnas, ko izmantot vēl viena sapludināta attēla iegūšanai. Tika izmantoti attēli no testa datu iegūšanas izmantojot trīs automašīnas.

Eksperimenta sākumā tika piemeklēti divi attēli no pirmās un otrās automašīnas, kas sniegtu labu sapludināto attēlu, iegūtais sapludinātais attēls redzams 3.22 att. (a).



3.22 att. **Sapludinātais attēls(a) un iegūtais sapludinātais attēls(b), kas iegūts kā vienu no attēliem izmantojot sapludināto attēlu (a)**

Pēc tam izmantojot šo attēlu kā attēlu no priekšā braucošās automašīnas tika piemeklēts atbilstošs attēls no trešās automašīnas ar ko veikt attēlu sapludināšanu, iegūtais rezultāts redzams 3.22 att. (b). Iegūtajā sapludinātajā attēlā redzams, ka sapludināšanas rezultāts nav izdevies īpaši veiksmīgs, attēlā eksistē salīdzinoši daudz ģeometrijas defektu, piemēram, baltajās līnijās, kas atrodas uz ceļa nesakrīt, kā arī ceļa kreisajā pusē esošā koka siluete nesakrīt. Bet pēc autora domām pats svarīgākais ir tas, ka skatoties uz divām

caurspīdīgām mašīnām, kas novietotas viena aiz otras nav iespējams saprast kas tur vispār atrodas, tāpēc šāds rezultāts nav pieņemams un autors uzskata, ka šādā situācijā labāk attēlu sapludināšanai lietot attēlus no pirmās un trešās automašīnas, kā redzams 3.23 att. izmantojot attēlus no pirmās un trešās automašīnas caurspīdīgas ir abas priekšā braucošās automašīnas, ja tās abas ir redzamas aizmugurē braucošās automašīnas uzņemtajos attēlos.



3.23 att. Divas priekšā braucošas caurspīdīgas automašīnas

3.5.2. Attēlu sapludināšana izmantojot attēlus no vienādām ierīcēm

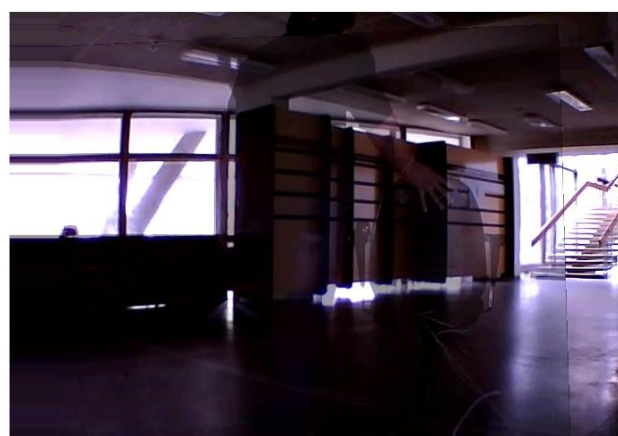
Lai noskaidrotu vai izmantojot vienādas ierīces attēlu uzņemšanai iegūtais sapludināšanas rezultāts ir būtiski labāks par rezultātu izmantojot dažādas ierīces, autors veica divus eksperimentus ar attēliem, kas uzņemti izmantojot vienādas ierīces.

3.5.2.1 Tīkla kameru iegūto attēlu sapludināšana

Šajā eksperimentā tika izmantoti testa dati, kuru iegūšana aprakstīta nodaļā 3.1.2. Iegūtie sapludināšanas rezultāti nav tik labi, kā autors bija cerējis, dažus no iegūtajiem rezultātiem var aplūkot 3.24 att., kā redzams 3.24 att. (a) eksistē vairākas ģeometriskas deformācijas, piemēram virs krēsla redzamā loga augšējā daļa ir stipri nobīdīta un aiz krēsla redzamā līnija uz zemes arī ir nobīdīta. Toties 3.24 att. (b) nav saskatāmas ģeometriskas deformācijas, bet iegūtais sapludinātais attēls ir nedaudz izplūdis, tātad attēlā ir neliela ģeometrijas nobīde. Ļoti daudziem attēliem vispār nav iespējams piemeklēt atbilstošu attēlu ar kuru veidot sapludināto attēlu.



(a) Sapludināšanas piemērs I



(b) Sapludināšanas piemērs II

3.24 att. Daži no iegūtajiem sapludinātajiem attēliem izmantojot attēlus no tīkla kamerām



3.25 att. Tīkla kameru radītā attēla kropļojuma piemērs

Izvērtējot šo eksperimentu autors secina, ka iegūtie sapludinātie attēli nav slikti, bet tie arī nav labāki par sapludinātajiem attēliem, kas iegūti izmantojot dažādas attēlu uzņemšanas ierīces. Autors to skaidro ar izmantoto tīkla kameru slikto uzņemto attēlu kvalitāti, testa datu

iegūšanas vides vājo apgaismojumu, kā arī tīkla kameru radīto attēlu kropļojumu ko var redzēt 3.25 att., skatoties uz durvīm labajā pusē ļoti labi redzams, ka tās ir nevis taisnas, bet izliektas, no kā secināms, ka pirms šo attēlu apstrādes nepieciešams veikt tīkla kameru iegūto attēlu kalibrēšanu, ko autors izvēlas neveikt, jo ir pieejami labāki testa dati, ko izmantot izvirzītās hipotēzes par labākiem sapludinātajiem attēliem izmantojot vienādas attēlu uzņemšanas ierīces pārbaudei.

3.5.2.2 Vienādu viedtālruņu iegūto attēlu sapludināšana

Šajā eksperimentā tika izmantoti testa dati, kuru iegūšana aprakstīta nodaļā 3.1.3. Izmantojot divus vienādus viedtālruņus iegūtie testa dati sniedz ļoti labus rezultātus. Neskatoties uz to, ka testa datu iegūšanas dienā bija lietains laiks un testa datos redzamas gan ūdens lāses uz automašīnas priekšējā vējstikla, gan automašīnas logu tīrīšanas mehānisms, gandrīz visiem attēliem ir iespējams piemeklēt atbilstošo attēlu ar kuru veidot pieņemamu sapludināto attēlu.



3.26 att. **Daži no iegūtajiem sapludinātajiem attēliem izmantojot attēlus vienādiem viedtālruņiem**

3.26 att. (a) redzams, ka caur iegūto caurspīdīgās automašīnas attēlu redzama pretējā virzienā braucošā automašīna, kura nav redzama attēlā, kas uzņemts no aizmugurē esošās automašīnas, 3.26 att. (b) redzams, ka iegūts pieņemams attēlu sapludināšanas rezultāts neskatoties uz to, ka attēlos redzamas abu automašīnu logu tīrīšanas mehānismi. Šie rezultāti autoram ļauj secināt, ka attēlu sapludināšanas rezultātā ir iegūstami kvalitatīvāki sapludinātie attēli, ja tiek izmantotas vienādas attēlu iegūšanas ierīces.

3.5.3. Video ar caurspīdīgu automašīnu iegūšana

Lai noskaidrotu vai izstrādātais algoritms sniedz pietiekoši labus rezultātus uz vairāk kā dažiem attēliem autors izveidoja aplikāciju, kas mēģina apvienot divus video lai iegūtu vienu, kurā redzama caurspīdīga priekšā braucošā automašīna.

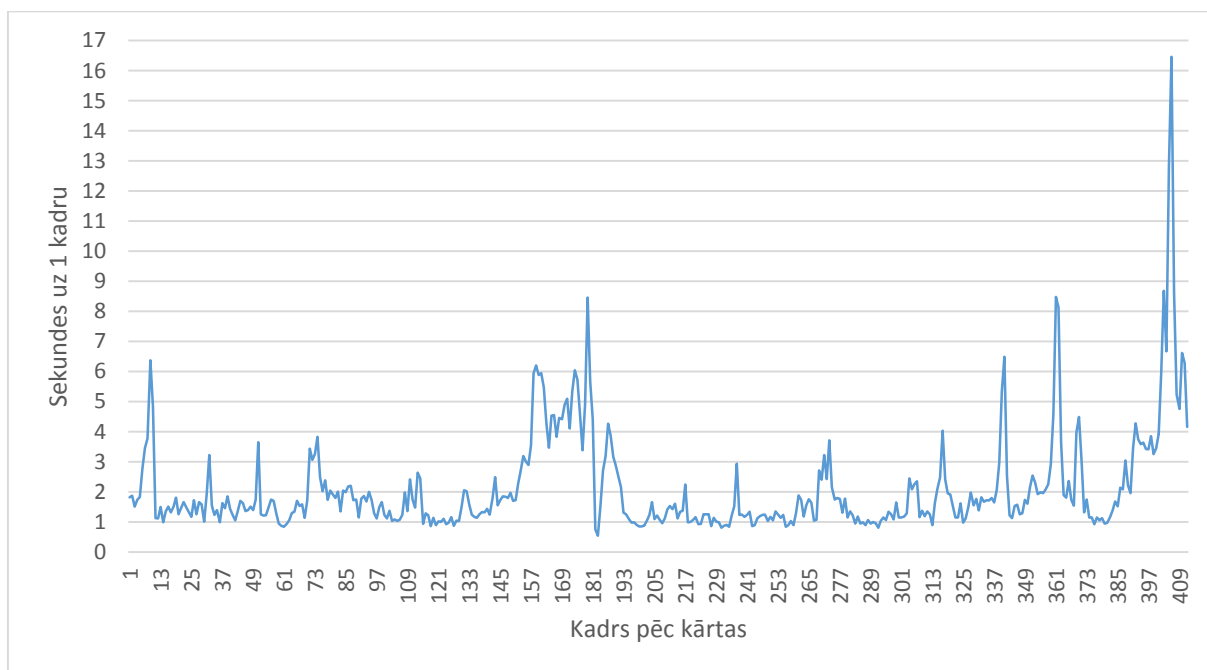
3.5.3.1 Video ar vienu kadru sekundē

Lai noskaidrotu, vai autora izstrādātais prototips spēj apstrādāt divus video un sniegt informāciju par to, kas atrodas aiz priekšā braucošās automašīnas vismaz reizi sekundē, tika izveidota aplikācija, kas apvieno divus video no aizmugurē braucošās automašīnas uzņemtā video ņemot kadru, kas uzņemts ik pēc sekundes un mēģinot tam piemeklēt atbilstošo kadru no priekšā braucošās automašīnas.

Izstrādātais algoritms kadra piemeklēšanai darbojas sekmīgi tikai un vienīgi tad, ja algoritma darbības sākumā abos video mašīnas atrodas aptuveni vienā un tai pašā vietā, vēlams, lai priekšā braucošā automašīna atrastos nedaudz aiz aizmugurē braucošās automašīnas. Algoritms izmanto sekojošu pieeju – abi video tiek tīti uz priekšu pa sekundi, attiecīgajam aizmugurē braucošā auto uzņemtajam kadram tiek meklēta labākā atbilstība starp 10 priekšā braucošā auto uzņemtajiem kadriem. Priekšā braucošā auto uzņemtais video tiek tīts uz priekšu pa vienu sekundi skaitot no pēdējā izmantotā kadra sapludināšanai, tādējādi kompensējot automašīnu attālināšanos vai pietuvināšanos savā starpā.

Izmantojot video no testa datu iegūšanas izmantojot trīs automašīnas, kadru pārus ņemot no pirmās un trešās automašīnas iegūtie rezultāti attēlu sapludināšanas ziņā bija samērā apmierinoši, no 412 sapludinātajiem kadriem divus kadrus par nederīgiem atzina nodaļā 3.4. aprakstītās metodes korektu sapludināto attēlu pārbaudei, vēl septiņus kadrus par absolūti nederīgiem atzina autors aplūkojot tos, tie sniedz ļoti izkropļotu attēlu, tāpēc tie arī uzskatāmi par nederīgiem. Atlikušos 403 kadrus autors uzskata par samērā labiem, protams starp tiem ir vairāki kadri, kuros skaidri saskatāmas ģeometriskas deformācijas, bet eksistē arī tādi kadri, kur ģeometriskas deformācijas saskatīt autoram neizdevās. Iegūtais sapludināto attēlu video aplūkojams tiešsaistē adresē <http://youtu.be/3AhJF721V3s>.

Diemžēl ar ātrdarbību rezultāti nav tik iepriecinoši, kopējais šo 412 kadru apstrādes laiks bija 870.9s, kas vidēji nozīmē 2.11s uz katru kadru. Ātrākais kadra iegūšanas laiks bija 0.54s, lēnākais 16.46s, 3.27 att. redzami visu kadru iegūšanas laiki.



3.27 att. Kadru iegūšanas laiki izmantojot video no testa datu iegūšanas izmantojot trīs automašīnas, kadru pārus ņemot no pirmās un trešās automašīnas uzņemtajiem video

Viens no veidiem kā uzlabot ātrdarbību ir samazināt kadru skaitu starp kuriem priekšā braucošā auto uzņemtajā video tiek meklēts atbilstošais kadrs aizmugurē braucošā auto uzņemtajam kadram, piemēram iepriekš izmantoto 10 kadru vietā izmantojot tikai 8 kadrus, 412 kadru apstrāde aizņem par 144,21s mazāk, tātad 726.69s, ļaujot vidēji vienu kadru iegūt 1.76s, bet šajā gadījumā automātiski kā nederīgi tiek atlasīti jau 9 kadri un autors aplūkojot video saskatīja vēl vairāk nepieņemamus kadrus, tāpēc autors uzskata, ka šādā veidā uzlabot ātrdarbību nav iespējams.

Izmantojot video no testa datu iegūšanas izmantojot divas automašīnas ar vienādām attēlu uzņemšanas ierīcēm iegūtie dati ir daudz labāki, 116 kadri tika iegūti 116.08s, sniedzot vidējo rezultātu tieši 1s uz kadru. Ātrākais kadra iegūšanas laiks bija 0.35s, lēnākais 4.5s, 3.28 att. redzami visu kadru iegūšanas laiki. Šādu ātrdarbības atšķirību starp eksperimentiem autors skaidro ar to, ka izmantojot vienādas attēlu iegūšanas ierīces kopējais pikseļu skaits abos attēlos ir 768 000 pikseļi, bet pirmajā eksperimentā izmantoto testa datu kopējais pikseļu skaits abos attēlos ir gandrīz divreiz lielāks – 1 398 000 pikseļi. Iegūtais sapludināto attēlu video aplūkojams tiešsaistē adresē <http://youtu.be/8JeQWNbeNVE>.



3.28 att. Kadru iegūšanas laiki izmantojot video no testa datu iegūšanas izmantojot divas automašīnas ar vienādiem viedtālruņiem

Nodaļā 3.4. aprakstītās metodes korektu sapludināto attēlu pārbaudei neatrada nevienu nederīgu kadru starp šiem 116 kadriem, bet autors apskatoties iegūtos sapludinātos attēlus divus no attēliem atzīst par nederīgiem.

3.5.3.2 Video ar vairākiem kadriem sekundē

Lai noskaidrotu kā izskatītos video, kur reālā laikā būtu redzama caurspīdīga automašīna katrā kadrā, autors izveidoja aplikāciju, kas izveido šādu video izmantojot testa datus, kas iegūti no divām automašīnām, kas aprīkotas ar vienādām attēlu uzņemšanas ierīcēm. Protams šāda video iegūšana nebūt nenotiek reālā laikā, 38.62s video iegūšana ar 29 kadriem sekundē aizņēma 1577s, t.i. 1.4s uz vienu kadru. Iegūtajā video, kā jau sagaidāms ir redzama caurspīdīga automašīna, tomēr pēc autora domām izmaiņas starp kadriem ir pārāk lielas, lai šāds video, kaut vai pasniegts reālā laikā, būtu īpaši lietderīgs, jo dēļ lielā izmaiņu daudzuma būtu grūti koncentrēties uz to, kas redzams aiz caurspīdīgās automašīnas. Iegūtais video pieejams tiešsaistē adresē <http://youtu.be/VXTEUmcETgg>.

3.6. Diskusija

Šajā nodaļā autors vēlas brīvā formā pastāstīt par jautājumiem, kurus lasot šo darbu pēc autora domām uzdos cilvēki, kas ir pazīstami ar autora tēmas – attēlu apstrāde domēnu.

Divas ļoti svarīgas lietas, kas saistās ar zīmīgo punktu atrašanu, bet nav padziļināti iztirzātas šajā darbā ir maksimālais zīmīgo punktu skaits ko meklēt attēlā un Hamming distances sliexsnis atbilstošo punktu atrašanas algoritmā.

Autors uzskata, ka tā kā šī darba ietvaros nav izdevies atrast risinājumu problēmai kā noteikt vai sapludinātā attēla ģeometrija ir korekta vai nav, tad veikt vērā ņemamus eksperimentus par abām augstāk minētajām tēmām ir ļoti laikietilpīgi un iegūtie rezultāti nebūtu objektīvi, jo saturētu autora subjektīvo viedokli par to, cik labs ir iegūtais sapludinātais attēls. Nodaļā 3.3.1. autors gan apskata maksimālo meklēto zīmīgo punktu skaitu attēlā kontekstā par labākā attēla izvēlēšanos attēlu sapludināšanai, apskatot ātrdarbības rezultātus, kas iegūti šajā nodaļā autors ir spiests secināt, ka meklēt pilnīgi visus zīmīgos punktus attēlā nav optimāli, jo meklējot daudz mazāk punktus ir iespējams iegūt daudz labākus ātrdarbības rezultātus. Maksimālais meklēto zīmīgo punktu skaits attēlā pēc autora domām ir tieši saistīts ar Hamming distanci, ko izmanto zīmīgo punktu sakritību noteikšanai, kas nozīmē, ka veicot eksperimentus ar vienu no šiem lielumiem ir jāņem vērā arī otrs lielums, no kā seko, ka nepieciešams ļoti liels daudzums attēlu sapludināšanu ar dažādām šo divu parametru konfigurācijām, lai noskaidrotu, kādi parametri sniedz labākos rezultātus attēlu sapludināšanā nodrošinot salīdzinoši labu veiktspēju. Autors nevar garantēt, ka veicot šādus eksperimentus ar dažādām šo parametru konfigurācijām izdotos atrast optimālo rezultātu, bet veidotos labāka nojausma par to, kādiem šiem parametriem ir jābūt. Ja autoram būtu izdevies implementēt sapludinātā attēla ģeometrijas korektuma noteikšanas algoritmu, to iespējams varētu izmantot attēlu sapludināšanas laikā, lai izvēlētos pareizo sliexsni maksimālajam meklēto zīmīgo punktu skaitam attēlā un Hamming distancei, kas tiek izmantota sakrītošo zīmīgo punktu atrašanai.

3.7. Secinājumi par otro prototipu

Otrais prototips spēj sniegt apmierinošus attēlu sapludināšanas rezultātus lielākajā daļā gadījumu, tomēr daļa sapludināto kadru satur pārāk lielu attēla kropļojumu lai tos atzītu par pieņemamiem. Otrais prototips ir salīdzinoši ātrāks par pirmo prototipu, bet tomēr nespēj visos gadījumos izveidot vismaz vienu sapludināto attēlu sekundē, tāpēc autoram nākas atzīt, ka uzstādītais uzdevums nav sasniegts, lai gan viena no izmantotajām testa datu kopām ļāva iegūt vidēji vienu kadru sekundē, dažu kadru iegūšana aizņēma pat vairāk nekā 4 sekundes. Otrā prototipa izstrādes laikā autors padziļināti izpētīja vairākas ar attēlu sapludināšanu

saistītas lietas, noskaidrojot, ka izvēlēto attēlu uzņemšanas vietai ir ļoti liela nozīme lai iegūtais sapludinātais attēls būtu korekts. Tika noskaidrots, ka viedtālrunu GPS sistēmas nav iespējams izmantot attēlu uzņemšanas vietas noteikšanai, jo to precizitāte ir pārāk zema. Autors arī secināja, ka izmantot sapludinātu attēlu, lai veidotu vēl vienu sapludināto attēlu nav laba doma, ja iegūtajā attēlā vairs nav iespējams saprast kas atrodas aiz caurspīdīgās automašīnas.

4. NOBEIGUMS UN SECINĀJUMI

Darba gaitā autors izpētīja vairākas pieejas caurspīdīga objekta iegūšanai par pamatu ņemot scenāriju kurā divas ar kamerām aprīkotas automašīnas pārvietojas pa ceļu viena aiz otras, autora mērķis ir bildē, kas uzņemta no aizmugurē braucošās automašīnas padarīt priekšā braucošo automašīnu caurspīdīgu.

Kā vienu no pieejām autors apskatīja iespēju attēlā atpazīt priekšā braucošo automašīnu, lai aizmugurē braucošās automašīnas attēlā aizvietotu tikai to daļu, kas satur priekšā braucošo automašīnu, bet darba gaitā autoram neizdevās atpazīt automašīnu ne lietojot kontūru noteikšanas metodi, ne lietojot objektu atpazīšanas metodi, tāpēc autors izvēlējās tālāk šīs metodes maģistra darba ietvaros neapskatīt.

Vēl viena metode, ko autors izmēģināja bija attēlu sapludināšanas metode, kas atrod līdzīgās iezīmes gan priekšā, gan aizmugurē braucošo automašīnu uzņemtajos attēlos un izmantojot informāciju par līdzīgajām iezīmēm mēģina saprast kā attēls ko uzņēmusi priekšā braucošā automašīna būtu jāievieto attēlā, ko uzņēmusi aizmugurē braucošā automašīna, lai iegūtu caurspīdīgu priekšā braucošo automašīnu un sniegtu korektu informāciju par to, kas notiek aiz šīs priekšā braucošās automašīnas. Šī metode izrādījās veiksmīgāka un autoram izdevās iegūt attēlus, kuros redzama caurspīdīga priekšā braucoša automašīna, turklāt caurspīdīguma līmenis ir regulējams, ļaujot iegūt pat pilnībā caurspīdīgu automašīnu.

Darba gaitā autors izstrādāja divus prototipus, kas izmanto attēlu sapludināšanas metodi. Pirmā prototipa ietvaros autors izmēģināja šo metodi, lai saprastu vai tā spēj sniegt pietiekoši labus rezultātus, otrā prototipa ietvaros autors mēģināja uzlabot šo metodi, lai tā atbilstu autora noteiktajiem mērķiem šī maģistra darba ietvaros.

4.1. Pirmā prototipa rezultātu apskats

Autors arī veica attēlu sapludināšanas metodes novērtējumu pielietojot šo metodi vairākiem secīgiem attēliem, kas uzņemti no divām automašīnām, kas pārvietojas pa ceļu viena aiz otras. Šis novērtējums vēlreiz apstiprināja autora aizdomas, ka šo metodi vēl nepieciešams uzlabot, jo tai atklājās vēl viens trūkums - attēliem, ko izmanto sapludināšanā jābūt uzņemtiem pēc iespējas tuvāk, t.i. ideālā gadījumā tiem jābūt uzņemtiem vienā un tai

pašā vietā, lai iegūtu pareizu sapludināto attēlu, bet pirmajā prototipā izmantotā iezīmju atrašanas metode ne vienmēr no vairākiem priekšā braucošās automašīnas uzņemtajiem attēliem dod vislielāko sakritību tieši ar optimālo attēlu, kuru būtu vislabāk izmantot attēlu sapludināšanai.

4.2. Otrā prototipa rezultātu apskats

Autors analizēja prototipa veiktspēju un panāca vērā ņemamu veiktspējas uzlabojumu nomainot vienu zīmīgo punktu meklēšanas algoritmu uz citu – ātrāku, pietam autors uzskata, ka šī algoritma maiņa iegūtos attēlu sapludināšanas rezultātus noteikti neparādīja sliktākus.

Otrā prototipa sniegtie attēlu sapludināšanas rezultāti vairākos gadījumos sniedza ļoti labus attēlus un ļoti retos gadījumos iegūtie attēli bija nepieņemamas kvalitātes. Autors uzskata, ka iegūto attēlu kvalitāti ir iespējams pilnveidot, lai visi iegūtie attēli būtu ar pieņemamu kvalitāti, pilnveidojot attēlu izvēles algoritmu priekš attēlu sapludināšanas.

Autors ir ieguvis vairākus apmierinošus video ar caurspīdīgu priekšā braucošo automašīnu, bet diemžēl šādus video nav iespējams iegūt reālā laikā izmantojot otro prototipu.

4.3. Kopsavilkums

Kopumā darba gaitā autors ir daļēji sasniedzis izvirzīto mērķi virtuālās realitātes efekta iegūšanu padarot priekšā braucošo automašīnu caurspīdīgu, jo darba gaitā ir izdevies padarīt priekšā braucošo automašīnu caurspīdīgu, bet iegūtie rezultāti pēc autora domām nav pietiekoši labi, lai ar tiem samierinātos. Daļēji sasniegts arī darba galvenais uzdevums – algoritma izstrāde caurspīdīga objekta iegūšanai, jo darbā izklāstīta metode, kas pielietota caurspīdīga objekta iegūšanai, kā arī izklāstītas vairākas metodes, kuras izmantojot autoram neizdevās sasniegt darbā izvirzīto mērķi. Autoram izdevās sasniegt izvirzīto mērķi par vismaz viena kadra ar caurspīdīgu objektu iegūšanu sekundē, bet tas nav apmierinošs rezultāts, jo 1s uz kadru bija šī eksperimenta vidējais rezultāts, dažiem kadriem bija nepieciešamas līdz pat 4.5s, kā arī izmantojot citus testa datus nemaz neizdevās iegūt vidējo rezultātu, kas būtu tuvu 1s uz kadru. Tāpēc autors uzskata, ka šī maģistra darba gaitā šis mērķis nav sasniegts.

Autors uzskata, ka turpmākam pētījuma virzienam jābūt sapludināto attēlo ģeometrijas korektuma noteikšanas algoritmam, jo šādu algoritmu varētu izmantot gan labāku sapludināto attēlu ieguvei, gan izveidotā prototipa ātrdarbības uzlabošanai.

5. PATEICĪBAS

Autors vēlas izteikt pateicību visiem cilvēkiem, kas palīdzēja maģistra darbam nepieciešamo testa datu iegūšanā, šie cilvēki ir: Ainars Judvaitis, Toms Judvaitis, Līga Judvaite, Una Untule, Katrīna Dolbina, Rihards Balašs un Sergejs Voronovs.

Autors vēlas izteikt pateicību Elektronikas un Datorzinātņu Institūtam par atļauju izmantot bezvadu sensoru tīklu testa vides aparatūru maģistra darbam nepieciešamo testa datu iegūšanai.

6. IZMANTOTĀ LITERATŪRA UN AVOTI

1. J. N. Bailenson, J. Blascovich, „Virtual Reality and Social Networks Will Be a Powerful Combination”, *IEEE Spectrum*, 31. Maijs, 2011; - [atsauce 22.01.2014.]. Pieejams: <http://spectrum.ieee.org/telecom/internet/virtual-reality-and-social-networks-will-be-a-powerful-combination>
2. Google Glass [tiešsaiste]. – [atsauce 22.01.2014.]. Pieejams: <http://www.google.com/glass/start/>
3. J. P. Rolland, K. P. Thompson, „See-Through Head Worn Displays for Mobile Augmented Reality”, *Proc. China National Computer Conference (CNCC 2011)*.
4. F. Argelagueta, A. Kulikb, A. Kunert, C. Andujara, B. Froehlichb, „See-through Techniques for Referential Awareness in Collaborative Virtual Reality”, *International Journal of Human-Computer Studies*, 69(6), Jūn. 2011, pp. 387-400.
5. V. Buchmann, T. Nilsen, M. Billingham, „Interaction With Partially Transparent Hands And Objects”, *Proc. 6th Australasian User Interface Conference (AUIC 2005)*, pp. 17-20.
6. D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell1, C. Shen, „LucidTouch: A See-Through Mobile Device”, *Proc. of the 20th annual ACM symposium on User interface software and technology (UIST 2007)*, pp. 269-278.
7. The New Age of Discovery [tiešsaiste]. – [atsauce 22.05.2014.]. Pieejams: <http://www.landrover.com/gb/en/lr/new-age-of-discovery-reveal/>
8. P. Gomes, F. Vieira, M. Ferreira, „The See-Through System: From Implementation to Test-Drive” *Proc. of the 2012 IEEE Vehicular Networking Conference (VNC 2012)*
9. OpenCV [tiešsaiste]. – [atsauce 22.01.2014.]. Pieejams: <http://opencv.org>
10. Emgu CV: OpenCV in .NET [tiešsaiste]. – [atsauce – 22.01.2014.]. Pieejams: http://www.emgu.com/wiki/index.php/Main_Page
11. J. Canny, „A Computational Approach to Edge Detection”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6), 1986, pp. 679-698.
12. S. Suzuki, K. Abe, „Topological Structural Analysis of Digitized Binary Images by Border Following”, *Computer Vision, Graphics, and Image Processing*, 30(1), Apr. 1985, pp. 32-46.

13. P. Viola, M. Jones, „Rapid Object Detection using a Boosted Cascade of Simple Features”, *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511-518.
14. P. Viola, M. Jones, „Robust real-time face detection”, *International Journal of Computer Vision*, 57(2), pp. 137-154.
15. Tutorial: OpenCV [tiešsaiste]. – [atsauce 22.01.2014.]. Pieejams: <http://note.sonots.com/SciSoftware/haartraining.html>
16. H. Bay, T. Tuytelaars, L. Van Gool, „Surf: Speeded up robust features”, *Proc. European Conference on Computer Vision (ECCV 2006)*, pp. 404-417.
17. D. G. Lowe, „Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, 60(2), Nov. 2004., pp. 91-110.
18. E. Rublee, V. Rabaud, K. Konolige, G. Bradski, „ORB: an efficient alternative to SIFT or SURF”, *International Conference on Computer Vision (ICCV 2011)*, pp. 2564-2571.
19. Camera Calibration and 3D reconstruction [tiešsaiste]. – [atsauce 22.05.2014.]. Pieejams: [http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#Mat_findHomography\(InputArray srcPoints, InputArray dstPoints, int method, double ransacReprojThreshold, OutputArray mask\)](http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#Mat_findHomography(InputArray srcPoints, InputArray dstPoints, int method, double ransacReprojThreshold, OutputArray mask))
20. AirLive_POE-100CAMv2_SpecSheet [tiešsaiste]. – [atsauce 22.05.2014.]. Pieejams: http://fs.airlive.com/manual/AirLive_POE-100CAMv2_SpecSheet.pdf
21. E. Rosten, T. Drummond, „Fusing Points and Lines for High Performance Tracking” *Proc. Tenth International Conference on Computer Vision (ICCV 2005)*, vol. 2, pp. 1508-1515.
22. M. Calonder, V. Lepetit, C. Strecha, P. Fua, „BRIEF: Binary Robust Independent Elementary Features” *Proc. 11th European Conference on Computer Vision (ECCV 2010)*, pp. 778-792.
23. J. E. Bresenham, „Algorithm for computer control of digital plotter”, *IBM Systems Journal*, Volume 4, issue 1, pp. 25-30.
24. P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291 – 307, 1999.

Maģistra darbs: **Virtuālā realitāte: Caurspīdīgi objekti**

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **p i e m ē r o t u / n e p i e m ē r o t u** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: _____
(Vadītāja paraksts)

Darbs iesniegts maģistrantūras sekretariātā _____ 26.05.2014. _____
(Iesniešanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____
(Metodiķes paraksts)

Recenzents: docents, Dr.sc.comp. Kārlis Freivalds
(Akad.amats, zin.grāds, vārds, uzvārds)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____
(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)