



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JIMI VIRTANEN
AUTON NOPEUDEN MITTAAMINEN VIDEOSTA

Kandidaatintyö

Tarkastaja: professori Joni Kämäräinen
Tarkastaja ja aihe hyväksytty
19. tammikuuta 2018

TIIVISTELMÄ

JIMI VIRTANEN: Auton nopeuden mittaaminen videosta
Tampereen teknillinen yliopisto
Kandidaatintyö, 19 sivua, 0 liitesivua
toukokuu 2018
Tieto- ja sähkötekniikan TkK-tutkinto-ohjelma
Pääaine: Signaalinkäsittely
Tarkastaja: professori Joni Kämäräinen

Avainsanat: tunnistus, seuranta, koordinaattimuutos, kuvankäsittely

Videopohjaiset nopeudenmittausmenetelmät ovat valtaamassa alaa Doppler-ilmiöön perustuvilta nopeustutkilta. Perinteiset tutkat kärsivät radiotaajuushäiriöstä ja rajoittuvat vain yhden kohteen yhtäaikaiseen mittaukseen. Tarkat kamerat sekä kehittyneet kohteen-tunnistusmenetelmät mahdollistavat uudenlaisen ratkaisumallin nopeuden mittaamiseen. Ruuhkanhallintaa sekä nopeusvalvontaa toteutetaan jo videodatan perusteella.

Tässä työssä käydään läpi videopohjaisen nopeudenmittaamisen kaksi päävaihetta: auton segmentointi videokuvasta sekä koordinaattimuunnos kuvatasosta 3D-maailmaan. Näiden vaiheiden toteuttamiseen on useita tapoja, mutta aiheeseen liittyvien töiden [1-3] perusteella vain suosituimmat menetelmät käsitellään. Lopuksi virhelähteitä sekä mittaamiseen liittyviä haasteita havainnollistetaan oman menetelmän avulla.

SISÄLLYSLUETTELO

1.	JOHDANTO	4
2.	KOHTEEN NOPEUDEN MITTAAMINEN VIDEOSTA.....	5
2.1	Videon tausta tunnettu.....	5
2.2	Videon tausta tuntematon.....	6
2.3	Nopeus pyörimisliikkeestä	8
3.	LIKKUVAN KOHTEEN SEGMENTOIMINEN KUVASTA	10
3.1	Keskiarvosuodin.....	10
3.2	Normaalijakaumaan perustuen.....	10
4.	MENETELMÄ.....	12
4.1	Frame differencing	12
4.2	Koordinaattimuutos.....	13
5.	TULOKSET JA JOHTOPÄÄTÖKSET.....	16
	LÄHTEET.....	18

LYHENTEET JA MERKINNÄT

HFOV engl. horizontal field of view, vaakasuuntainen näkökenttä
VFOV engl. vertical field of view, pystysuuntainen näkökenttä

1. JOHDANTO

Nopeuden mittaaminen voi perustua fysikaalisiin ilmiöihin tai kappaleen paikan tuntemiseen eri ajanhetkillä. Näistä esimerkkeinä ovat tutkissa hyödynnettävä Dobbler-ilmiö ja GPS-pohjainen nopeuden mittaaminen. Eri tilanteisiin pitää löytää soveltuvin menetelmä, joka tarjoaa tarvittavan tarkkuuden ja käytettävyyden sopivassa ”paketissa”. Autojen nopeusmittareissa pyörimisliikkeen ja etenemisnopeuden yhteys on osoittautunut hyväksi ratkaisuksi; vain toteutustapa on muuttunut kaapelikäyttöisestä elektroniseksi sadan vuoden aikavälillä [4].

Kun keskitytään ulkopuolisen kappaleen nopeuden mittaamiseen, viranomaistenkin käyttämät tutkat ja lasermittauslaitteet (engl. radar ja lidar) toimivat useissa tapauksissa [5]. Molemmat vaativat kohteen kulkevan suoraan mittaria päin tai siitä pois päin tarkan lukeman saavuttamiseksi. Muissa tapauksissa mittaustulos vääristyy kosiniefektin takia [6]. Laser- ja radiotaajuuksilla toimivat tutkat ovat alttiita myös radiotaajuushäiriöille sekä kohteentunnistusongelmille [6]. Kuvankäsittelyyn sekä trigonometriaan perustuva lähestymistapa on robustimpi vastaaville häiriöille, ja houkutteleva vaihtoehto myös tarkkojen kameroiden ja prosessointitehon ollessa jokaisen ulottuvilla.

Tässä työssä keskitytään kappaleen nopeuden määrittämiseen videosta ja siihen liittyviin haasteisiin. Tavoitteena on tutustuttaa lukija jo olemassa oleviin videopohjaisiin menetelmiin ja luoda niiden sekä käytetyn opetusdatan avulla mittaamenetelmä tilanteisiin, joissa tausta on tuntematon. Käytännössä tämä mahdollistaa esimerkiksi älypuhelimien käyttämistä ”mobiilina tutkana”.

Kappaleen nopeuden mittaaminen voidaan jakaa kahteen luokkaan riippuen siitä, tunnetaanko jotain taustasta tai kappaleesta. Näiden tapauksien erot ja vaikutukset mittaustulokseen ja tarvittaviin menetelmiin on kerrottu toisessa luvussa. Tämän jälkeen luvussa 3 käydään läpi eri hahmontunnistusmenetelmiä. Luvussa 4 esitetään työn kokeellinen osa. Viides luku kokoaa saavutetut tulokset ja suurimmat virheiden lähteet.

2. KOHTEEN NOPEUDEN MITTAAMINEN VIDEOSTA

Nopeuden mittaamiseksi tarvitaan esitietoja, jotka kuvaavat kohteen, ympäristön ja mitauskaluston tunnettuja parametreja. Yhdessä mitatun tiedon kanssa nopeus voidaan laskea tilanteeseen soveltuvalla menetelmällä. Esimerkkinä käytetään auton nopeusmittaria termien ymmärtämiseksi. Esitietona on renkaan halkaisija ja mitattuna tietona renkaan kulmanopeus. Tarvitaan algoritmi, joka muuttaa mainitut tiedot nopeusmittarin lukemaksi.

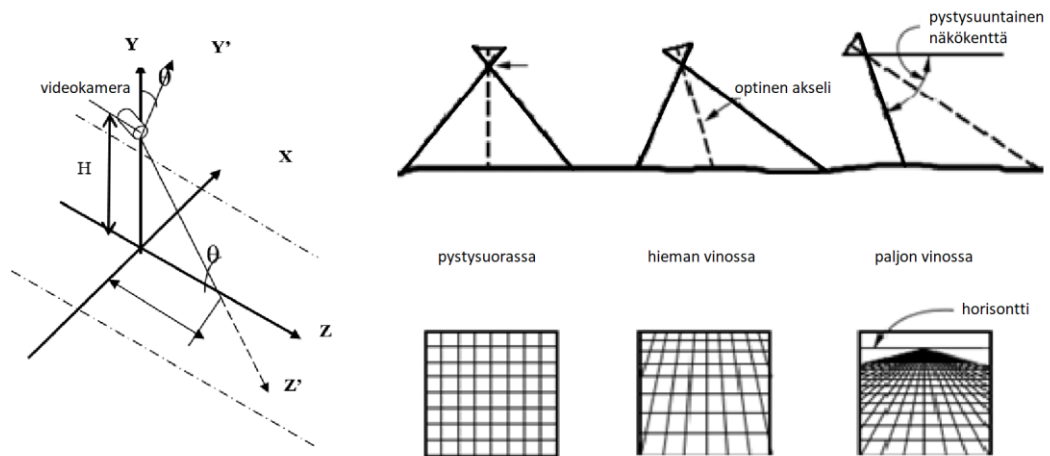
Seuraavissa alaluvuissa tarkastellaan videopohjaisia nopeudenmittausmetodeja, joissa esitiedot eroavat toisistaan. Esitietojen oletetaan määrityksen jälkeen pysyvän muuttumattomina. Todellisuudessa ne eivät pysy täysin vakiona; esimerkissä renkaan halkaisija voi muuttua siirryttäessä kesä- ja talvirenkaiden välillä, jolloin virhe esitiedoissa aiheuttaa vääristymän lopputulokseen.

Videopohjaisissa nopeudenmittausmenetelmissä kohteen aikaero pisteiden välillä saadaan suoraan videodatasta, joten haasteeksi jää kohteen tunnistus ja sen sijainnin selvittäminen. Jokaisella menetelmällä on rajoituksia, joiden sisällä ne on suunniteltu toimimaan. Esimerkkinä käytetty auton nopeusmittari rajoittuu yhden auton nopeuden mittaamiseen.

2.1 Videon tausta tunnettu

Käytössä olevat videopohjaiset nopeudentunnistusmenetelmät perustuvat usein tilanteisiin, joissa tunnetaan taustan etäisyystietoja. Yksinkertaisimmillaan tarvitaan vain kahden kuvapisteen välinen etäisyys maailmakoordinaateissa. Esineeltä kuluu jokin aika tämän välin kulkemiseen, ja kappaleen keskinopeus voidaan määrittää. Nämä pisteet määräytyvät kuvatasossa kahden koordinaatin avulla eivätkä näin ollen voi yksiselitteisesti kuvantaa 3-ulotteisen maailman pistettä. Ajatellaan kameran olevan sijoitettu kuten poliisin nopeudenvalvontakamerat kuvaamaan tietä yläviistosta. Autojen liike tapahtuu tasoa pitkin, ja koordinaattimuunnos 2-ulotteisten koordinaatistojen välillä on mahdollista.

Aiheesta on julkaistu useita tutkimuspapereita, joissa esitetään ratkaisuehdotuksia liikenteen monitorointiin ja nopeuden valvontaan [1-3]. Kyseisten tutkimusten tavoitteena on selvittää videopohjaista korvaajaa käytössä oleville nopeusvalvontakameroille. Koordinaattimuutoksen ja autojen seurannan takia kameran sijainnilla on rajoituksia. Parhaan tuloksen saamiseksi kamera pitää sijoittaa tien yläpuolelle kuvamaan alaviistoon kuvan 1 mukaisesti.



Kuva 1. Vasemmalla kameran positio ja oikealla allekkain kameran kulman vaikutus näkymään. Muokattu lähteistä [1, 3].

Kuvan 1 vasemmalla puolella on esitetty kamerakeskeiset koordinaatit Z' ja Y' suhteessa maailmakoordinaatteihin X , Y ja Z . Oikealla puolella on havainnollistettu kameran optisen akselin Z' suuntauksen vaikutusta. Tie kameran alapuolella on kuvattu tasana, joka on jaettu yhtä suuriin neliöihin. Mainituissa tutkimuksissa [1-3] oli päädytty pieneen kallistuskulmaan suhteessa akseliin Y niin, että horisontti ei ole näkyvässä. Tällä järjestelyllä pyritään saavuttamaan riittävän pitkä autojen tunnistusalue tarkkuuden liikaa kärsimättä.

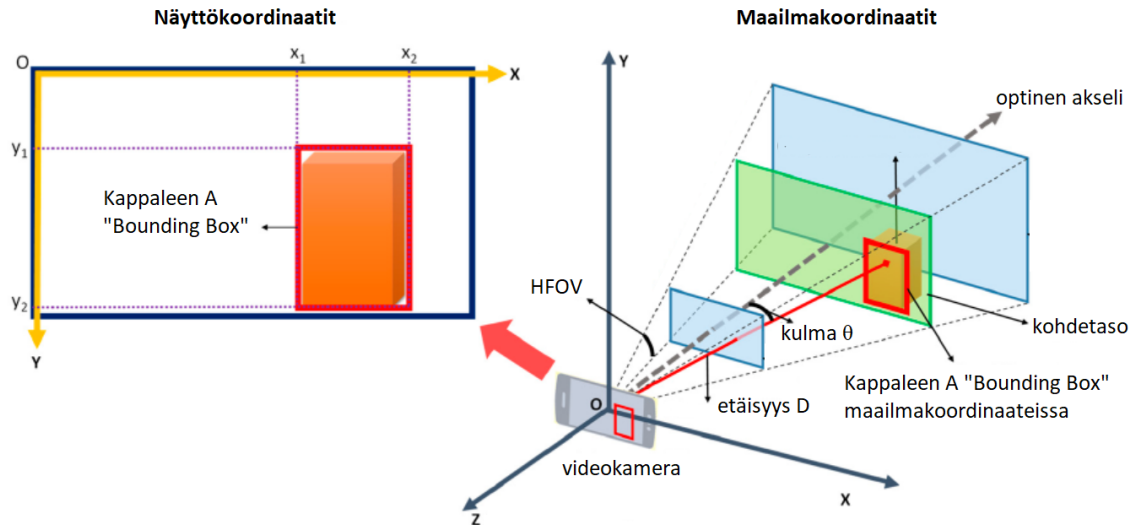
Koska autojen sijainti saadaan laskettua missä tahansa tason pisteessä, on jatkuva-aikainen nopeusseuranta mahdollista. Resoluution ja tunnistustarkkuuden pitää olla niin hyviä, että auton sijoittaminen tienpinnan oikeaan kohtaan onnistuu riittävällä tarkkuudella. Usean auton samanaikainen seuranta sekä korkea nopeuksien päivitystahti kasvattaa prosessointitehon tarvetta.

Jianping *et al.* saivat autojen nopeuden määritettyä 4 %:n tarkkuudella [3]. Järjestelmällä on potentiaalia korvata perinteiset nopeusvalvontakamerat, mutta mobiiliksi tutkaksi se ei suoraan sovellu.

2.2 Videon tausta tuntematon

Jotta nopeuden määrittäminen onnistuu ilman esitietoja taustasta, pitää itse kappaleesta tuntea jotain. Muuten lähellä oleva pieni kappale ja kaukana olevaan suuri kappale käsitellään samalla tavalla eivätkä tulokset ole luotettavia. Mahdollisia tietoja kappaleesta ovat sen etäisyys ja koko. Seuraavaksi tarkastellaan tilannetta, jossa kappaleen etäisyys kameraan tunnetaan.

Jinmeng Rao *et al.* [7] esittävät artikkelissaan lisätty todellisuus -sovelluksen mobiililaitteille. Sovelluksen ideana on visualisoida esimerkiksi rakennuksia laitteen näytöllä hyödyntäen hahmontunnistusta ja koordinaattien yhteyksiä maailman ja näytön välillä. Nämä koordinaattisysteemit ja niiden yhteydet toisiinsa on esitetty Kuvassa 2.



Kuva 2. Näyttö- ja maailmakoordinaattien yhteydet. Muokattu lähteestä [7].

Kohteen A sijainti on selvitetty Kuvan 2 vasemmassa reunassa olevan näyttökuvan perusteella. On syytä huomioda, että kohde A saadaan eroteltua kuvasta vasta hahmontunnistuksen jälkeen. Hahmontunnistuksesta kerrotaan luvuissa 3 ja 4.1.

Kappaleelle A on muodostettu ”Bounding box”, joka kuvaa pienintä näyttökoordinaattien rajaamaa aluetta, jonka sisään kohde A mahtuu. Näyttökoordinaattien origo sijaitsee näytön vasemmassa yläreunassa ja maailmakoordinaattien origo näytön keskellä. Näyttökoordinaattien origo on siirretty näytön keskelle, jotta 2D-3D -muunnos onnistuu.

Seuraavaksi työssä ratkaistiin kohdetason Z-koordinaatti. Sen selvittämiseksi pitää tietää mobiililaitteen ja kohteen välinen etäisyys D, jota on Kuvassa 2 havainnollistettu punaisella viivalla. Etäisyys saadaan laskettua laitteen GPS-anturien ja järjestelmässä olevien kohteiden sijaintitietojen avulla. Kun kameran optisen akselin ja kohteen A välinen kulma on θ , on kohdetason Z-koordinaatti muotoa $D \cdot \cos(\theta)$.

Kun kohdetason Z-koordinaatti on tiedossa, saadaan näyttö- ja maailmakoordinaattien suhde muotoon

$$ratio = \frac{Height_s}{2 \cdot \tan\left(\frac{\beta}{2}\right) \cdot Z_r}, \quad (1)$$

jossa $Height_s$ on näytön korkeus, β kameran pystysuuntainen näkökenttä ja Z_r kohdetason ja kameran välinen etäisyys. Tällä muutossuhteella jokainen kappaleen A piste saa-

daan muutettua oikean maailman koordinaateiksi. Tarkastellussa työssä näiden koordinaattien avulla järjestelmään tallennettuja virtuaalisia kohteita voidaan esittää oikeassa kohdassa mobiililaitteen näytöllä. Järjestelmä ei toimi kuitenkaan nopeustutkana, sillä kohteen etäisyys kamerasta oli esitieto, joka saatiin GPS-datasta. Jos autojen reaaliaikaiset sijaintitiedot olisivat käytössä, niiden avulla nopeuden määrittäminen onnistuisi suoraan ilman kuvankäsittelyä. Käsitellään seuraavaksi toinen vaihtoehto, jossa esitietona on mitattavan kappaleen koko. Se on myös tilanne, jolle tämän työn kokeellinen osuus suoritetaan.

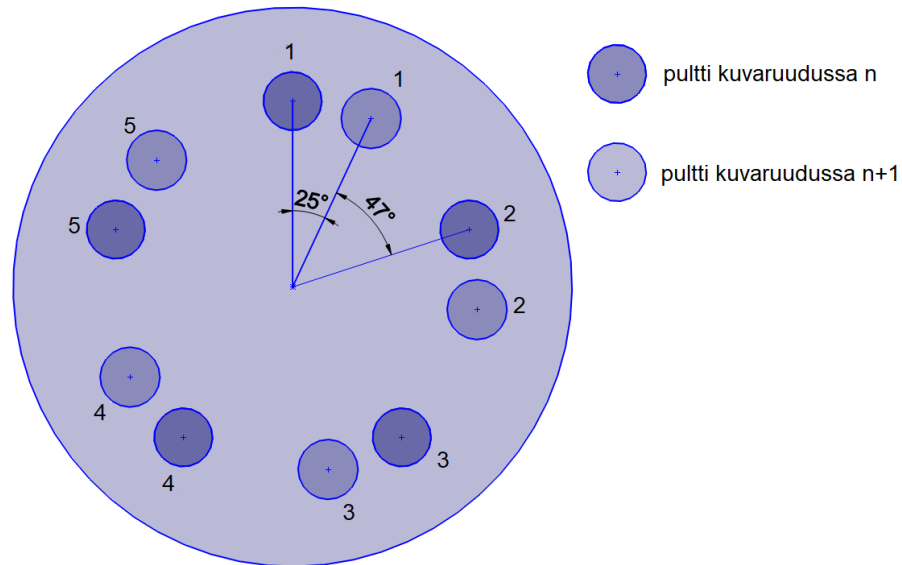
Tarkastellaan Kuvan 2 avulla, miten esitiedon muutos vaikuttaa kappaleen maailmakoordinaattien selvittämiseen. Aiemmassa tilanteessa tiedossa ollut kappaleen etäisyys (punainen viiva) on nyt tuntematon. Kappaleen koko, esimerkiksi auton korkeus oikeassa maailmassa sen sijaan tunnetaan, mikä riittää kappaleen yksikäsitteiseen sijoittamiseen 3D-maailmaan. Tilannetta voidaan ajatella seuraavasti: Kameran pysty- ja vaakasuuntainen näkökenttä ovat tiedossa, joten näkökenttä rajaa tietyn kokoisen tason tietyllä etäisyydellä kamerasta. Kun tunnetun kokoinen kappale täyttää näyttötasosta mitattavissa olevan alueen, voidaan kohdetason koko ja sen etäisyys määrittää. Koordinaattimuunnos näyttökoordinaateista maailmakoordinaatteihin tässä tilanteessa käsitellään tarkemmin alaluvussa 4.2.

2.3 Nopeus pyörimisliikkeestä

Toinen lähestymistapa auton nopeuden määrittämiseen on hyödyntää renkaan kulmanopeuden suhdetta auton nopeuteen. Toimintaperiaate olisi identtinen auton oman nopeusmittarin kanssa; mitatun kulmanopeuden ja tiedetyn renkaan halkaisijan avulla laskeaan auton nopeus. Käytännössä menetelmä ei ole mielekäs, sillä videopohjainen renkaan kulmanopeuden määrittäminen asettaa tiukat vaatimukset kameran sijainnille, tarkkuudelle ja kuvataajuudelle.

Ensimmäinen rajoitus on kameran sijainti: jotta rengas olisi näkyvässä, pitää kameran sijaita auton sivulla. Auton liike vie renkaan nopeasti pois kuvasta, ellei kamera siirry auton mukana. Kamera liikettä ei tosin ole rajoitettu kuten alaluvuissa 2.1 ja 2.2, joissa autolle määritettiin kamerakeskeiset maailmakoordinaatit eri ajanhetkillä. Oli kamera sitten paikoillaan esimerkiksi tien reunassa tai liikkeessä kuvattavan renkaan kanssa, pitää resoluution ja kuvataajuuden olla riittävät.

Renkaasta pitää valita piste hahmonseuranta varten. Helpoiten seurattavat kohteet renkaasta ovat todennäköisesti vanteen puolet ja pyöränpultit. Puolien lukumäärä on usein yli kymmenen, kun pyöränpulttien lukumäärä on yleensä neljä tai viisi. Ellei pyöränpultteja ole peitetty ovat ne parempi vaihtoehto vaadittavan kuvataajuuden kannalta. Kuvataajuus pitää olla riittävän suuri, jotta seurattava pyöränpultti ei sekoitu muiden pulttien kanssa eri kuvaruutujen välillä. Kuvassa 3 on piirretty suurennos viiden pultin renkaasta, kun peräkkäiset ruudut ovat kuvattu päällekkäin.



Kuva 3. Havainnekuva valitun seurantapisteen liikkeestä kuvaruutujen välillä.

Rengas on pyörinyt kuvaruutujen välillä 25 astetta, ja seurattavan pultin 1 uusi sijainti on yksikäsitteinen. Renkaan täytyy pyöriä alle 36 astetta kuvaruutujen välillä, jotta lähimmät pultit vastaisivat aina toisiaan. Jos renkaan pyörimissuunta tiedetään, voi seurattava piste pyöriä tuplasti eli alle 72 astetta. Näin pultti löytyy kuvaruudusta $n+1$ kulmasta $[0,72[$ tiedetystä suunnasta.

Lasketaan tarvittava minimikuvataajuus edellisen esimerkin tapauksessa. Valitaan rengaskooksi yleisesti käytössä oleva 15" [8]. Tällä vannekoolla renkaan ympärysmitta on noin 2,0 m [9]. Jos auton nopeus halutaan mitata 150 km/h :iin saakka, voidaan vaadittu kuvataajuus ratkaista seuraavasti:

$$\left(\text{kehä} * \frac{\text{max_kulma}}{360} * \frac{3,6}{\text{nopeus_kmh}} \right)^{-1} \frac{1}{s} = \left(2,0 * \frac{72}{360} * \frac{3,6}{150} \right)^{-1} \frac{1}{s} = 105 \frac{1}{s}. \quad (2)$$

Tulokseksi saatu 105 kuvaa sekunnissa yhdessä korkean resoluution kanssa vaativat paljon resursseja kameralta. Tämän sekä kameran sijainnin rajoitusten takia menetelmä ei sovellu jatkuvaan nopeusseurantaan.

3. LIIKKUVAN KOHTEEN SEGMENTOIMINEN KUVASTA

Hahmontunnistus perustuu videopohjaisissa nopeudenvälvontalaitteissa yksinkertaiseen olettamukseen: jokainen kuvaruutu sisältää tietoa etualasta (engl. foreground) sekä taustasta (engl. background) [1]. Taustan kuvapisteen sisältävät kaikki staattiset kohteet kuten rakennukset, tienpinnan, parkkeeratut autot sekä taivaan. Myös muuttuvat heijastukset ja jopa mitattavan auton varjo ovat taustaa. Etuala sisältää vain liikkuvat kohteet joiden nopeutta määritetään. Jokainen kuvapiste kuuluu siis yksikäsitteisesti joko etualaan tai taustaan ja hahmontunnistuksen tavoitteena suorittaa tämä lajittelu mahdollisimman tarkasti.

Koska etuala sisältää kaiken tarvittavan tiedon laskuja varten, pitää tausta poistaa tutkitavasta kuvaruudusta (engl. background extraction). Tähän tarvitaan taustaa kuvaava malli, joka voidaan hankkia usealla tavalla.

3.1 Keskiarvosuodin

Todellisuudessa tausta kokee jatkuvia muutoksia. Valaistusvoimakkuus muuttuu säätilojen mukana ja parkista lähtevät autot jättävät taustaan ”aukkoja”. Mukautuva tausta ottaa huomioon taustassa tapahtuvat muutokset ja päivittää taustamallia jatkuvasti. Yksi muokautuvan taustan malli perustuu peräkkäisten ruutujen keskiarvoon (engl. mean filter). Yksittäisen kuvapisteen (x,y) keskiarvo $k_{xy}(t_n)$ kuvaruudussa n on [1]:

$$k_{xy}(t_n) = \frac{\sum_{m=0}^{j-1} I_{xy}(t_{n-m})}{j} . \quad (3)$$

Kuvapisteen intensiteetti ruudussa n on $I_{xy}(t_n)$ ja j kuvaruutujen määrä josta keskiarvo lasketaan. Pienellä j :n arvolla tausta mukautuu nopeammin tapahtuviin muutoksiin, mutta on toisaalta herkempi luokittelemaan hitaasti liikkuvat kohteet taustaksi. Kuten edellisen alaluvun 4.1 tapauksessa, tausta voidaan vähentää nykyisestä kuvaruudusta ja erotus muuttaa binäärikuvaksi raja-arvon perusteella.

3.2 Normaalijakaumaan perustuen

Juoksevaan normaalijakaumaan (engl. running Gaussian distribution) perustuva menetelmä on yleisesti käytössä taustan erottelussa [10]. Ideana on määrittää jokaisen kuvapisteen intensiteetille I_{xy} odotusarvo μ_{xy} sekä varianssi σ_{xy}^2 ja päivittää niitä taustan muuttuessa [11].

Odotusarvo voidaan alustaa ensimmäisestä kuvaruudusta ja varianssi laskea kuvapisteen ympäristöstä. Taustan muutosten takia odotusarvoa ja varianssia pitää päivittää. Eräs tapa on esitellä uuden intensiteetin painoarvo ρ ja laskea uusi odotusarvo ja varianssi seuraavasti [11]:

$$\mu_{xy}(t) = \rho I_{xy}(t_n) + (1 - \rho)\mu_{xy}(t_{n-1}), \quad (4)$$

$$\sigma_{xy}(t)^2 = d^2 \rho + (1 + \rho)\sigma_{xy}(t_{n-1})^2, \quad (5)$$

$$d = |I_{xy}(t_n) - \mu_{xy}(t)|. \quad (6)$$

d on kuvapisteen intensiteetin ja odotusarvon etäisyys. Tutkittavan kuvaruudun pistettä (x,y) voidaan nyt verrata taustamallin odotusarvoon ja varianssiin:

$$\frac{|I_{xy}(t_n) - \mu_{xy}(t)|}{\sigma_{xy}(t)}. \quad (7)$$

Jos tulos on suurempi kuin määritelty luottamusväli, piste luokitellaan kuuluvaksi etu-alaan.

Friedman & Russel esittävät tutkimuksessaan [12] mallin, jossa kuvapistet käsitellään painotettuna keskiarvona usean normaalijakauman kesken. Pisteiden intensiteetti on siis jakautunut seuraavasti:

$$I_{xy} = w_{xy}(r_{xy}, s_{xy}, v_{xy}). \quad (8)$$

Jakaumat r , s ja v kuvaavat todennäköisyyksiä kuvapisteen intensiteetille eri luokissa (kuvapiste tienpinnassa, osana varjoa tai ajoneuvossa). Painokerroin w_{xy} on paikasta riippuva, koska kuvapisteen todennäköisyys kuulua tiettyyn luokkaan riippuu sen sijainnista kuvaruudussa. Jianping *et al.* hyödynsivät tätä menetelmää nopeusvalvonta-algoritmissaan ja saavuttivat 4%:n tarkkuuden [3].

4. MENETELMÄ

Tämän työn kokeellisen osan tavoitteena on selvittää **auton mittoihin perustuvan** nopeuden määrittämisen tarkkuus. Poiketen yleisestä ratkaisumenetelmästä [1-3], taustasta ei tarvitse nyt tietää mitään. Koska nopeuden laskeminen tapahtuu hahmontunnistuksen jälkeen, virheet autojen tunnistustarkkuudessa lisäävät virhettä lopputulokseen. Hahmontunnistuksen onnistuminen ja sen tarkkuus riippuvat voimakkaasti käytetystä datasta. Jotta hahmontunnistuksen tarkkuus voidaan tarvittaessa unohtaa virhelähteistä, pitää videodatan olla selkeää tarkan tunnistuksen saavuttamiseksi.

Datana käytetään kaapattuja videoleikkeitä ajosimulaattorista Live for Speed. Videoiden resoluutio on 1920x1080 ja kuvataajuus 60 kuvaa/s. HFOV eli vaakasuuntainen näkökenttä on 60° ja VFOV eli pystysuuntainen näkökenttä on 33,75°. Kuvassa 4 on havainnollistettu käytettyä videodataa liittämällä kuvaruutuja eri ajanhetkiltä yhteen.



Kuva 4. Näyte ajosimulaattorista Live For Speed.

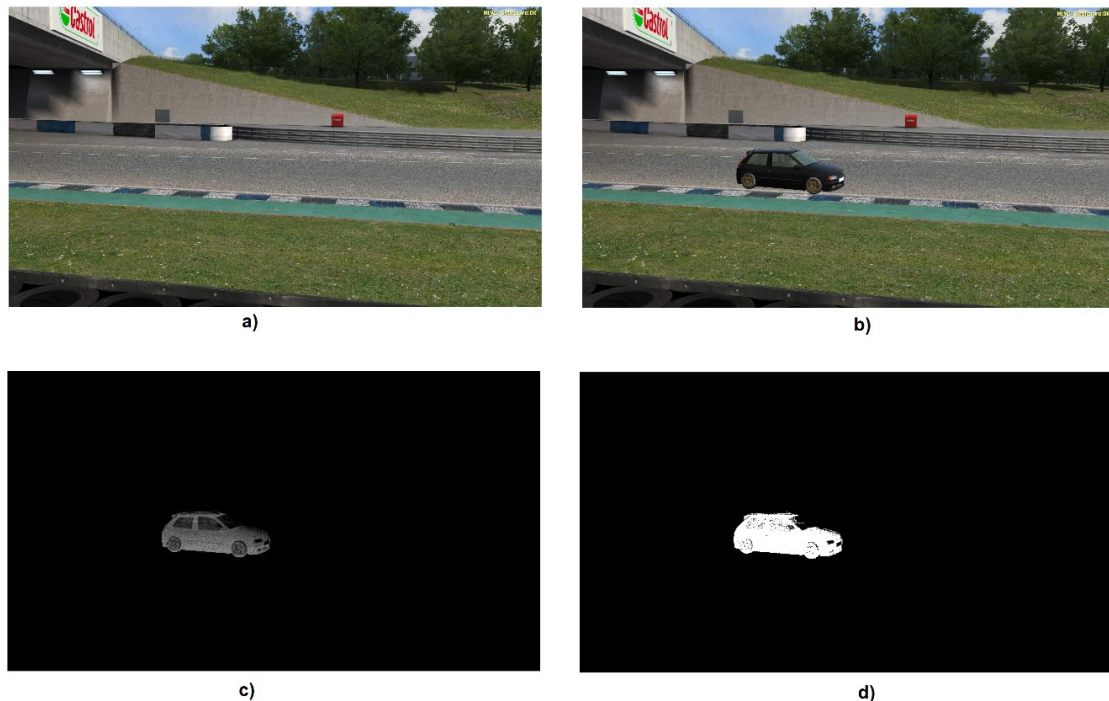
Testiauto perustuu Citroën AX:n mittoihin[13], ja sen pituus on 3,525 m, leveys 1,555 m ja korkeus 1,355 m [14]. Simulaattorista voidaan kytkeä auton varjot päälle, mikä kuvaa paremmin todellisuutta.

4.1 Frame differencing

Yksinkertaisimmillaan tausta voidaan tallentaa silloin, kun liikkuvia kohteita ei ole kuvassa [10]. Näin saatu tausta voidaan vähentää nykyisestä kuvasta, jolloin erotuskuvaan

jää vain liikkuvat kohteet. Tämä tekniikka (engl. frame differencing) toimii vain silloin kun kaikki etualan kuvapisteeet ovat liikkeessä ja taustan kuvapisteeet staattisia [11].

Jatkokäsittelyä varten kuvapisteeet pitää lajitella kuuluvaksi joko etualaan tai taustaan. Tämä onnistuu muuttamalla erotuskuva binäärikuvaksi raja-arvon avulla. Kuvassa 5 on esitetty eri vaiheita taustan poistossa.

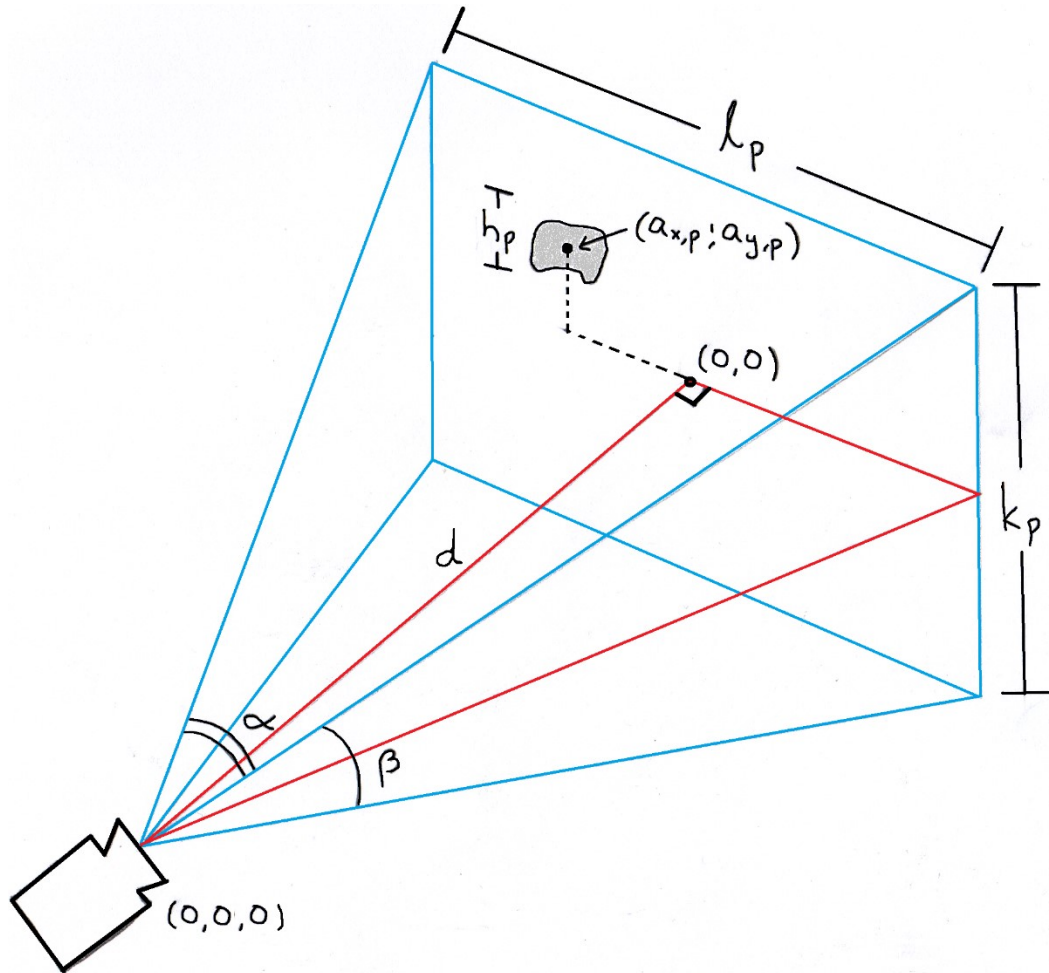


Kuva 5. Taustan poisto. a) Tallennettu tausta, kun liikkuvia kohteita ei ole kuvassa. b) nykyinen kuvaruutu, c) harmaasävykuviksi muutettujen taustan ja nykyisen kuvaruudun erotus, d) etuala muutettu yhtenäiseksi joukoksi raja-arvon ja liitännäisyyden avulla

Raja-arvo vaikuttaa miten herkästi muuttuneet kuvapisteeet luokitellaan kuuluvaksi etualaan. Sopivalla raja-arvolla saadaan suodatettua osa varjoista ilman, että tunnistustarkuus heikkenee liikaa. Liitännäisyyden avulla kohteista saadaan yhtenäisiä ja syntyneistä pistejoukoista voidaan valita jatkokäsittelyyn vain suurimmat.

4.2 Koordinaattimuutos

Nopeuden määrittäminen tässä työssä perustuu auton sijainnin vertailuun eri ajanhetkinä. Tämän takia auton sijainti kuvaruudulla pitää muuttaa sijaintiin 3D-maailmassa. Muutos on mahdollinen, koska auton mitat ovat tiedossa. Käsitellään koordinaattimuutos näyttökoordinaateista maailmakoordinaatteihin kuvan 6 avulla.



Kuva 6. Kameran näkökenttä ja kuvataso

Kamera on sijoitettu kuvan 6 vasempaan alalaitaan. Sen koordinaatit ovat $(0,0,0)$, koska maailmakoordinaatit on valittu kamerakeskeiseksi. Kamerasta lähtevät neljä sinistä viivaa kuvaavat sen näkökentän kulmia. Kulma α on vaakasuuntainen ja β pystysuuntainen näkökenttä. Molemmat ovat tunnettuja suureita. Nämä viivat muodostavat suorakulmion muotoisen kuvatason etäisyydelle d kamerasta. Kuvatason korkeus k_p ja leveys l_p kuvapisteissä saadaan kameran resoluutiosta. Alaindeksi p kertoo, että kyseinen mitta on ilmoitettu kuvapisteissä.

Havaitun auton siluetti on piirretty harmaaksi. Auton sijaintia kuvaa sen keskellä oleva piste, jonka näyttökoordinaatit ovat $(a_{x,p}; a_{y,p})$. Tavoitteena on selvittää auton sijainti metreissä ja silloin alaindeksinä on m . Auton korkeus kuvapisteissä on h_p ja metreissä h_m . Auton näyttökoordinaatit sekä korkeus kuvapisteissä ja metreissä ovat tiedossa.

Lasketaan ensin kuvatason mitat metreissä. Kuvatason korkeuden ja auton korkeuden välillä on yhteys

$$\frac{k_m}{k_p} = \frac{h_m}{h_p} \rightarrow k_m = \frac{k_p h_m}{h_p} . \quad (9)$$

Kuvatason leveys saadaan kuvasuhteen avulla seuraavasti:

$$l_m = \frac{16}{9} k_m . \quad (10)$$

Kuvatason etäisyys kamerasta voidaan laskea punaisen apukolmion avulla.

$$\tan(\alpha/2) = \frac{l_m/2}{d} \rightarrow d = \frac{l_m/2}{\tan(\alpha/2)} \quad (11)$$

Etäisyys d on myös maailmakoordinaattien z-koordinaatti jokaisessa kuvatason pisteessä. Auton x-maailmakoordinaatti on

$$\frac{a_{x,p}}{l_{p/2}} = \frac{a_{x,m}}{l_{m/2}} \rightarrow a_{x,m} = \frac{a_{x,p} l_m}{l_p} . \quad (12)$$

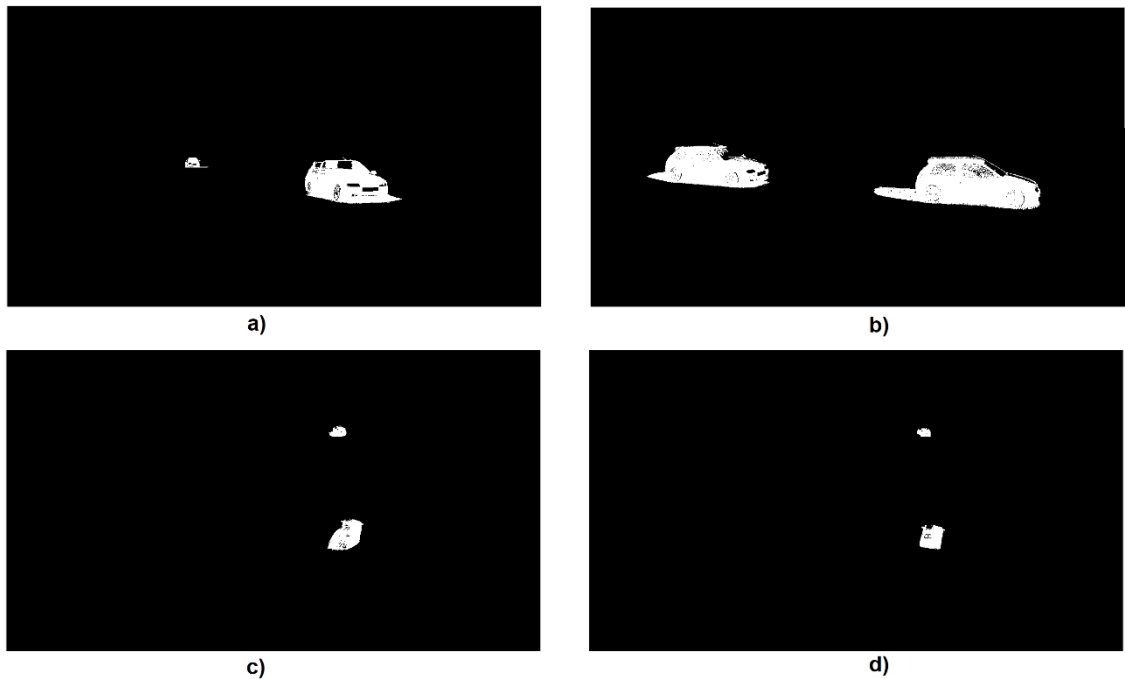
Vastaavasti auton y-maailmakoordinaatti on

$$\frac{a_{y,p}}{k_{p/2}} = \frac{a_{y,m}}{k_{m/2}} \rightarrow a_{y,m} = \frac{a_{y,p} k_m}{k_p} . \quad (13)$$

Esitetty menetelmä olettaa kameran olevan ideaalinen. Todellisuudessa kameroissa on linssivääristymiä, joiden vaikutuksesta kohteiden mittasuhteet vääristyvät kuvaruudun sijainnin funktiona [15]. Kun linssin korjausparametrit ovat tiedossa, vääristymien kompensointi onnistuu useilla kaupallisilla sovelluksilla [16].

5. TULOKSET JA JOHTOPÄÄTÖKSET

Tulokset laskettiin käyttäen ”frame differencing” etualan segmentointimenetelmää. Nopeuden määrittäminen perustui laskemalla auton maailmakoordinaattien välinen etäisyys eri ajanhetkinä. Kuvassa 7 on havainnollistettu käytettyä testausdataa.



Kuva 7. Käytetty testausdata. a) auto lähenee, auton ja kameran sijainti pystysuunnassa likimain sama, b) auto liikkuu kuvaruudun poikki, kuvakulma hieman yläviistosta, c) auto loittonee ylämäkeen, d) simulaattorista kytketty varjot pois päältä

Nopeudet laskettiin kuvassa 7 näkyville kolmelle videopätkälle. Auton todellinen nopeus oli 100 km/h. Tulokset on koottu taulukkoon 1.

Taulukko 1. Lasketut nopeudet, todellinen nopeus 100 km/h

Video	Tulos (km/h)	Tulos ilman varjoa (km/h)
a)	108,6	107,3
b)	88,0	93,2
c)	90,9	101,3

Kun tuloksia vertaa kuvaan 7, on mahdollista löytää virhelähteitä jotka vääristävät tulosta. Suuri virhelähde on auton siluetin muuttuminen. Jos kamera on samalla tasolla auton kanssa, havaitun pistejoukon korkeus vastaa auton korkeutta. Muista kuvakulmista auto ”venyy” ja auton korkeus havaitaan todellista suurempana.

Myös varjot vaikuttavat lopputulokseen, jos hahmontunnistus ei ole poistanut niitä onnistuneesti. Varjo voi muuttaa auton korkeutta havaintopisteiden välillä kuten kuvassa 7 a). Varjo on kasvattanut pistejoukon korkeutta toisessa kuvaruudussa, ja auto rekisteröityy liian lähelle. Tämän takia laskettu nopeus suurempi varjojen ollessa päällä. Kuvassa 7 b) auton painopiste on siirtynyt kulkusuuntaa vastaan toisessa kuvaruudussa. Auton kulkema matka havaittiin liian lyhyeksi ja nopeus varjon kanssa laskettuna on huomattavasti pienempi.

Tutkitulla menetelmällä on potentiaalia laskea auton nopeus kymmenen prosentin tarkkuudella. Tämä vaatii kuitenkin tarkan hahmontunnistuksen sekä korjausparametrin käyttöönoton, joka ottaa huomioon auton siluetin muutokset. Ilman korjausparametria mittaus on tarkkaa vain silloin, kun kamera on auton tasolla. Tämän lisäksi vaaditut esitiedot (kameran parametrit, auton korkeus) rajoittavat menetelmän käytettävyyttä.

LÄHTEET

- [1] A. Gholami, A. Dehghani, M. Karim, Vehicle speed detection in video image sequences using CVS method, 2010, Saatavissa (viitattu 20.1.2018) www.ijcee.org/papers/418-E1077.pdf
- [2] O. Ibrahim, H. ElGendy, A. ElShafee M., Speed Detection Camera System using Image Processing Techniques on Video Streams, 2011, s. 771-778
- [3] Jianping Wu, Zhaobin Liu, Jinxiang Li, Caidong Gu, Maoxin Si, Fangyong Tan, An algorithm for automatic vehicle speed detection using video camera, 2009 4th International Conference on Computer Science & Education, s. 193-196.
- [4] William Harris, How speedometers work, howstuffworks.com, 2007, verkkosivu. Saatavissa (viitattu 16.2.2018) <https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/speedometer.htm>.
- [5] Suomen poliisi nopeusvalvonta, verkkosivu. Saatavissa (viitattu 16.2.2018) <https://www.poliisi.fi/liikenneturvallisuus/nopeusvalvonta>.
- [6] D. Sawicki, Police Radar Basics, 2015, verkkosivu. Saatavissa (viitattu 20.2.2018) <http://copradar.com/index.html>.
- [7] J. Rao, Y. Qiao, F. Ren, J. Wang, Q. Du, A Mobile Outdoor Augmented Reality Method Combining Deep Learning Object Detection and Spatial Relationships for Geovisualization, Sensors (Basel, Switzerland), Vol. 17, Iss. 9, 2017, s. 1951. Saatavissa (viitattu 20.1.2018) <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5621345/>.
- [8] THE MOST POPULAR TIRE SIZES: R15, verkkosivu. Saatavissa (viitattu 2.4.2018) <https://capitol-tires.com/r15.html>.
- [9] Tyre Size Calculator, verkkosivu. Saatavissa (viitattu 2.4.2018) <http://www.tyresizecalculator.com/tyre-wheel-calculators/tire-size-calculator-tire-dimensions>.
- [10] Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, Review and evaluation of commonly-implemented background subtraction algorithms, 2008 19th International Conference on Pattern Recognition, s. 1-4.
- [11] Background subtraction, verkkosivu. Saatavissa (viitattu 21.4.2018) https://en.wikipedia.org/wiki/Background_subtraction.
- [12] N. Friedman, S. Russell, Image Segmentation in Video Sequences: A Probabilistic Approach, 2013. Saatavissa (viitattu 21.4.2018) <https://people.eecs.berkeley.edu/~russell/papers/uai97-shadows.pdf>.

- [13] Citroën AX, verkkosivu. Saatavissa (viitattu 10.4.2018) https://en.wikipedia.org/wiki/Citro%C3%ABn_AX.
- [14] Lfsmanual, verkkosivu. Saatavissa (viitattu 16.5.2018) https://en.lfsmanual.net/wiki/FAQ#On_what_cars_are_the_LFS_cars_based.3F
- [15] What is Distortion?, verkkosivu. Saatavissa (viitattu 22.4.2018) <https://photographylife.com/what-is-distortion>.
- [16] Distortion (optics), verkkosivu. Saatavissa (viitattu 22.4.2018) [https://en.wikipedia.org/wiki/Distortion\(optics\)#Software_correction](https://en.wikipedia.org/wiki/Distortion(optics)#Software_correction).