

OBJEKTINSEURANNASTA KUVASARJASSA

Hannu J. Haven

6.6.1996

JOENSUUN YLIOPISTO

Tietojenkäsittelytieteen laitos

Syventävien opintojen tutkielma

TIIVISTELMÄ

Tutkielman tarkoituksena on tarkastella objektinseurannan perusteita, esitellä minkälaisia objektinseurantasovelluksia ja -menetelmiä löytyy kirjallisuudesta ja empiirisesti tutkia erilaisten lohkotäsmäykseen perustuvien tekniikoiden keskinäistä paremmuutta. Tärkeimpinä objektinseurannan menetelminä esitellään lohkotäsmäysmenetelmä, aktiivisen reunan menetelmä ja alueenlaajennustekniikka. Empiirisissä testeissä tutkittiin pikselitason virhefunktioiden ja tilastollisten menetelmien kykyä seurata objekteja ennalta määritellyn staattisen objektilohkon avulla. Pikselitason virhefunktioina käytettiin absoluuttista- ja neliöllistä virhefunktiota, sekä painotettua absoluuttista virhefunktiota. Tilastollisina menetelminä käytettiin histogrammiin ja varianssiin perustuvia menetelmiä. Etsinnässä käytettiin liikkeen estimointia etsintäalueen rajaamiseen. Liikkeen estimointi perustui suoraviivaisen liikkeen ennustamiseen. Testien tuloksena todettiin, että täsmäys painotetulla neliöllisellä keskivirheellä ja etsintäalueen rajaus liikkeen estimointiin perustuvalla sijainnin ennustuksella tuotti parhaimman tuloksen. Järjestelmä kykeni seuraamaan objektia reaali maailmasta kuvatussa kuvasarjassa.

SISÄLLYSLUETTELO

1. JOHDANTO	4
2. OBJEKTI JA LIIKE	6
2.1 Havaintoaineisto	6
2.2 Objektinseurannan sovellusalueiden parametreja	8
2.3 Objektinseurannan sovelluksia	13
3. OBJEKTINSEURANNAN PERUSTEITA	19
3.1 Objektinseurantajärjestelmän rakenne.....	19
3.2 Moniobjektiseuranta	23
3.3 Liikkeen analysointi	27
4. OBJEKTINETSINTÄ LOHKOTÄSMÄYKSELLÄ	30
4.1 Objektin määritelmä	30
4.2 Täsmäys kustannusfunktion minimointina	32
4.3 Etsintäalueen rajaaminen.....	38
5. OBJEKTIN REUNOJEN SEURAAMINEN	42
5.1 Aktiivinen reuna	42
5.2 Alueenlaajennus	47
6. OBJEKTINSEURANTAJÄRJESTELMÄ.....	52
6.1 Järjestelmän kuvaus	52
6.2 Mittaustuloksia	57
7. YHTEENVETO JA LOPPUPÄÄTELMÄT.....	67
VIITELUETTELO	69

1. JOHDANTO

Objektinseurannalla (object tracking) tarkoitetaan seurattavien kohteiden liikkeiden analysointia paikantamalla objektien sijainti säännöllisinä tarkasteluhetkinä. Objektinseuranta on tutkittu jo 1950-luvulta lähtien ydinfysiikan tutkimuskohteissa, joissa tarkasteltiin hiukkasten liikeratoja väliaineessa [CHI94]. Nykyaikaisissa objektinseuranta-sovelluksissa seurattava kohde on yleensä reaali maailman objekti, joka liikkuu kolmiulotteisessa ympäristössä. Tärkeää on tällöin tunnistaa seurattavien objektien liike. Jos objektin liike tunnistetaan tai se voidaan olettaa seurattavan objektin ominaisuuksien perusteella, niin objektin liikettä voidaan ennustaa. Tällöin kyseessä on *liikkeen estimointi (motion estimation)*. Liikkeen estimointia käytetään myös liikkuvan kuvan tiivistämisessä. Esimerkiksi MPEG-standardin mukaisessa liikkuvan kuvan tiivistyksessä hyödynnetään objektinseurannan menetelmiä. MPEG-standardi ei sido objektinseurannan menetelmiä, joten tässä tutkielmassa esiteltävät menetelmät ovat sovellettavissa liikkuvan kuvan tiivistykseen.

Objektinseurannan merkitys korostuu myös robottiteknologian tutkimuksessa, jossa keinotekoisien näön eli *konenäön (machine vision)* tutkimuksessa sillä on suuri merkitys. Spatiaalisen informaation perusteella ratkaisujen tekemiseen kykenevän robotin rakentaminen mahdollistaa mielenkiintoisia sovelluksia. Esimerkiksi avaruustutkimuksessa konenäköön kykenevä robotti voisi kyetä itsenäisesti kohdistumaan tarkkailtavaan kohteeseen. Muita tämänkaltaisia järjestelmiä voisivat olla esimerkiksi kappaleen väistöön kykenevä avaruusluotain, kohteeseen hakeutuva ohjus ja reaali maailmassa itsenäiseen liikkumiseen kykenevä robotti. Objektinseurantajärjestelmän syöttötietona ei välttämättä tarvitse olla reaali maailmaa reaaliajassa kuvaava kamera, vaan kyseessä voi olla ennalta talletettu kuvasarja. Tällöin tavoitteena voi olla kuvasarjan kohteiden seuraaminen analysoinnin helpottamiseksi.

Objektinseurantajärjestelmä toimii yleensä niin, että se etsii objektin tarkasteltavasta kuvasta käyttämällä apuna liikkeen estimoinnin tuottamaa informaatiota. Objektin etsintä tapahtuu siten, että objektin määrittävää kuvaa vertaillaan saman kokoisiin kuva-alueisiin

tarkasteltavassa kuvassa. Parhaiten objektia vastaava kuva-alue valitaan objektin uudeksi sijainniksi. Tätä sijainnin määrittystä kutsutaan *täsmäykseksi (matching)*. Objektinseuranta-järjestelmä jakaantuu karkealla tasolla kahteen komponenttiin, joita ovat liikkeen estimointi ja objektin etsintä. Liikkeen estimointi toimii objektin etsinnän apuna, joten se ei ole rakenteellisesti välttämätön osa objektinseuranta-järjestelmää. Järjestelmän toiminnan kannalta sillä on kuitenkin oleellinen merkitys.

Objektinseurantaan käytetyt menetelmät riippuvat seurattavasta objektista ja siitä mitä tietoja seurattavasta objektista halutaan analysoida. Jos seurattava objekti säilyttää hyvin muotonsa seurannan aikana, soveltuu seurantaan menetelmä jota kutsutaan *lohkotäsmäykseksi (block matching)*. Se perustuu samankokoisten kuva-alueiden vertailemiseen, jolloin niiden välinen eroavaisuus ilmaisee täsmäyksen tuloksen. Jos objekti muuttaa muotoaan kuvasarjassa ja objektin reunat halutaan selvittää tarkasti, niin seurantaan soveltuvia menetelmiä ovat *aktiivisen reunan menetelmä (active contour)* ja *alueenlaajennustekniikka (region growing)*. Näissä menetelmissä objektin reunat määritetään pikselin tarkkuudella.

Tämän tutkielman tarkoituksena on hahmottaa objektinseurannan olemusta tarkastelemalla objektinseurantaan liittyviä käsitteitä ja menetelmiä, tutkimalla minkälaisia objektinseurantasovellukset ovat ja selvittämällä miten liikeinformaatiota voi hyödyntää objektinseurannassa. Lisäksi tarkoitus on empiirisesti tutkia lohkotäsmäykseen perustuvien erilaisten etsintä- ja täsmäystekniikoiden keskinäistä paremmuutta.

Luvussa 2 tarkastellaan objektin liikettä, objektinseurannan sovellusalueiden parametreja ja esitellään joitain objektinseurantasovelluksia. Luvussa 3 kuvataan objektinseuranta-järjestelmän perusrakennetta, tarkastellaan objektinseuranta moniobjektinseurannan kannalta ja tarkastellaan objektinseurantaan käytettäviä menetelmiä. Luvussa 4 tarkastellaan objektin etsintään soveltuvaa lohkotäsmäysmenetelmää esittelemällä erilaisia täsmäystekniikoita. Luvussa 5 tarkastellaan muotoaan muuttavan objektin seuranta tarkastelemalla objektin ääriviivoja seuraavia menetelmiä. Tarkastelun kohteena ovat aktiivisen reunan menetelmä ja alueenlaajennustekniikka. Lohkotäsmäyksessä käytettävien täsmäystekniikoiden toimivuutta tutkitaan empiirisesti luvussa 6. Luvussa 7 tehdään päätelmiä saaduista tuloksista.

2. OBJEKTI JA LIIKE

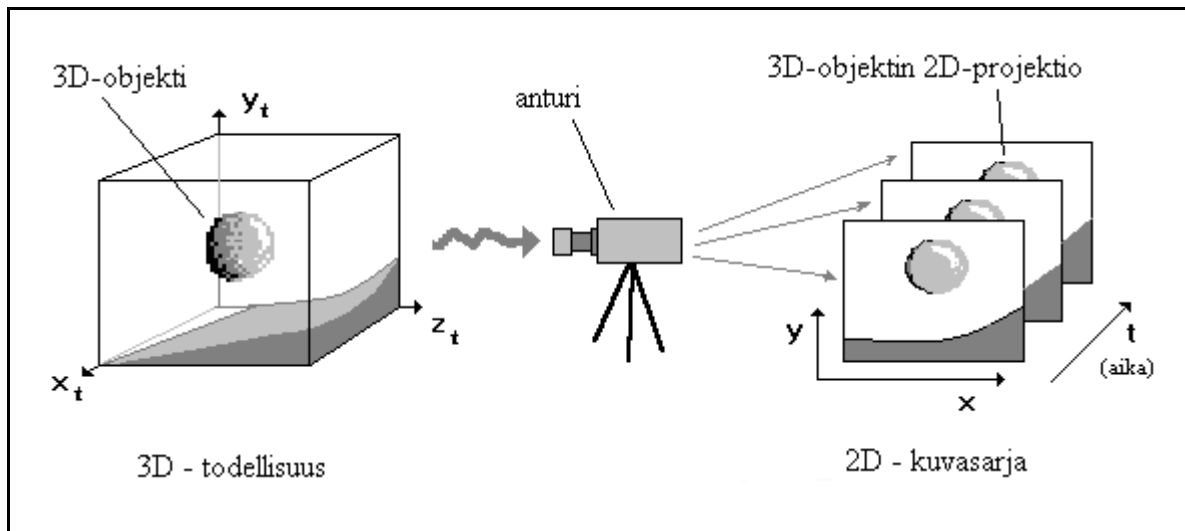
Tässä luvussa käsittelemme objektinseurannan havaintoaineiston muodostamista, tarkastelemme objektinseurantasovelluksiin vaikuttavia parametreja ja esittelemme muutaman objektinseurantasovelluksen.

2.1 Havaintoaineisto

Objektinseurantajärjestelmä tarkkailee ympäristöään *anturin (sensor)* avulla. Anturi voi olla näkyvää valoa lukeva kamera, infrapunavaloa tunnistava kamera, tutka tai muu vastaava tilainformaatiota tallettava väline [CHI94]. Kun anturi tallettaa kuvan kaksiulotteisena, on kyseessä *monokulaarinen (monocular)* kuvainformaatio. Objektinseurantajärjestelmä voi tallettaa ja analysoida myös kolmiulotteista informaatiota ympäristöstään esimerkiksi kahden kameran avulla. Tällöin informaatio on *binokulaarista (binocular)* ja kyseessä *binokulaarinen seuranta (binocular tracking)* [MAK93]. Tätä käytetään esimerkiksi *kolmiulotteisessa konenäössä (3D machine vision)*. Binokulaarisen informaation käsittely on kuitenkin vaikeaa. Tässä tutkielmassa keskitytäänkin monokulaarisen informaation käyttöön. Lukijaa rohkaistaan kuitenkin tarkastelemaan esimerkiksi viitteitä [MAK93] ja [KNC93].

Anturin avulla muodostetaan kolmiulotteisesta tilainformaatiosta kaksiulotteinen projektiio, jonka anturi rekisteröi tietyin aikavälein, esimerkiksi sekunnin välein. Tätä kaksiulotteista projektiota kutsutaan *kuvatasoksi (image frame)*, joka muodostuu *kuvapisteistä* eli *pikseleistä (pixels)*, jotka määrittävät kuvan pisteiden intensiteetin. Anturi rekisteröi kuvatasot aikajärjestykseen, joten kyseessä on *kuvasarja (image sequence)*. Jokaista kuvasarjan kuvatasoa voidaan käsitellä kuten yksittäistä kuvaa. Kuvassa 2.1 nähdään kolmiulotteisen maailman projisointi kaksiulotteisten kuvatasojen muodostamaksi kuvasarjaksi.

Objektinseurannassa *todellisella objektilla (real world object)* tarkoitetaan todellisessa kolmiulotteisessa maailmassa liikkuvaa objektia. Todellinen objekti voi olla esimerkiksi



Kuva 2.1 Kolmiulotteinen ympäristö projisoidaan 2D-kuvatasojen sarjaksi

lentokone, auto tai mikä tahansa objektinseurantajärjestelmän ulkopuolella oleva kohde jonka liikkeitä halutaan seurata. Vastaavasti anturin rekisteröimää todellisen objektin kaksiulotteista projektiota kuvatasolla kutsutaan *seurattavaksi objektiksi* tai yksinkertaisemmin *objektiksi (object)*. Seurattava objekti on järjestelmän sisäinen ilmentymä ja sillä tarkoitetaan siis todellisesta objektista anturin avulla saatuja havaintoarvoja. Koska kuvatasot ovat todellisen maailman kaksiulotteisia projektioita, on seurattava objekti todellisen objektin kaksiulotteinen projektiio.

Anturin tallettama informaatio voidaan ymmärtää myös *tila- ja aikajatkumona (spatio-temporal continuum)*, joka sisältää kaiken minkä kameralla varustettu robotti näkee tietyn ajan kuluessa. Tällöin yksittäinen kaksiulotteinen kuvataso määrittää järjestelmän *näkökentän (FOV, field of view)* kuvatason rekisteröintiajankohtana [OXL94]. Tämä tarkoittaa sitä, että tila- ja aikajatkumo määrittelee näkökentän kokoisen alueen koko tila-avaruudesta. Tällöin objektinseurantajärjestelmä käsittelee ympäristöään vain "näkemänsä" tila- ja aikajatkumon avulla.

Antureita voi olla myös useampi kuin yksi. Tällöin kaikkien antureiden tuottama havaintoinformaatio täytyy yhdistää. Esimerkiksi kuusi kameraa voi tarkastella kukin omaa aluettaan ympäristöstä. Tällöin kaikkien kameroiden näkökentistä voidaan muodostaa yksi yhtenäinen näkökenttä. Havaintoarvojen yhdistämistä kutsutaan *datafuusioksi (data fusion)* [CHI94].

Taulukko 2.1 Objektinseurannan sovellusalueiden parametrit

Muuttuja	Arvot
Objektin muoto	tuntematon/tunnettu
Objektin liike	tuntematon/tunnettu
Objektien lukumäärä	yksi/useita
Seurannan reaaliaikaisuus	kyllä/ei
Kameran liike	tuntematon/tunnettu
Ympäristön häiriöt	tuntematon/tunnettu

2.2 Objektinseurannan sovellusalueiden parametreja

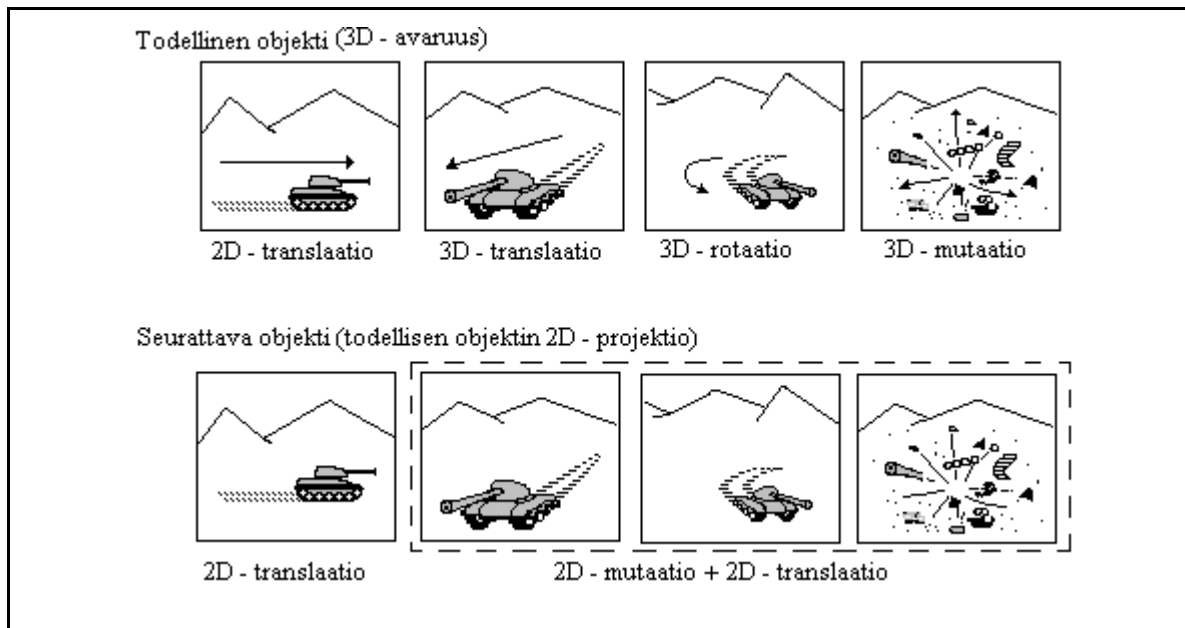
Objektinseurannan olemusta analysoimalla voidaan erottaa kuusi objektinseurannan toimintaa määrittävää parametria. Nämä parametrit esitellään taulukossa 2.1.

Todellisen objektin muoto

Todellinen objekti voi olla *kiinteä (rigid)*, kuten auto, lentokone tai panssarivaunu. Vastaavasti objekti voi olla *muuttuvamuotoinen (deformable)* [COH93], jolloin objektin ääriviivat eivät ole selkeitä ja ne muuttavat muotoaan. Tämänkaltaisia todellisia objekteja ovat esimerkiksi pilvet. Muuttuvamuotoisten objektien seuranta on yleensä kiinteiden objektien seurantaa vaikeampaa, sillä järjestelmän on varauduttava mukautumaan muodon muutoksiin.

Objektin muoto voi olla myös rakenteellisesti monimutkainen. Tällöin objekti koostuu useista itsenäiseen liikkeeseen kykenevästä osasta [PNG94]. Tämän kaltainen rakenne on esimerkiksi ihmisellä, jonka kädet ja jalat liikkuvat jossain määrin itsenäisesti. Rakenteellisesti monimutkaisen objektin voidaan sanoa olevan muuttuvamuotoinen, sillä objektin muoto muuttuu, joskin säännöllisesti, rakenteiden liikkuessa.

Objektinseurannan kannalta on tarpeellista tietää, seurataanko ennalta tunnistamattomia objekteja vai seurataanko vain jo tunnettuja objekteja. Ennalta tunnistamattomien objektien seurannassa ei voida käyttää objektille ominaista informaatiota, joka on usein hyödyllistä



Kuva 2.2 Kolmiulotteinen liike on mutaatiota, ellei se ole puhdasta translaatiota

seurannan kannalta. Esimerkiksi jos tunnetaan miten muuttuvamuotoinen objekti muuttuu seurannan aikana, voidaan rajoittua seuraamaan vain ennalta tunnettuja objektin muotoja.

Objektin liike

Objektille voidaan määrittellä kolme liikkeen lajia: *translaatio* (*translation*), *rotaatio* (*rotation*) ja *mutaatio* (*mutation*). Kuvassa 2.2 nähdään miten todellisen objektin liikkeen lajit määritellään seurantajärjestelmän kannalta.

Yksinkertaisin objektin liike on translaatio, jolla tarkoitetaan objektin siirtymistä paikasta P1 paikkaan P2 ajassa t . Kaksiulotteisessa tapauksessa kyseessä on *puhdas translaatio* (*pure translation*), jolloin objektin siirtyminen on ortogonaalista tila- ja aikajatkumon aika-akselin suhteen [WEN90]. Tällöin myös objekti voidaan ymmärtää kaksiulotteisena. Tämä on yksinkertaisin liikkeen muoto siksi, että kamera "näkee" ympäristönsä kaksiulotteisena. Todellisen objektin liike on kuitenkin usein kolmiulotteista. Kun kamera tallettaa tila-avaruuden kaksiulotteisena, näkyy todellisesta objektista vain kaksiulotteinen projektiio. Tällöin kameraa lähestyvän objektin koko näyttää kasvavan ja vastaavasti kamerasta etääntyvän objektin koko pienenee. Näin ollen jos todellisen objektin liike ei ole puhdasta translaatiota, niin sen liike on järjestelmän kannalta mutaatiota eli muodon muutosta.

Toinen objektin liikkeen laji on rotaatio, jolla tarkoitetaan objektin kiertymistä oman akselinsa ympäri. Jos kiertymäakseli on kohtisuorassa kuvatason suhteen, kyseessä on *puhdas rotaatio (pure rotation)* [WEN93]. Yleisesti kyseessä on kolmiulotteinen todellinen objekti joka kiertyy kolmiulotteisessa tila- ja aikajatkumossa, ja jonka kaksiulotteinen projektio heijastuu kuvatasoon. Myös rotaatio on järjestelmän kannalta mutaatiota, koska seurattavan objektin muoto muuttuu.

Todellisen objektin mutaatio on laajenemista tai kutistumista. Jos laajeneminen ja kutistuminen on jaksottaista, on kyseessä *sykkivä liike (cyclic motion)* [PNG94]. Esimerkiksi sydämen liike on tässä mielessä mutaatiota. Sykkivää liikettä on myös samaa liikerataa toistava translaatio. Esimerkiksi rakenteeltaan monimutkaisen objektin liike, kuten ihmisen kävely, on usein sykkivää. Mutaatio on tavallaan translaation erikoistapaus, jossa objekti liikkuu laajeten tai kutistuen. Kaksiulotteisen objektinseurannan kannalta kaikki muu paitsi puhdas translaatio, voidaan laskea mutaatioksi.

Tietenkin on huomattava, että kaikki liikkeen lajit saattavat esiintyä yhtäaikaan. Objekti voi esimerkiksi lähestyä kameraa heiluri-liikkeellä kiertyen itsensä ympäri välillä laajeten ja kutistuen. Jatkossa liikelajeista puhuttaessa tarkoitetaan liikettä nimenomaan objektinseurantajärjestelmän näkökulmasta, jolloin liikelajit rajoittuvat translaatioon ja mutaatioon.

Liikelajin lisäksi objektinseurannan kannalta mielenkiintoista on objektin liikerata ja nopeus. Kun objekti liikkuu kohtisuoraan liikesuuntansa mukaisesti tasaisella nopeudella, sanotaan liikkeen olevan suoraviivaista. Vastaavasti jos objekti poikkeaa selvästi liikeradaltaan tai kiihdyttää tai hidastaa vauhtiaan, sanotaan objektin tekevän *manööverin (manoeuvre)*. Suoraviivaisesta liikkeestä poiketen manöövereitä tekevän objektin liikettä on hankala ennustaa. Objektin liikerata voi olla ennalta tunnettu tai se voidaan olettaa. Esimerkiksi voidaan ennalta tietää, että objekti liikkuu suoraviivaisesti. Tämä oletus voidaan tehdä, kun tiedetään objektin fyysisiä ominaisuuksia. Esimerkiksi, jos todellinen objekti on suoraa tietä eteenpäin liikkuva auto, voidaan olettaa että seurattava objekti ei oleellisesti poikkea suoraviivaiselta liikeradaltaan.

Objektien lukumäärä

Seurattavien objektien lukumäärä vaikuttaa järjestelmän toimintavaatimuksiin. Yksinkertaisimmillaan seurattavia objekteja on vain yksi. Seurattaessa vain yhtä objektia, ei järjestelmä varaudu toisten samankaltaisten objektien ilmaantumiseen näkökenttään. Rajoitteena on myös se, että muut liikkuvat objektit eivät häiritse seurattavaa objektia. Yhden objektin seurannassa yleensä ympäristön on oltava tunnettu, sillä ennalta tuntemattomassa ympäristössä ei objektien lukumäärää voida rajoittaa yhteen. Yhden objektin seuranta on kuitenkin yksinkertaisuuden vuoksi sovelias, kun ympäristö on tunnettu ja tiedetään, että uusia seurattavia objekteja ei ilmesty näkökenttään. Eräs tämän kaltainen sovellus on sydämen seuranta paikallaan olevalla kameralla [COH93].

Useimmissa objektinseurantasovelluksissa ei voida tehdä rajausta objektien lukumäärän suhteen. Tällöin kyseessä on *moniobjektiseuranta* (*MTT, multi-target tracking*). Useamman seurattavan objektin tapauksessa järjestelmän on kyettävä erottamaan eri objektit toisistaan. Tämä on ongelmallista erityisesti silloin kun objektit ovat muodoltaan samankaltaisia ja seurattavia objekteja on paljon.

Moniobjektiseuraa voidaan hyödyntää myös yhden todellisen objektin seurannassa, kun objektin rakenne on monimutkainen. Esimerkiksi jos seurattava todellinen objekti on ihminen, niin objektinseurannan kannalta seurattavia objekteja on useita: pää, kädet ja jalat. Tämä tarkoittaa sitä, että seurattava todellinen objekti saattaa sisältää seurattavia *aliobjekteja* (*tokens*), jotka kuuluvat todelliseen objektiin [ZNG94]. Jokaisen aliobjektin seurantaan käytetään samoja tekniikoita kuin itsenäisten objektien seurantaan.

Aliobjektin käsite on tarpeellinen, sillä aliobjektien keskinäisen liikkeen perusteella on mahdollista tunnistaa aliobjektien muodostama kokonainen todellinen objekti [PNG94]. Sovellusalueesta riippuu, tarvitaanko aliobjekteja. Jos esimerkiksi seurataan laivoja kymmenen kilometrin päästä tutkalla, ei aliobjektien käyttö liene tarpeellista todellisen objektin muuttumattomuuden ja pistemäisen olomuodon vuoksi. Tämä on kuitenkin sovelluskohtainen ongelma.

Seurannan reaaliaikaisuus

Objektinseurantajärjestelmän toteutuksen kannalta on merkityksellistä, että pyritäänkö objekteja seuraamaan reaaliajassa. Reaaliaikaisessa järjestelmässä on havaintoaineiston muodostus ja sen käsittely tapahduttava siten, että järjestelmä kykenee määrittämään todellisen objektin sijainnin reaaliajassa. Reaaliaikaisten objektinseurantajärjestelmien tärkeä ominaisuus on laskennan tehokkuus.

Erilaiset valvontajärjestelmät toimivat usein reaaliajassa. Tästä esimerkkinä voidaan antaa liikenteenvalvontasovellus, jonka avulla ajoneuvoja voidaan seurata reaaliajassa [BUL93]. Reaaliaikaisuus ei kuitenkaan aina ole välttämätöntä objektinseurannalle. Esimerkiksi sovelluksissa, joissa pyritään analysoimaan kuvasarjassa esiintyvää objektia, on laskennan nopeutta tärkeämpi ominaisuus seurannan tarkkuus.

Kameran liike

Objektinseurantasovelluksen kannalta merkityksellistä on myös se, onko kamera kiinteästi jalustallaan vai kykeneekö kamera siirtymään. Tämä siksi, että yleensä on tarkoituksenmukaista laskea seurattavien objektien liike suhteessa kameraan. Vastaavasti, jos tiedetään, että kamera on kiinteästi paikallaan, niin ympäristön muuttumattomuutta voidaan hyödyntää seurannassa. Esimerkiksi liikenteenlaskentasovelluksissa kamera on usein kiinteästi paikallaan.

Kameran toimintaan liittyy käsite *objektiin lukittuminen*. Objektiin lukittumisella tarkoitetaan sitä, että kun objektinseurantajärjestelmä on havainnut seurattavan kohteen, se pyrkii suuntaamaan kameran sitä kohti. Tällöin seurattava objekti löytyy näkökentän keskipisteestä. Objekti on tällöin *kiinnostuspisteessä (focus of attention)* [ENN95], jolla tarkoitetaan näkökentän aluetta, joka on seurannan kannalta mielenkiintoisempi kuin muu alue näkökentässä. Näin objektinseurantajärjestelmä voi tarkastella kiinnostuspisteessä olevaa informaatiota huolellisemmin kuin muuta informaatiota näkökentässä. Esimerkiksi ihmisen silmä toimii tällä periaatteella. Yleisemmällä tasolla kiinnostuspisteen ei tarvitse olla näkökentän keskipiste, vaan mikä tahansa alue näkökentässä voi olla kiinnostuspiste.

Ympäristön häiriöllisyys

Yksittäisen objektin kannalta ympäristöä on kaikki muu osa tilasta objektin ulkopuolella. Ympäristö voidaan ymmärtää objektin taustana, jonka "päällä" objekti liikkuu. Esimerkiksi jos objekti on valkoinen suorakaide joka liikkuu tasaisen harmaalla taustalla, voidaan objekti erottaa taustastaan yksiselitteisesti. Objektinseurannan todellisten sovellusalueiden kohdalla ympäristö on harvoin näin triviaali. Tausta ja objekti saattavat olla hyvin samankaltaisia, jolloin objekti ei erotu yksiselitteisesti ympäristöstään. Lisäksi todellisessa ympäristössä voi olla häiriötekijöitä, jotka vaikeuttavat objektin seuranta.

SNR-suhteella (Signal-to-Noise-Ratio) tarkoitetaan signaalin suhdetta häiriöön. Se kertoo miten hyvin merkittävä informaatio erottuu häiriöstä. Mitä alhaisempi SNR-suhde, sen häiriöllisempi on ympäristö. Jos ympäristön häiriö tunnetaan, on häiriön vaikutusten eliminointi yleensä helpompaa. [BET94]

2.3 Objektinseurannan sovelluksia

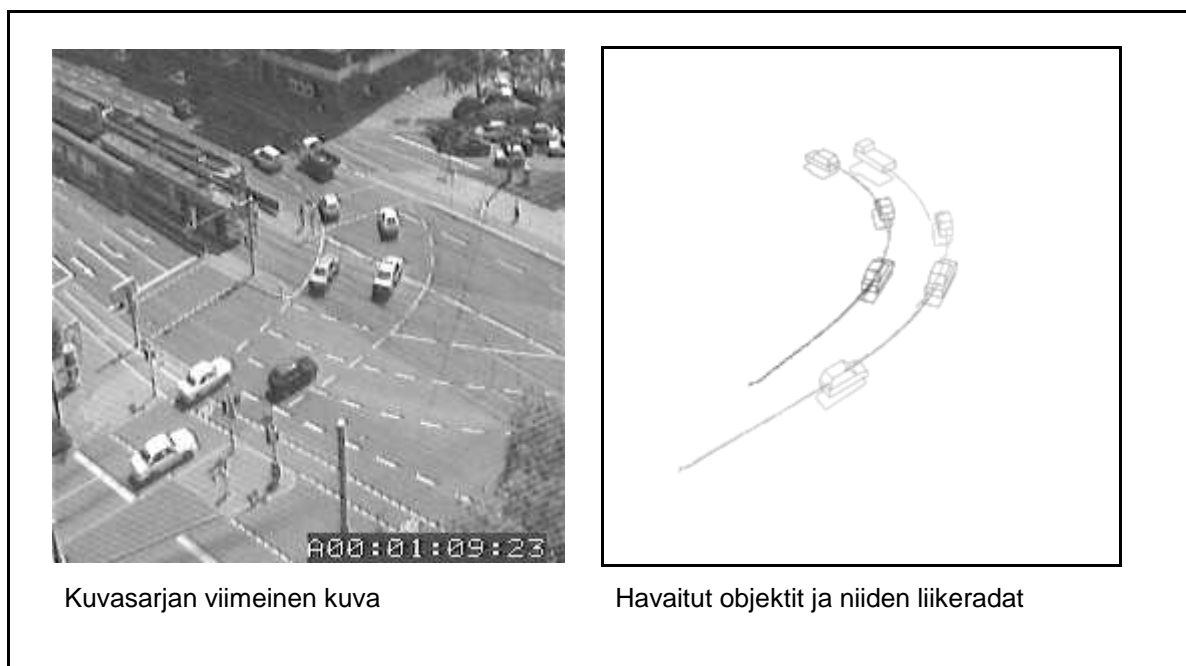
Tässä kuvattavat objektinseurannan sovellukset on lueteltu seurattavan objektin ominaisuuksien mukaan. Alussa esitetyt sovellukset perustuvat kiinteisiin todellisiin objekteihin, kuten liikenteen seurannassa ajoneuvot. Viimeisenä esiteltävissä sovelluksissa todellinen objekti on vaikeasti määriteltävissä.

Liikenteen seuranta

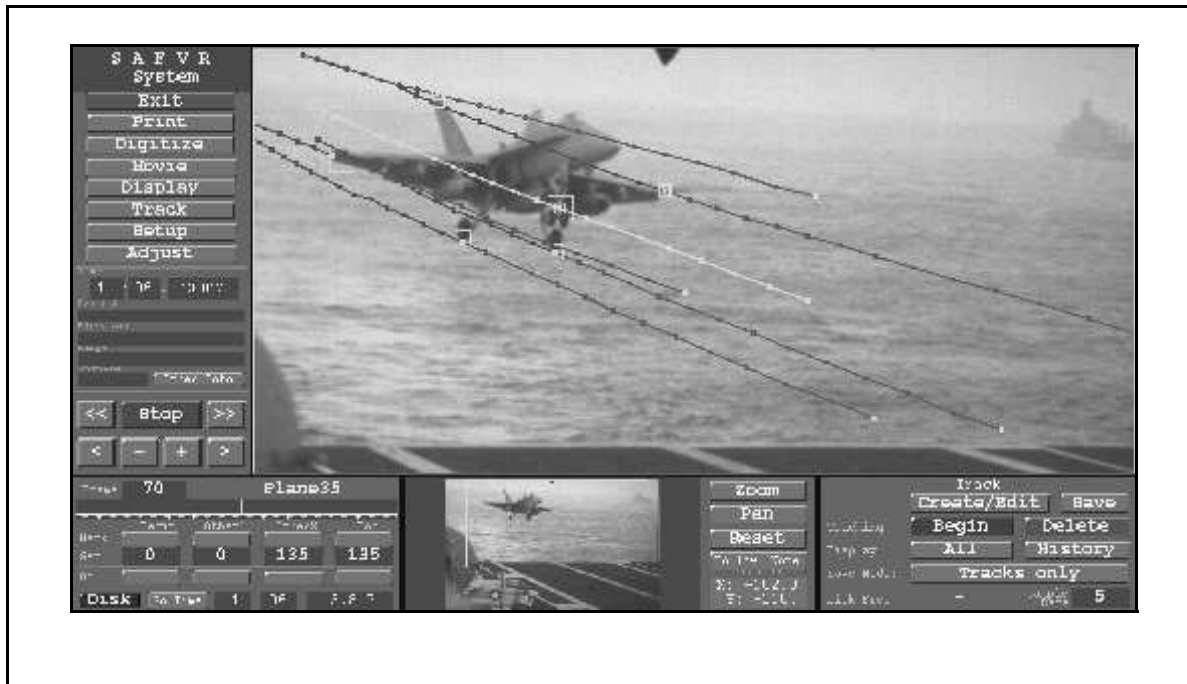
Bristolin yliopistossa Englannissa on kehitetty ajoneuvojen liikeratojen seurantaan soveltuvaa järjestelmää [BUL93]. Sen avulla voidaan seurata yksittäisten ajoneuvojen liikkeitä risteysalueella tai pysäköintialueella. Järjestelmän anturina toimii kiinteästi kohdealueen yläpuolelle asetettu kamera. Seurattavat objektit havaitaan niiden liikkeen perusteella vertailemalla anturin antamaa kuvaa ennalta talletettuun taustakuvaan, jossa kohdealue on kuvattuna ilman autoja. Objektin muoto määräytyy havaitun liikkuvan alueen mukaan.

Saksalainen Kilger [KIL93] esittelee reaaliajassa toimivan liikenteenseurantajärjestelmän (*traffic monitoring system*), joka kykenee ruuhkaisen liikenteen seurantaan. Liikenteenseurannan tarkoituksena on laskea ohi kulkevien autojen lukumäärä ja mahdollisesti tunnistaa minkälaisia ajoneuvoja tiellä kulkee. Seurattavat objektit ovat kiinteitä ja niiden liike on mutaatiota, sillä todellisten objektien liike ei ole puhdasta translaatiota. Ruuhkainen liikenne asettaa vaatimuksia seurannan toteutukseen. Järjestelmän on kyettävä erottamaan hyvin lähellä toisiaan kulkevat ajoneuvot. Tämä on ratkaistu siten, että yksittäisestä autosta seurataan vain sen etuosaa, joka erottuu tarkasteltavasta kuvasta parhaiten. Tämä taas perustuu tietämykseen kameran sijainnista suhteessa liikkeeseen. Tiedetään, että autot kulkevat kohti paikallaan olevaa kameraa.

Vastaavanlainen systeemi on ollut tekeillä Tanskassa [SOR93]. Tässä systeemissä seurattavat objektit pyritään seurannan lisäksi tunnistamaan. Ajoneuvot jaetaan kolmeen eri ryhmään, joita ovat kuorma-autot, henkilöautot ja moottoripyörät. Järjestelmään on talletettu kaikista ajoneuvoista kolmiulotteinen malli, jota voidaan hyödyntää tunnistuksessa. Tällöin menetelmän sanotaan olevan *malliperusteinen (model-based)*. Saksalainen Dieter Koller on kehittänyt vastaavanlaisen malleihin perustuvan liikenteenseurantasovelluksen, josta nähdään esimerkki kuvassa 2.3.



Kuva 2.3 Malleihin perustuva ajoneuvojen liikeratojen seuranta [KOL96]



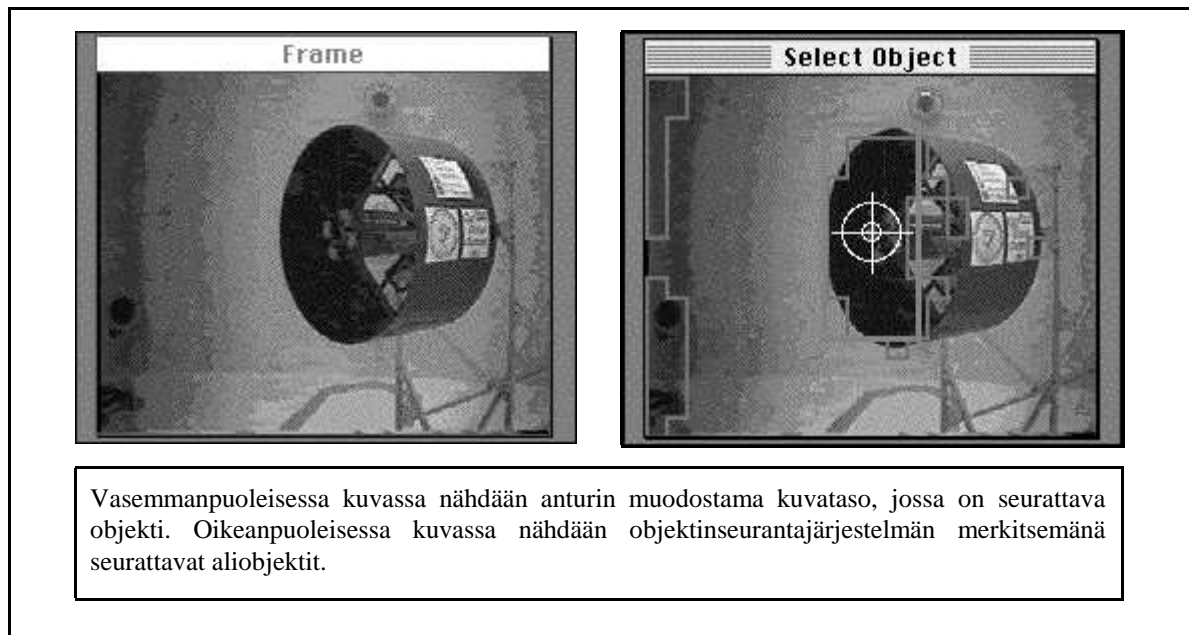
Kuva 2.4 Laskeutuvan lentokoneen liikkeen analysointi objektinseurannalla [KOH96]

Laskeutuvan lentokoneen liikkeen analysointi

Aliobjektien seurantaan perustuu myös Amerinex Applied Imaging nimisen yrityksen Yhdysvaltain Laivastolle kehittämä järjestelmä [KOH96]. Sen avulla lentotukialukselle laskeutuvan lentokoneen eri osien, kuten siivenkärkien ja pyörien, liikeratoja voidaan tallettaa ja analysoida. Seurattavat objektit ovat siis kiinteitä aliobjekteja, joiden liikkeiden perusteella voidaan analysoida koko todellisen objektin käyttäytymistä. Todellisena objektina voi olla harjoittelevan pilotin ohjaama lentokone, jonka laskeutumisen onnistumista kokenut pilotti voi analysoida. Vastaavasti ongelmatilanteet on mahdollista käsitellä yksityiskohtaisesti. Kuvassa 2.4 nähdään järjestelmän käyttöliittymä. Laskeutuvasta lentokoneesta voidaan ottaa lähikuva, jossa sen lentokoneen eri osien liikeratoja voi tarkastella myös visuaalisesti.

Avaruuskamera

Marylandin yliopistossa on kehitetty avaruudessa toimivaa kameraa, joka suuntautuu itsenäisesti ennalta määrättyyn kohteeseen [SEE96]. Näin ollen esimerkiksi avaruusaluksen ulkopuolella korjaustöissä oleva astronautti ei tarvitsisi toista astronauttia huolehtimaan



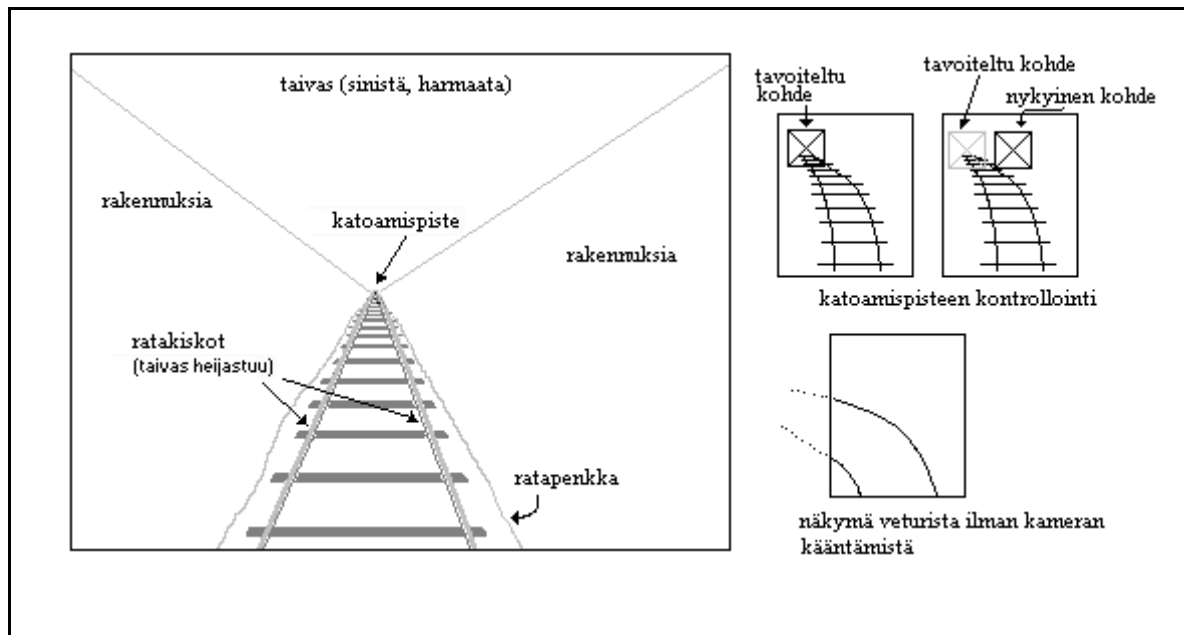
Kuva 2.5 Avaruuskamera [SEE96]

kameran suuntaamisesta työkohteeseen (kuva 2.5). Objektinseuranta perustuu tässä kiinteiden aliobjektien seurantaan, joiden keskinäisen liikkeen perusteella kamera ohjautuu osoittamaan kappaleen keskipistettä kohti. Avaruuskameran seuranta on reaaliaikaista ja sen kameran liike perustuu objektiin lukittumiseen.

Valvontajärjestelmät

Tutkakuvan analysointiin perustuu myös esimerkiksi lentokoneiden lentoratojen analysointi lennonjohtojärjestelmissä. Objekteja seuraamalla voidaan päätellä niiden nopeus ja suunta. Jos kahden tai useamman objektin ennustetut lentoradat kohtaavat, voi lennonjohtojärjestelmä antaa automaattisesti varoituksen törmäysuhasta. Vastaavaa periaatetta voi hyödyntää myös laivaliikenteen seurannassa.

Esimerkkinä hieman toisenlaisesta valvontajärjestelmästä voidaan antaa objektinseurantaan perustuva valvontakamera. Tässä sovelluksessa kamera tarkkailee ympäristöänsä ja havaittuaan epänormaalia liikettä se kohdistaa kameran ja mahdollisesti jonkin muun välineen kohti havaitun liikkeen aiheuttajaa. Epänormaalia liikettä voisi olla esimerkiksi pankin valvontasovelluksessa hyppääminen pankkitiskin yli. Epänormaalien liikkeen mallintaminen on kuitenkin varsin hankalaa [PET93].



Kuva 2.6 Junaratakamerassa kamera käännetään kohti radan katoamispistettä [UKA93]

Junaratakamera

Japanilaisessa rautateiden tekniseen tutkimukseen keskittyvässä laitoksessa (*RTRI, Railway Technological Research Institute*) kehitetään "älykäästä junaa", joka kykenee toimimaan autonomisesti [UKA93]. Koska junassa ei ole kuljettajaa, on sen itse kyettävä seuraamaan radan turvallisuutta. Tämä tapahtuu visuaalisella järjestelmällä, jonka yksi osatehtävä on veturiin asennetun kamerasuuntaaminen radan suuntaisesti.

Objektinseurannan kannalta seurattavana objektina on tällöin junaradan *katoamispiste* (*vanishing point*); siis piste, jossa junarata katoaa horisonttiin. Käytännössä tätä objektia on vaikea määrittellä sen pistemäisen olomuodon vuoksi, joten seurattavana objektina on junan kiskot. Itse asiassa seurattavia objekteja on kaksi, sillä objektinseuranta perustuu tässä geometristen käyrien seurantaan, joiksi junan kiskot määritellään. Seuranta on muutenkin mielenkiintoista, sillä tässä objekti ei liiku ollenkaan, vaan kamera liikkuu. Näin ollen objektin liike on suhteellista kamerasuuntaamiseen. Kiskojen kolmiulotteista liikettä ei seurata, vaan seurattava liike on translaatiota. Seurannan apuna voidaan käyttää rautateille ominaista informaatiota, sillä junaradan ympäristö tunnetaan. Esimerkiksi tiedetään, että kiskot heijastavat taivaan väriä. Näin ollen ne voidaan erottaa ratapenkan väristä. Junaratakameraa havainnollistetaan kuvassa 2.6.

Sydämen kaikumittaus

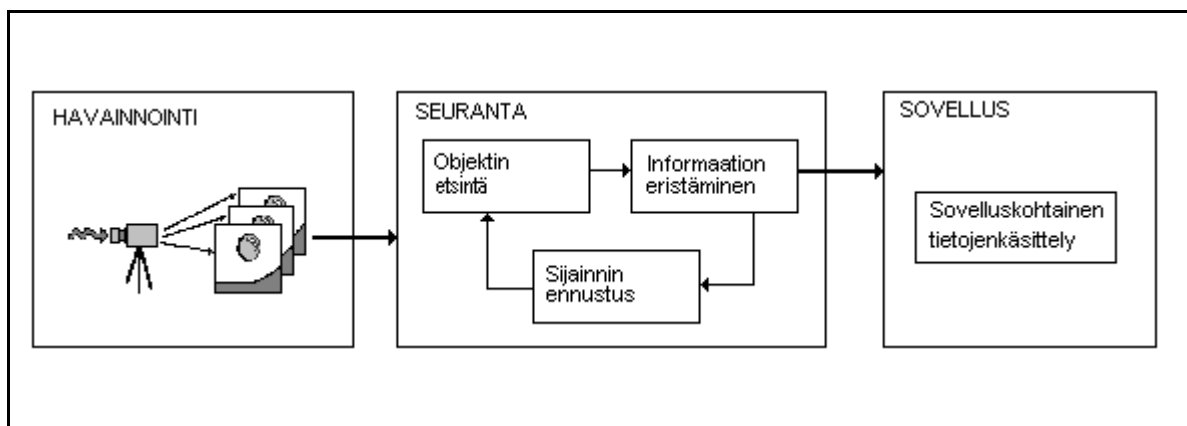
Lääketieteellisissä sovelluksissa seurattava objekti ei useinkaan ole kiinteä. Kohteena oleva objekti on yleensä elävä organismi, jonka muoto vaihtelee usein paljon. Objektin muodon muuttuminen eli mutaatio tapahtuu kuitenkin yleensä tasaisesti. Japanilaiset Fujimura, Yokoja ja Yamamoto esittelevät sovelluksen, jolla voi seurata muuttuvamuotoisen objektin ääriiviivoja kuvasarjassa [FUJ93]. Esimerkkinä he käyttävät *sydämen kaikumittauksella (echocardiogram)* muodostettua kuvasarjaa sydämen liikkeistä. Lääketieteellisissä sovelluksissa ongelmana on usein seurattavan objektin sijainnin vuoksi aiheutunut ympäristön häiriöllisyys. Myös sydämen liikkeiden tarkastelussa voimakas häiriöllisyys mutatoi objektia sen normaalin liikkeen lisäksi.

3. OBJEKTINSEURANNAN PERUSTEITA

Tässä luvussa tarkastelemme objektinseurantajärjestelmän rakennetta ja järjestelmän eri osien toimintaa. Tarkastelemme myös moniobjektiseuraa. Lisäksi luettelemme erilaisia objektinseurantaan soveltuvia menetelmiä kuvasarjassa tapahtuvan liikkeen analysoimiseksi.

3.1 Objektinseurantajärjestelmän rakenne

Objektinseurantajärjestelmä koostuu kolmesta moduulista, jotka ovat havainnointimoduuli, seurantamoduuli ja sovellusmoduuli. Tämä rakenteellinen jako selviää kuvasta 3.1. Havainnointimoduulissa anturi muodostaa kolmiulotteisesta todellisuudesta kaksiulotteisten kuvatasojen muodostaman kuvasarjan. Havainnointimoduuli voi myös päättää seurannan aloituksesta. Seurantamoduulin tehtävänä on määrittää seurattavan objektin sijainti kussakin kuvatasossa. Seurantamoduuli jakautuu kolmeen alimoduuliin, joiden tehtävänä on objektin sijainnin ennustaminen tarkasteltavasta kuvatasosta, objektin sijainnin määrittäminen ja informaation eristäminen objektista. Objektista eristettävää informaatiota käytetään ennustuksen apuna tai se annetaan syötteenä sovellusmoduulille, joka toteuttaa sovelluskohtaisen tietojenkäsittelyn. Eristettävä informaatio voi olla esimerkiksi objektin ääriviivat, koko, sijainti koordinaatteina, nopeus tai se voi olla koko objekti irrotettuna ääriviivojen tarkkuudella.



Kuva 3.1 Objektinseurantajärjestelmän rakenne

Seurannan käynnistäminen

Seuranta käynnistetään analysoimalla kuvasarjan ensimmäistä kuvatasoa. Näin ollen käytettävissä ei ole aikaisempaa informaatiota, joten segmentointi perustuu pelkästään spatiaaliseen informaatioon. Tällöin kuvatasoa voidaan analysoida yksittäisen kuvan segmentointiin käytetyillä menetelmillä, joita löytyy viitteestä [FRÄ95]. Sovelluksesta riippuen voidaan myös olettaa, että seurattavat objektit määritellään eksplisiittisesti. Esimerkiksi interaktiivisessa sovelluksessa objektin sijainti ensimmäisessä kuvatasossa voidaan "piirtää" hiirellä suoraan kuvaruudulle tai objekti voidaan määritellä suoraan sovelluksen syöttötietona. Tällöin järjestelmän tarkoituksena on yleensä seurata ainoastaan syötteenä annettua objektiä.

Objektien seuranta vaatii yleensä aina objektin mallintamista. Objektista on siis oltava jonkinlainen malli, jotta sitä voidaan vertailla kuvasarjan kuvatasoihin. Objektin mallina voi olla esimerkiksi matriisi, joka sisältää objektin pikseliarvot. Tällöin seuranta käynnistetään etsimällä objektin mallin sijainti ensimmäisestä kuvatasosta. Muodollisemmin määriteltynä objektin malli on funktio, joka identifioi objektin. Objektin mallin ei välttämättä tarvitse olla staattisesti ennalta määritelty, vaan järjestelmä voi mallintaa seurattavan objektin dynaamisesti [KRI95]. Tällöin kyseessä on ennalta tuntemattoman objektin seuranta.

Monissa valvontasovelluksissa seuranta aloitetaan vasta kun seurattava objekti havaitaan. Tällöin tarkoituksena on havaita objektin ilmestyminen näkökenttään. Seurattavan objektin havainnointi voi perustua liikkeen havaitsemiseen. Tällöin järjestelmä tarkkailee näkökentässä tapahtuvaa liikettä, ja jos se havaitsee objektin mallia vastaavaa liikettä, se aloittaa seurannan.

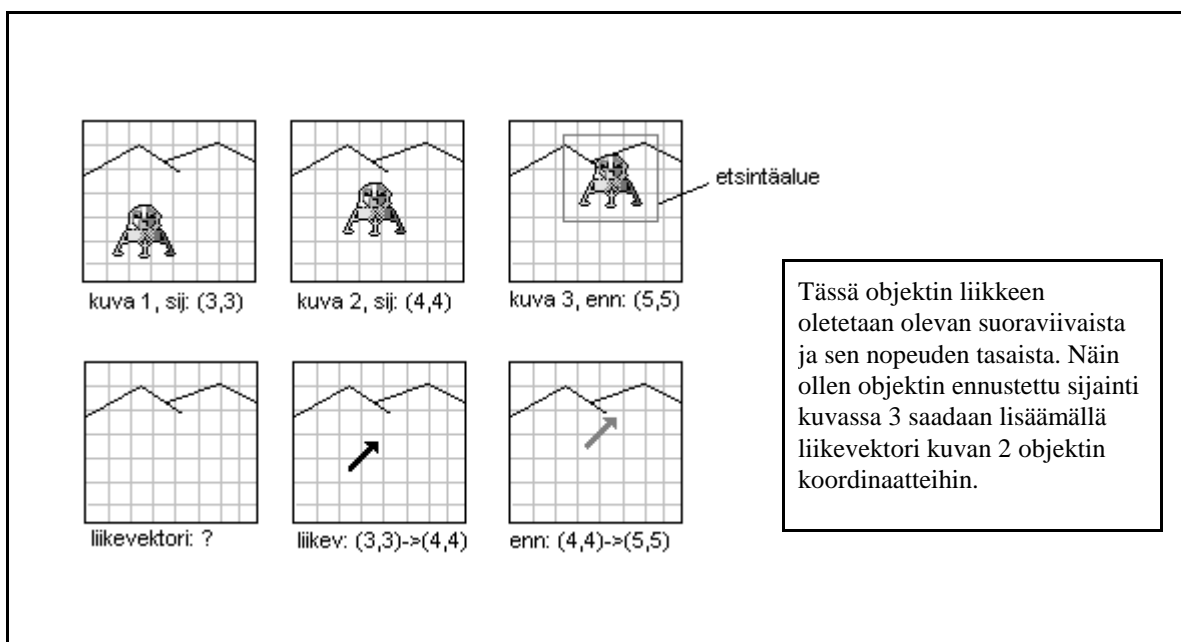
Seurannan ylläpito

Yksinkertaisin ylläpidon toteutus on menetellä jokaisen kuvasarjan kuvan kohdalla kuten seurannan aloituksessa tehtiin. Tämä ei kuitenkaan ole kovin tehokas ratkaisu, varsinkaan jos seurattava objekti määritellään manuaalisesti. Parempi vaihtoehto on ratkaisu, jossa

objektien seurannan ylläpito perustuu peräkkäisten kuvatasojen välisten *vastaavuuksien* (*correspondence*) analysoimiseen [LEE93]. Analysoinnin tarkoituksena on löytää edellisessä kuvatasossa määritelty yksittäinen objekti tarkasteltavasta kuvatasosta vertailemalla objektin mallia suoraan kuvatasoon. Tätä objektin etsintää kuvatasosta kutsutaan usein *täsmäykseksi* (*matching*). Objektin mallia vastaava funktio voidaan määrittellä kustannusfunktioiksi, siten että funktio palauttaa objektin ja vertailtavan kuvatason alueen välisen eroavaisuuden. Tällöin objektin sijainnin määrittelee kustannusfunktion minimi. Vertailu voidaan suorittaa pikselitasolla.

Liikeinformaation hyödyntäminen

Objektinseurannassa todellisen objektin liike on usein fysiikan lakien rajoittamaa. Lisäksi usein määritellään, että objektin siirtymä kahden kuvatason välillä ei ole suuri. Näiden oletusten perusteella voidaan tehdä oletus, että objekti löytyy edellisessä kuvatasossa olevan sijaintinsa läheisyydestä. Liikeinformaatiota hyödyntämällä voidaan lisäksi ennustaa objektin mahdollinen siirtymä. Tämä tapahtuu laskemalla kahdessa edellisessä kuvatasossa täsmättyjen objektien sijaintien perusteella objektin suunta ja nopeus. Objektin liikkeen ennustusta kutsutaan myös *liikkeen estimoinniksi* (*motion estimation, motion prediction*). Kuvassa 3.2 visualisoidaan liikkeen ennustusta kahden peräkkäisen kuvatason avulla.



Kuva 3.2 Liikkeen estimointi objektinseurannassa

Objektin liikkeen suunta ja nopeus ilmaistaan *liikevektorina* (*motion vector*), joka osoittaa liikkeen suuntaan ja jonka pituus kertoo objektin nopeuden. Kun liikevektori on määritelty, voidaan objektia etsiä vain ennustetun sijainnin läheisyydestä liikevektorin osoittamasta suunnasta. Näin etsintä tehostuu, sillä objektia ei tarvitse etsiä koko kuvatasosta. Kuvatason aluetta josta objektia etsitään kutsutaan *etsintäalueeksi* (*search window*).

Peräkkäisten kuvatason tarkastelu voi ulottua pidemmällekin kuin vain kahden peräkkäisen kuvatason tarkasteluun. Koko objektin kulkema liikerata voidaan tallettaa ja sitä voidaan käyttää objektin liikkeen ennustuksen apuna. Edellä mainittua liikerataa kulkevaan yksittäiseen objektiin keskittyvää seurantaprosessia kutsutaan *seurannaksi* (*track*).

Seurannan ongelmia

Seurantaan kuuluu myös mahdollisten ongelmatilanteiden käsittely. Kolmiulotteisessa tilassa liikkuva todellinen objekti saattaa piiloutua kokonaan tai osittain jonkin ympäristön kappaleen taakse. Tällöin objektin kaksiulotteinen projektio muuttuu nopeasti muotoaan ja objektin havaitseminen saattaa olla hankalaa. Objekti saattaa myös kadota kokonaan esimerkiksi häiriöistä johtuen. Tämänkaltaisessa ongelmatilanteessa järjestelmä voi joko lopettaa seurannan tai se voi jäädä tarkkailemaan aluetta missä objekti viimeksi havaittiin ja pyrkiä jatkamaan seurantaa heti kun objekti ilmaantuu. Seuranta voidaan lopettaa silloin, kun objektista ei saada havaintoarvoja tietyn ajan kuluessa. Esimerkiksi silloin kun objekti poistuu näkökentästä.

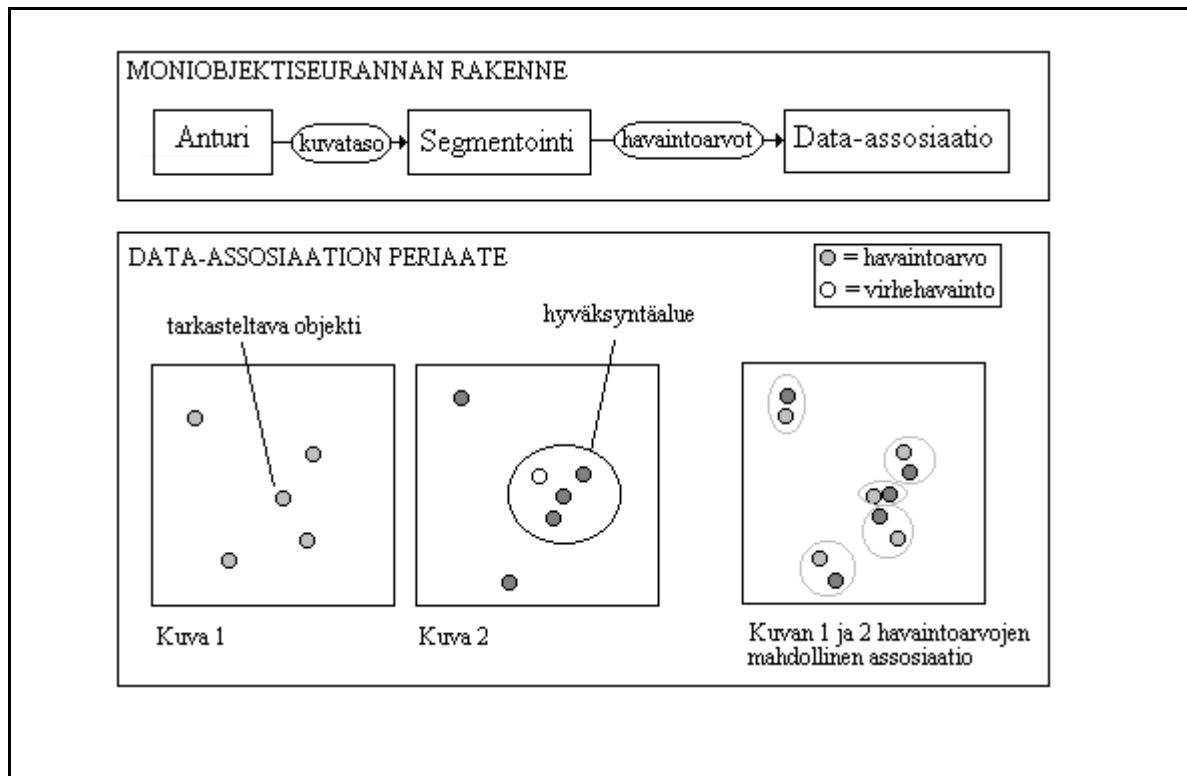
Toinen ongelmatilanne on objektien kohtaaminen. Kahden objektin liikeratojen kulkiessa toistensa poikki, on objektinseurantajärjestelmän kyettävä erottamaan objektit toisistaan. Toisin sanoen objektien seurannat eivät saa sekoittua keskenään. Tämä on usein hankalaa jos objektit ovat toistensa kaltaisia. Kolmas ongelma on seurattavan objektin jakautuminen. Jakautuminen on ongelmallista jos irronneet objektin osat ovat suuria, kuten objektin jakautuessa keskeltä kahtia. Tällöin objektin muoto muuttuu nopeasti. Ratkaisu tähän ongelmaan on yhtenäiseen objektiin perustuvan seurannan lopettaminen ja kahden uuden

osan seurannan aloittaminen. Jos tiedetään, että seurattava objekti jakautuu useasti, on ehkä tarpeellista määrittellä se aliobjektien avulla. Tällöin yksittäisten aliobjektien katoaminen tai liikeradan muutokset eivät välttämättä vaikuta kokonaisen aliobjekteista muodostuvan objektin seurantaan. Yksiobjektiseurannassa ei yleensä tällaisia tilanteita sallita, vaan seurattavan objektin oletetaan olevan ainoa seurattava objekti ja sen oletetaan pysyvän näkökentässä. Useiden objektien liikkeistä aiheutuvat ongelmat kuuluvatkin lähinnä moniobjektiseurantaan.

3.2 Moniobjektiseuranta

Yleisesti moniobjektiseurannan tarkoituksena on useiden objektien seuranta samanaikaisesti. Moniobjektiseurannan perusidea esiteltiin jo vuonna 1955 kun N. Wax keksi, että objektinseurantaan liittyvät ongelmat ovat samankaltaisia kuin ydinfysiikan ongelma, jossa analysoidaan hiukkasen liikkumista väliaineessa [CHI94]. Moniobjektinseurannan perustehtävät ovat näin ollen seurannan aloitus (*track formation*, *track initiation*), seurannan ylläpito (*track maintenance*) ja seurannan lopetus (*track deletion*). Näiden tehtävien tarkoituksena on päätellä milloin objekti ilmestyy ensimmäisen kerran näkökenttään, missä objekti sijaitsee kunakin ajanhetkenä ja milloin objekti katoaa lopullisesti näkökentästä.

Moniobjektiseurannassa järjestelmän seurantamoduuli toimii siten, että kuvatasosta etsitään kaikki objektin mallia vastaavat havaintoarvot ja sitten vertaillaan niitä edellisestä kuvatasosta saatuihin havaintoarvoihin tarkoituksena yhdistää samasta lähteestä aiheutuvat havaintoarvot. Havaintoarvojen muodostusta kutsutaan *segmentoinniksi* (*segmentation*) ja havaintoarvojen yhdistämistä *data-assosiaatioksi* (*data association*). Aiemmin esitetystä rajatusta etsintäalueesta käytetään moniobjektiseurannassa nimitystä *hyväksyntäalue* (*validation gate*) [ZHU94]. Hyväksyntäaluetta käytetään data-assosiaation helpottamisessa ja sillä tarkoitetaan aluetta jonka sisällä havaintoarvojen on oltava, jotta ne voisivat kuulua tarkasteltavalle objektille. Tämä voi perustua oletukseen, että samasta lähteestä aiheutuvat havaintoarvot eivät voi olla kaukana toisistaan peräkkäisissä kuvatasoissa. Kuvassa 3.3 selvitetään moniobjektinseurannan toiminnallista rakennetta ja visualisoidaan havaintoarvojen käsittelyä data-assosiaation avulla.



Kuva 3.3 Moniobjektiseurannan rakenne ja data-assosiaatio

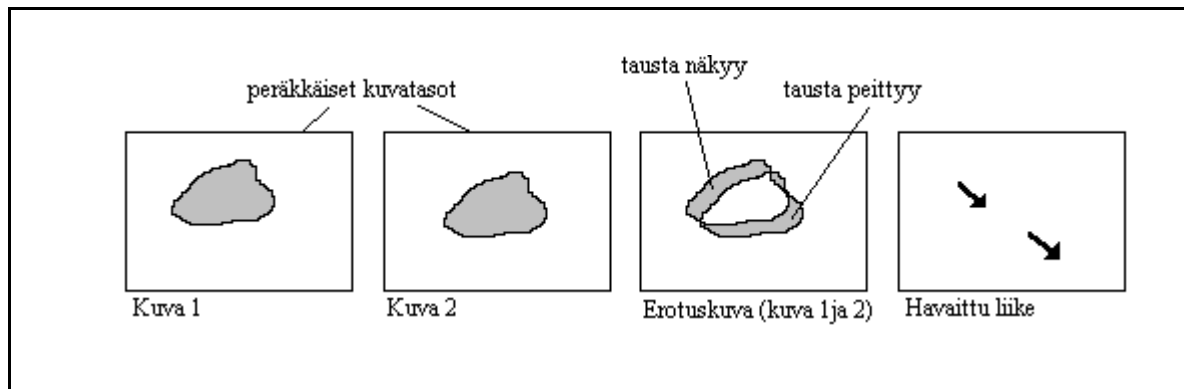
Segmentointi erotuskuvalla

Useasti käytetyt segmentointitekniikat perustuvat objektin mallin ja kuvatason väliseen korrelaatioon. Tällöin etsitään ne kuvatason alueet, joissa objektin malli ja kuvatason alue vastaavat eniten toisiaan. Korrelaatioon perustuvaa objektinseurantamenetelmää käsitellään viitteessä [OXL94].

Segmentoinnissa voidaan hyödyntää myös kahden peräkkäisen kuvatason erotusta. Tätä erotusta kutsutaan erotuskuvaksi (*difference image, frame difference signal*). Erotuskuva muodostetaan pikseleittäin kaavalla

$$I_{\text{Dif}}(x,y) = I_{\text{T}}(x,y) - I_{\text{R}}(x,y) \quad (3.1)$$

jossa I_{R} on referenssikuva ja I_{T} tarkasteltavana oleva kuvataso tarkasteltavana ajanhetkenä. Tällöin muuttumattomina pysyneet alueet eivät talletu erotuskuvaan. Erotuskuva ilmaisee kuvatasojen I_{T} ja I_{R} välillä tapahtuneen liikkeen. Kuvassa 3.4 nähdään esimerkki



Kuva 3.4 Erotuskuvan avulla voidaan havaita kuvien välillä tapahtuva liike [SEF94].

erotuskuvasta. Voidaan myös havaita, että homogeeninen tekstuuri objektin pinnassa vaikuttaa liikkeen havainnointiin erotuskuvan avulla. Tällöin erotuskuvan objektin keskelle muodostuu "reikä". Liikkeen suunta on mahdollista päätellä taustan ilmestymisestä ja peittymisestä. Jos kuvassa 1 tausta näkyy ja kuvassa 2 sen tilalla on objektin reuna, voidaan päätellä objektin etenevän tarkasteltavaan suuntaan.

Referenssikuva on tyypillisesti kuvasarjan edellinen kuvataso, mutta se voi myös olla ennalta talletettu taustakuva, jota verrataan kuvasarjan kuvatasoihin. Tällöin nykyistä kuvatasoa verrataan edellisen kuvatason asemesta kuvatasoon I_{ref} . Nopean segmentoinnin ansiosta staattinen referenssikuva soveltuu hyvin tilanteisiin, joissa tausta on staattinen ja liikettä on paljon. Taustan staattisuus on kuitenkin suuri rajoitus, sillä usein esimerkiksi valaistusolosuhteet muuttuvat. Taustan staattisuusrajoitusta on pyritty lieventämään määrittelemällä adaptiivinen referenssikuva [BUL93], missä referenssikuva päivittyy tietyin aikaväleihin. Tällöin esimerkiksi pysäköintialueelta poistuneen auton paikka muuttuu hitaasti taustan osaksi. Adaptiivisen referenssikuvan ongelmana on hyvin hidas liike. Jos päivitysintervalli on liian nopea, ei esimerkiksi jalankulkija erotu ollenkaan, vaan muuttuu välittömästi osaksi taustaa. Vastaavasti päivitysintervallin ollessa liian hidas, nopeasti kulkevalle ajoneuvolle näyttää jäävän "häntä" hitaan päivittymisen johdosta. Tällöin objektinseuranta on hankalaa objektin vääristyneen muodon vuoksi. [BUL93]

Data-assosiaation ongelmia

Data-assosiaation tarkoitus on yhdistää samasta lähteestä aiheutuneet havaintoarvot kahden peräkkäisen kuvatason välillä. Hyväksyntäalueella saattaa olla useita lähellä toisiaan tai

toistensa päällä olevia havaintoarvoja. Data-assosiaatiassa voidaan käyttää apuna erilaisia estimointitekniikoita. Yleisimmin liikkeen estimointiin käytetään *Kalman-suodinta* (*Kalman Filter*), jonka avulla objektin sijainti ja nopeus seuraavassa kuvatasossa voidaan ennustaa. Kalman-suotimen rajoitteena on se, että objektin käyttäytyminen ja ympäristön häiriöt on pystyttävä mallintamaan ennalta käsin. Kalman-suodinta käytettäessä usein rajoitutaankin vain suoraviivaisen ja tasaisen liikkeen ennustamiseen [WAN94]. Manöövereitä tunnistavaa *laajennettua Kalman-suodinta* (*EKF, Extended Kalman Filter*) käsitellään viitteessä [SWO94].

Data-assosiaatiassa voi esiintyä tilanne, jossa kaksi tai useampi havaintoarvo täsmää yhtäläisesti objektin kanssa. Yksinkertaisin ratkaisu tähän ongelmaan on "*lähin naapuri*"-tekniikka (*nearest neighbour*). Silloin objektin todettua sijaintia lähimpänä oleva havaintoarvo valitaan ilmaisemaan objektin uutta sijaintia. Lähin havaintoarvo ei kuitenkaan aina ole aiheutunut samasta lähteestä. Tällöin tapahtuu virheellinen data-assosiaatio. Etenkin Kalman-suodin on herkkä tämänkaltaisille virheille. Havaintoarvo saattaa olla aiheutunut myös ympäristön tai anturin virheestä. Tällöin kyseessä *virrehavainto* (*false alarm*). Data-assosiaatiota voidaan parantaa erilaisilla menetelmillä. Todennäköisyyteen perustuvassa data-assosiaatiassa (*PDA, probabilistic data association*) jokaiselle havaintoarvolle lasketaan painoarvo. Tämä painoarvo ilmaisee miten todennäköisesti kyseinen havaintoarvo on aiheutunut tarkasteltavasta objektista. [WAN94]

Eräs ratkaisu data-assosiaation ongelmiin on *säde-etsintätekniikka* (*beam search*), jossa jokaiselle yhtäläisesti täsmävälle havaintoarvolle aloitetaan uusi seuranta. Tekniikka perustuu oletukseen, jossa seurannan edetessä virrehavaintoihin perustuvat seurannat loppuvat ja ainoastaan oikeaan havaintoarvoon perustuva seuranta jatkuu. Säde-etsintätekniikan ongelmana on kuitenkin laskennan eksponentiaalinen kompleksisuus. Tähän ongelmaan tosin voidaan soveltaa erilaisia heuristisia tekniikoita. [ZNG94]

Esi- ja jälkikäsitteily

Esikäsitteilyllä (*pre-processing*) tarkoitetaan anturin muodostaman kuvan muokkaamista segmentoinnille soveliaampaan muotoon. Esikäsitteily voi sisältää perinteisiä

kuvankäsittelytekniikoita, kuten kuvan parannusta tai kuvan restaurointia [FRÄ95]. Esikäsittelyn tarkoituksena voi olla esimerkiksi anturin aiheuttaman virheen vaikutusten vähentäminen.

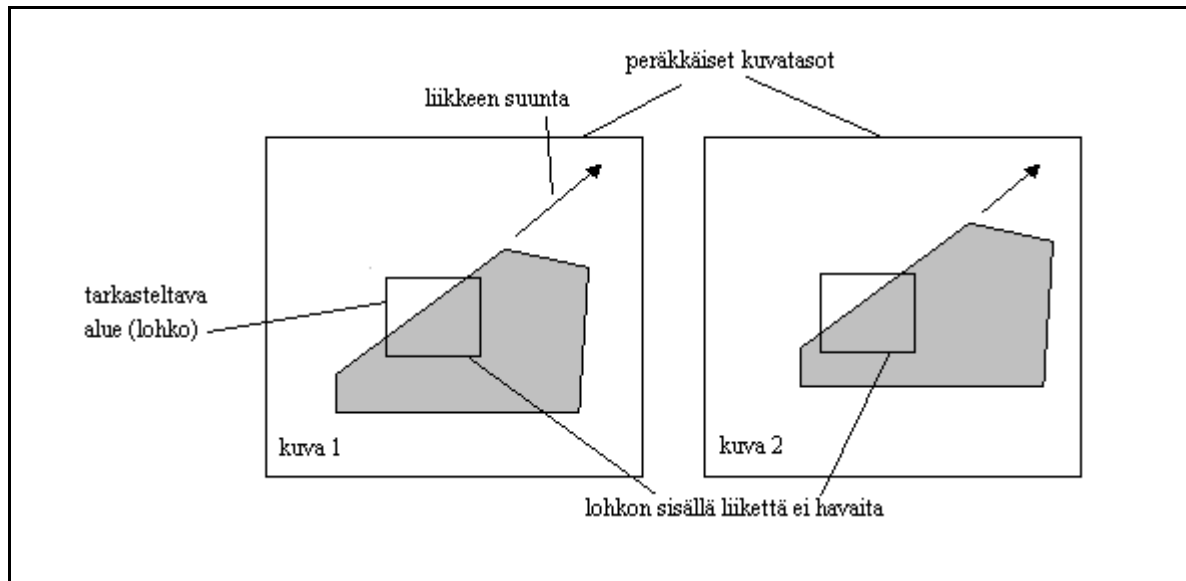
Jälkikäsittelyllä (post processing) tarkoitetaan segmentoinnin tulosten muokkaamista data-assosiaation tarkkuuden parantamiseksi. Esimerkiksi objektien aiheuttamia havaintoarvoja korrelaatiotasolla voidaan käsitellä tai vertailla assosiaation helpottamiseksi [OXL94]. Näin ollen jälkikäsittelyn avulla data-assosiaatiossa ilmentyvien virrehavaintojen määrää voidaan vähentää.

3.3 Liikkeen analysointi

Moniobjektiseurannassa käytettyjen tilastollisten menetelmien lisäksi kuvasarjassa tapahtuvaa liikettä voidaan analysoida suoraan spatiaaliseen informaatioon perustuen. Yksinkertaisimmillaan tämänlainen spatiaalinen analysointi on objektin etsintää kuvatasoista. Yleisemmin kyse on *liikkeen analysoinnista (motion analysis)*. Liikkeen analysoinnin lähestymistavat voidaan jakaa kahteen ryhmään: *optiseen vuohon (optical flow)* perustuviin menetelmiin ja niihin jotka perustuvat *ominaisuuksien seurantaan (feature tracking)*. Optiseen vuohon perustuvissa menetelmissä päätellään liikeinformaatio selvittämällä pikselitasolla intensiteetin vaihtelut koko kuvatasossa [HOR86]. Ominaisuuksien seurantaan perustuvissa menetelmissä taas tarkastellaan kuvatasossa olevia muotoja, kuten *suoria viivoja (line segments)*, *käyriä (curves)*, *kulmia (corners)*, *lohkoja (blocks)*, *äärioviivoja (contours)* tai *alueita (regions)*.

Apertuuriongelma

Monissa liikkeen analysoinnin menetelmissä liikettä tarkastellaan paikallisesti. Esimerkiksi lohkoihin perustuvassa menetelmässä vain paikallisen kuva-alueen liikettä analysoidaan. Lohkolla tarkoitetaan suorakaiteen muotoista kuva-aluetta, jonka siirtymää kuvatasojen välillä tarkastellaan. Jos liike kuvassa on laajempaa kuin tarkasteltava lohko, niin laskennassa saattaa esiintyä niin kutsuttu *apertuuriongelma (aperture problem)*. Tällä tarkoitetaan sitä että suoran tason suuntaista liikettä ei havaita paikallisesti tarkasteltuna.



Kuva 3.5 Apertuuri-ongelma esiintyy kun liike on laajempaa kuin tarkasteltava alue

Apertuuriongelma esiintyy erityisesti kun liikettä analysoidaan aliobjektien avulla. Esimerkiksi kuvantiivistyksessä, jossa koko kuvataso jaetaan lohkoihin ja joiden siirtymiä tarkastellaan, tämä ongelma aiheuttaa virheitä lohkojen siirtymien laskennassa [SEF94]. Esimerkki apertuuriongelmosta nähdään kuvassa 3.5.

Ominaisuuksien seurannan menetelmiä

Ominaisuuksien seurantaan voidaan käyttää useita eri tekniikoita. Eräs niistä on suorien viivojen seuranta (*line segment tracking*), jossa viivojen kuvatasojen välillä tapahtuvan siirtymän perusteella voidaan päätellä kappaleen liike. Suorien viivojen seurannassa vaikeuksia tuottaa kuitenkin apertuuriongelma. Apertuuriongelman ratkaisuna on esitetty käyrien seuranta (*curve tracking*), jolloin liikkeen suuntaisia suoria tasoja ei esiinny ja liike voidaan havaita myös paikallisesti. Toinen ratkaisu apertuuriongelmaan on seurata kuvassa esiintyvien rakenteiden muodostamia kulmia, joita seurataan viitteessä [LAM93].

Eräs käytetty tekniikka on *lohkotäsmäys* (*block matching*), jonka avulla voidaan seurata kiinteiden lohkojen liikettä. Liikeinformaatio voidaan päätellä etsimällä edellisessä kuvassa määritelty lohko tarkasteltavasta kuvasta. Tällöin lohkon siirtymän perusteella voidaan päätellä lohkon nopeus ja suunta.

Lohkojen seurannan asemesta voidaan seurata objektin ääri viivoja. *Ääri viivojen seuranta (contour tracking)* mahdollistaa muuttuvamuotoisen ja ääri viivoiltaan monimutkaisen objektin seurannan. Lisäksi etuna on se, että objektin ääri viivat tiedetään seurannan joka vaiheessa. Tämä on usein tarpeellista sovellusten kannalta. Esimerkiksi kun solujen liikkeitä seurattaessa halutaan solu eristää ääri viivojen tarkkuudella jokaisesta kuvasarjan kuvasta.

Alueen seuranta (region tracking) on lähestymistapa, jossa yhdistyy lohko täsmäyksen ja ääri viivojen seurannan ominaisuuksia. Alueen seurannassa on tarkoitus seurata kuvasarjassa kokonaista aluetta pikselin tarkkuudella. Tähän käytetään kuvankäsittely tekniikkaa jota kutsutaan *alueen laajennukseksi (region growing)*. Alueen laajennuksessa objektin määrittävää aluetta laajennetaan kunnes ennalta määritellyt lopetusehdot täyttyvät, esimerkiksi silloin kun havaitaan reuna tai tarkasteltavana on taustainformaatiota.

4. OBJEKTINETSINTÄ LOHKOTÄSMÄYKSELLÄ

Tässä luvussa tarkastellaan lohkotäsmäys menetelmää, jolla objekti voidaan etsiä yksittäisestä kuvatasosta. Tarkoituksena on selvittää mitä erilaisia tekniikoita voidaan käyttää lohkotäsmäyksessä ja miten täsmäystä voidaan nopeuttaa. Lohkotäsmäys perustuu tekniikkaan, jossa edellisessä kuvatasossa määritellyn objektia mallittavan kuva-alueen eli lohkon sijainti pyritään määrittelemään vertailemalla sitä tarkasteltavaan kuvatasoon. Tuloksena saadaan objektin siirtymä kahden kuvatason välillä tai tarkemmin liikevektori, joka ilmaisee objektin suunnan ja nopeuden.

4.1 Objektin määritelmä

Lohkotäsmäyksessä objekti määritellään yleensä suorakaiteen muotoisena lohkona, joka sisältää seurattavan objektin. Tätä lohkoa kutsutaan *objekttilohkoksi* (*object template*). Objekttilohko voi olla staattinen tai adaptiivinen seurattavan objektin liikkeen mukaan. Vaikka objekttilohko yleensä määritellään suorakaiteen muotoiseksi, sen muotoa ei sinänsä rajoiteta. Se voi esimerkiksi mukautua objektin muodon mukaan.

Staattinen objekttilohko

Staattisella objekttilohkolla tarkoitetaan objekttilohkoa, joka pysyy muuttumattomana koko seurannan ajan. Oletuksena on se, että seurattavan objektin koko ja muoto eivät muutu. Tällöin objektin liike on puhdasta translaatiota. Objekttilohko määritellään matriisina, jonka kertoimina on objektin kuvapisteidien eli pikseleiden intensiteettiarvot.

Objekttilohko sisältää objektin lisäksi myös taustainformaatiota, jos seurattava objekti ei ole suorakaiteen muotoinen. Taustainformaation mukana olo heikentää täsmäyksen tarkkuutta, etenkin lohkon ollessa suuri. Taustainformaation eliminointi objekttilohkosta voidaan aikaansaada määrittelemällä erillinen objekttilohkon kokoinen matriisi, jonka kertoimet määrittelevät objekttilohkon pikseleiden kuulumisen objektiin tai taustaan. Tällöin vain

objektiin kuuluvat pikselit otetaan mukaan laskutoimituksiin. Tämä mahdollistaa objektin täsmäyksen pikselin tarkkuudella. Objektiin kuuluvien pikseleiden valinta saattaa kuitenkin olla ongelmallista. Etenkin jos objekti ei erotu selkeästi taustasta.

Adaptiivinen objektilohko

Objektin liikkeen ollessa mutaatiota staattisen objektilohkon käyttö ei ole mielekäästä, koska objektin muoto muuttuu mutaation vuoksi. Ratkaisuna ongelmaan on *adaptiivinen objektilohko* (*adaptive object template*) [OXL94]. Joka kerta kun objektilohko on täsmätty johonkin kuvatason lohkoon, tämä lohko eristetään kuvatasosta ja siitä muodostetaan uusi objektilohko. Myös objektin koko saattaa muuttua. Tällöin määritellään objektin *minimaalinen kehys* (*MBR, minimum bounding rectangle*) [MAT94]. Minimaalinen kehys määrittelee lohkon, joka on pienin mahdollinen lohko, jonka sisälle objekti mahtuu kokonaisuudessaan.

Kuten staattista objektilohkoa käytettäessä, taustainformaatio objektilohkossa häiritsee täsmäystä. Adaptiivisen objektilohkon tapauksessa ilmenee myös toinen ongelma, joka on *virhesiirtymä* (*walk-off*) [OXL94]. Virhesiirtymä aiheutuu tilanteesta, jossa objektilohko ei täsmää täydellisesti vertailtavaan kuvatasoon. Koska täsmätty lohko kopioidaan objektilohkoksi, saattaa täsmäyksessä tapahtunut virhe kumuloitua siten, että objektilohko ei enää vastaa minimaalista kehystä.

Virhesiirtymän vaikutusta voidaan vähentää eliminoimalla mahdollisimman paljon taustainformaatiota objektilohkosta. Tämä onnistuu rajaamalla objektilohko jokaisen päivityksen jälkeen minimaaliseksi kehykseksi. Oletuksena on että objektilohko erottuu selvästi taustastaan. Tällöin objektilohko voidaan kynnystää vertaamalla objektia koko kuvan histogrammiin. Taustaan kuuluu yleensä suurin osa pikseleistä, joten objektilohkon sisältämä taustainformaatio voidaan rajata pois valitsemalla vain taustainformaatiosta poikkeavat pikseliarvot [OXL94].

4.2 Täsmäys kustannusfunktion minimointina

Myös liikkuvan kuvan tiivistyksessä käytetään liikkeen estimointia. Esimerkiksi MPEG standardissa kuva jaetaan 16 x 16 lohkoiksi, joita kutsutaan *makrolohkoiksi* (*macro block*) [GAL92]. Jokaisen makrolohkon siirtymä lasketaan kustannusfunktiota minimoimalla. Samaa menetelmää voidaan hyödyntää myös objektinseurannassa, jossa yksittäinen lohko voidaan käsittää objektilohkoksi. Tällöin kustannusfunktion minimointi voidaan ymmärtää lohkon täsmäyksenä. Mitä paremmin etsittävä lohko täsmää etsintäalueen lohkoon, sitä pienemmän arvon kustannusfunktio tuottaa. Tällöin täsmäys suoritetaan etsimällä kustannusfunktion minimi.

Pikselitason virhefunktiot

Objektin täsmäyksen laskemisessa voidaan käyttää *absoluuttista keskivirhettä* (*MAE, mean absolute error*). Absoluuttinen keskivirhe voidaan laskea seuraavalla kaavalla [FRÄ95]:

$$\text{MAE} = 1/N \sum_{i=1}^N |L_i - C_i| \quad (4.1)$$

jossa N on lohkon pikseleiden lukumäärä, L on objektilohko ja C on etsintäalueen lohko johon objektilohkoa vertaillaan. Absoluuttinen keskivirhe ilmaisee kahden lohkon välisen eroavaisuuden. Esimerkiksi jos kiinteän objektin liike on puhdasta translaatiota, niin objekti vain siirtyy paikasta toiseen pysyen muuttumattomana. Tällöin lohkot ovat identtiset ja absoluuttinen keskivirhe on 0. Useimmiten objekti kuitenkin muuttuu, esimerkiksi valaistusolosuhteiden vaikutuksesta. Keskivirhe ottaa huomioon pikseleiden välisen eron suuruuden, joten vähäinen valaistuksen muutos ei yleensä aiheuta yhtä suurta virhettä kuin esimerkiksi objektin muodon muuttuminen.

Absoluuttisen keskivirheen lisäksi voidaan hyödyntää myös *neliöllistä keskivirhettä* (*MSE, mean square error*), joka määritellään kaavalla [FRÄ95]:

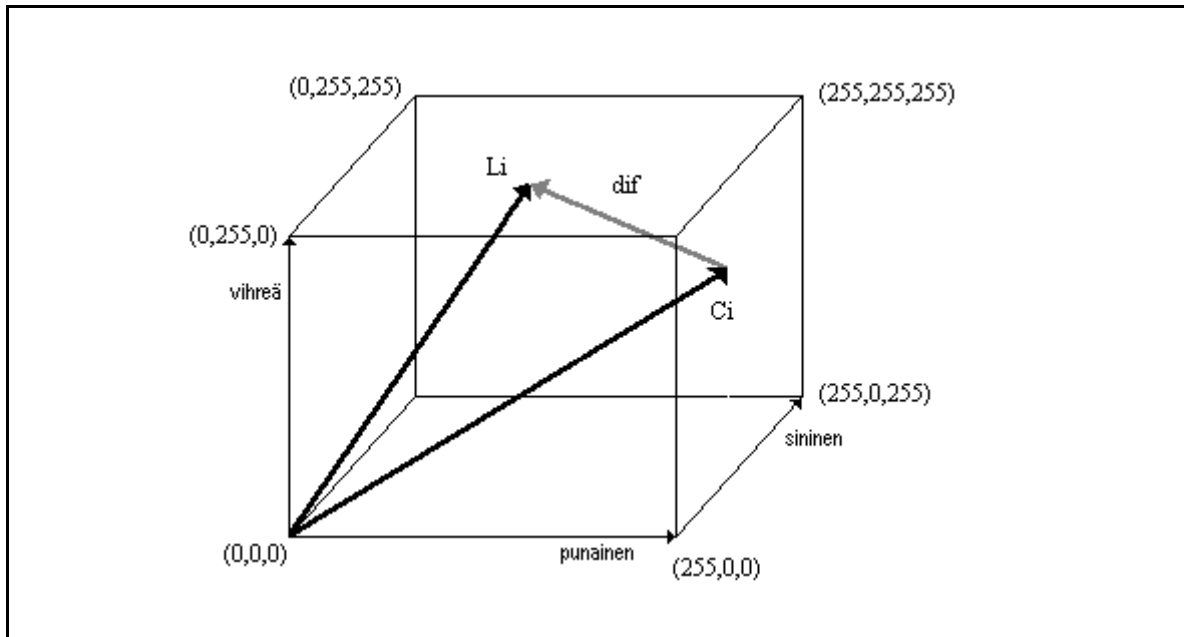
$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (L_i - C_i)^2 \quad (4.2)$$

Neliöllinen keskivirhe eroaa absoluuttisesta keskivirheestä siten, että vertailtavien lohkojen eroavaisuudet lasketaan neliöllisinä. Tämä mahdollistaa absoluuttista keskivirhettä tarkemman täsmäyksen, sillä pikseleiden suuret eroavaisuudet korostuvat. Esimerkkinä voidaan antaa tilanne, jossa 3 x 3 kokoisessa lohossa A jokaisen pikselin eron suuruus on 1, ja vastaavan kokoisessa lohossa B eron suuruus on 2, mutta vain kolme pikseliä eroaa. Tällöin absoluuttinen keskivirhe tuottaa täsmäyksen lohkon B kanssa, kun taas neliöllinen keskivirhe täsmää lohkon A kanssa. Neliöllinen keskivirhe tuottaa näin paremman täsmäyksen, sillä objekti säilyttää muotonsa paremmin pikselierojen ollessa vähäiset.

Täsmäys väriavaruudessa

Objekttilohkon määrittämisessä käytetty pikseleiden intensiteetti voidaan ymmärtää harmaan sävyinä, joten kyseessä on harmaasävykuva. Kuitenkin jos tiedetään, että seurantaan käytetty kuvainformaatio on värillistä, voidaan värejä hyödyntää myös objektin täsmäyksessä. Tällöin objekttilohkon yksittäisen pikselin arvo muodostuu värikomponenteista. Värikomponentit määritellään usein fyysisten perusvärien mukaan punaisen, vihreän ja sinisen värin komponenttien muodostamaksi kolmiulotteiseksi vektoriksi. Väri-informaation avulla objekttilohkon objekti voidaan määrittellä tarkemmin kuin käyttämällä pelkkää intensiteettiarvoa, sillä kaikkien kolmen värikomponentin eri variaatiot muodostavat suuren määrän erilaisia värisävyjä. Tähän perustuu myös täsmäyksen tarkkuuden parantuminen verrattuna intensiteettiesitykseen.

Värisävyjen muodostamaa kokonaisuutta kutsutaan *väriavaruudeksi* (*color space*). Kolmen värikomponentin tapauksessa väriavaruus on kuution muotoinen. Väriavaruuden värin ilmaisee vektori P, jonka origo on väriavaruuden reuna, jossa kaikki värikomponentit ovat nolli. Kustannusfunktion minimointi tapahtuu laskemalla kahden vektorin erotus. Objekttilohkon pikseli täsmää hyvin kun vertailtavien pisteiden etäisyys toisistaan väriavaruudessa on vähäinen. Kuvassa 4.1 selvitetään väriavaruudessa täsmäystä.



Kuva 4.1 Täsmäys kolmen komponentin väriavaruudessa

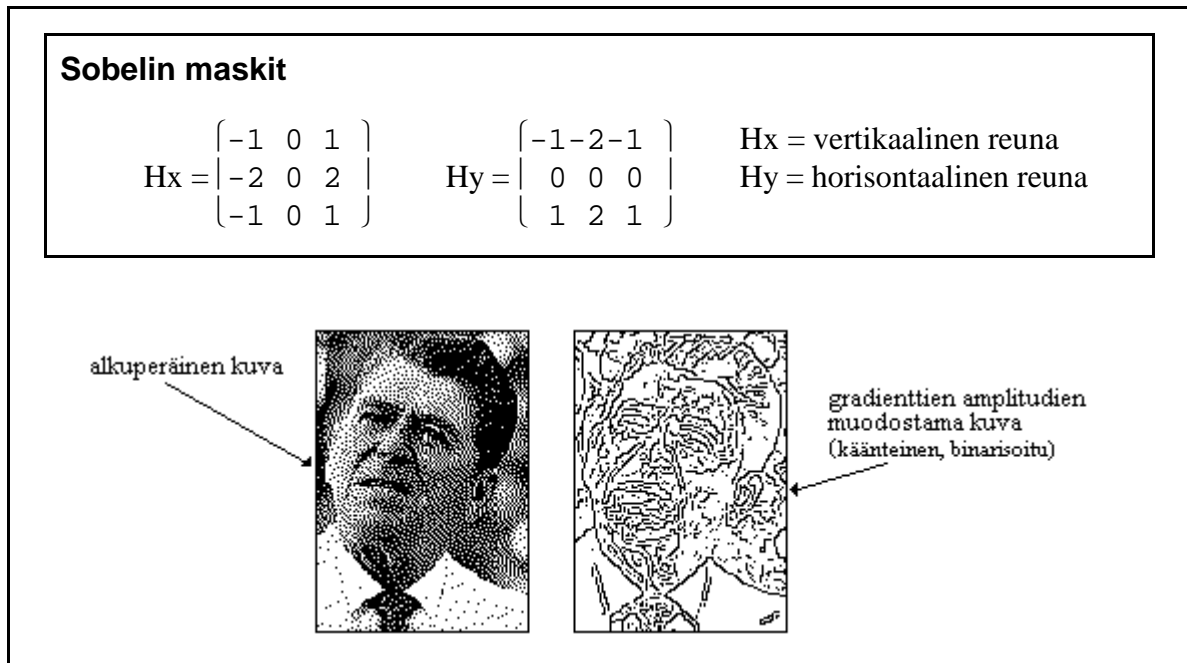
Nyt tiedetään, että mitä lyheempi vertailtavien värivektoreiden erotuksena saatu vektori on, sitä paremmin pikselit täsmäyvät. Tämän perusteella voidaan määrittellä neliöllinen keskivirhe väriavaruudessa tapahtuvaa täsmäystä varten seuraavalla kaavalla:

$$\text{MSE}_{\text{väri}} = \frac{1}{N} \sum_{i=1}^N (|L_i - C_i|)^2 \quad (4.3)$$

jossa L_i on objektilohkon pikselin värivektori ja C_i vastaavan pikselin värivektori vertailtavassa lohkossa.

Reunaan perustuva täsmäys

Täsmäys voi perustua myös kuvassa oleviin reunoihin. Reunalla tarkoitetaan kuvan aluetta, jossa intensiteetti muuttuu voimakkaasti. Tällöin on mahdollista käyttää täsmäyksessä hyväksi pikseleiden sijaintia suhteessa toisiinsa eli seurattavan objektin muotoa. Koko kuvataso ja objektilohko *suodatetaan (filter)* reunojen tunnistukseen soveltuvilla *maskeilla (mask)*. Maskilla tarkoitetaan esimerkiksi 3 x 3 kokoista matriisia ja suodatuksella sitä, että maski kerrotaan jokaisen kuvatason maskin kokoisen alueen kanssa. Maskia ikään kuin



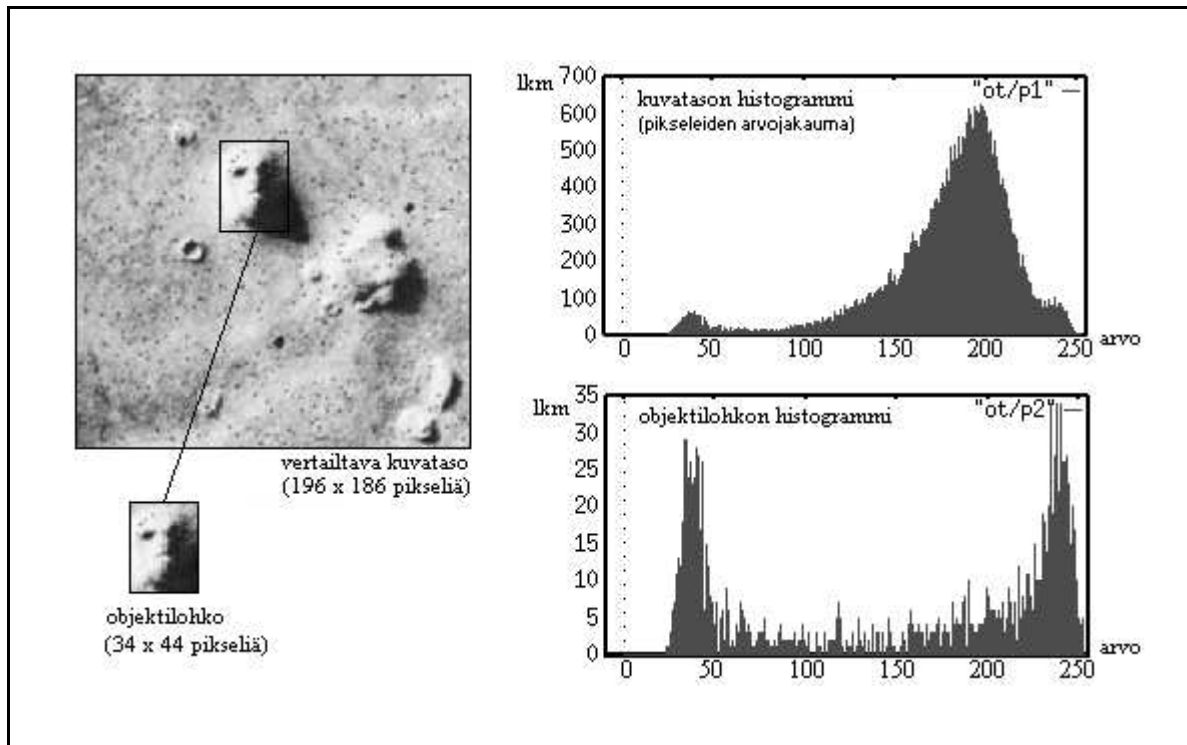
Kuva 4.2 Sobelin maskit ja niiden avulla muodostettu gradientti-kuva

"lakaistaan" kuvataason yli. Tuloksena saadaan amplitudikenttä, jonka jokainen alkio ilmaisee vastaavan pisteen kohdalla olevan reunan suuruutta. Tätä reunan suuruutta ilmaisee *gradientti* (*gradient*), joka on reunan suuntaan osoittava vektori. Täsmäys perustuu näin ollen gradientin amplitudien erotukseen. Mitä enemmän objektilohkon ja kuva-alueen reunat ovat samansuuruisia, sitä parempi täsmäys. Erotuksen laskenta voi perustua esimerkiksi absoluuttiseen tai neliölliseen keskivirheeseen.

Reunan suodatukseen soveltuvia maskeja ovat esimerkiksi Sobelin maskit. Sobelin maskeja on kaksi, joista toinen tunnistaa horisontaalisessa ja toinen vertikaalisessa suunnassa olevia reunoja. Muita reunoja tunnistavia maskeja on esimerkiksi Prewittin ja Kirschin maskit. Lisäksi reunainformaatiota on mahdollista analysoida Laplace-suotimella, jossa on vain yksi maski. Reunaan perustuva täsmäys on suoraa jatkoa aiemmassa tutkielmassani esittelemilleni kuvankäsittelymenetelmille, joista enemmän viitteessä [HAV95]. Kuvassa 4.2 nähdään Sobelin maskit ja esimerkki suodatetusta kuvasta.

Täsmäys histogrammin avulla

Objektilohkon täsmäykseen voidaan käyttää lohkon pikseleiden *arvojakaumaa* (*value*



Kuva 4.3 Objekttilohko, kuvataso ja niiden histogrammit

distribution). Arvojakauman laskemiseksi muodostetaan lohkon *histogrammi* (*histogram*), joka ilmaisee kuinka monta pikseliä kutakin intensiteettiarvoa tai väriarvoa lohkossa esiintyy. Kuvassa 4.3 nähdään kuvatason ja objekttilohkon histogrammit.

Histogrammi muodostetaan objektilohkolle ja jokaiselle etsintäalueen vertailtavana olevalle lohkolle. Jälleen voidaan laskea keskivirhe, joka ilmaisee histogrammien eroavaisuuden. Histogrammien neliöllinen keskivirhe intensiteettikuville voidaan määrittellä kaavalla:

$$\text{MSE}_{\text{hist}} = 1/V_{\text{max}} \sum_{i=1}^{V_{\text{max}}} (H1_i - H2_i)^2 \quad (4.4)$$

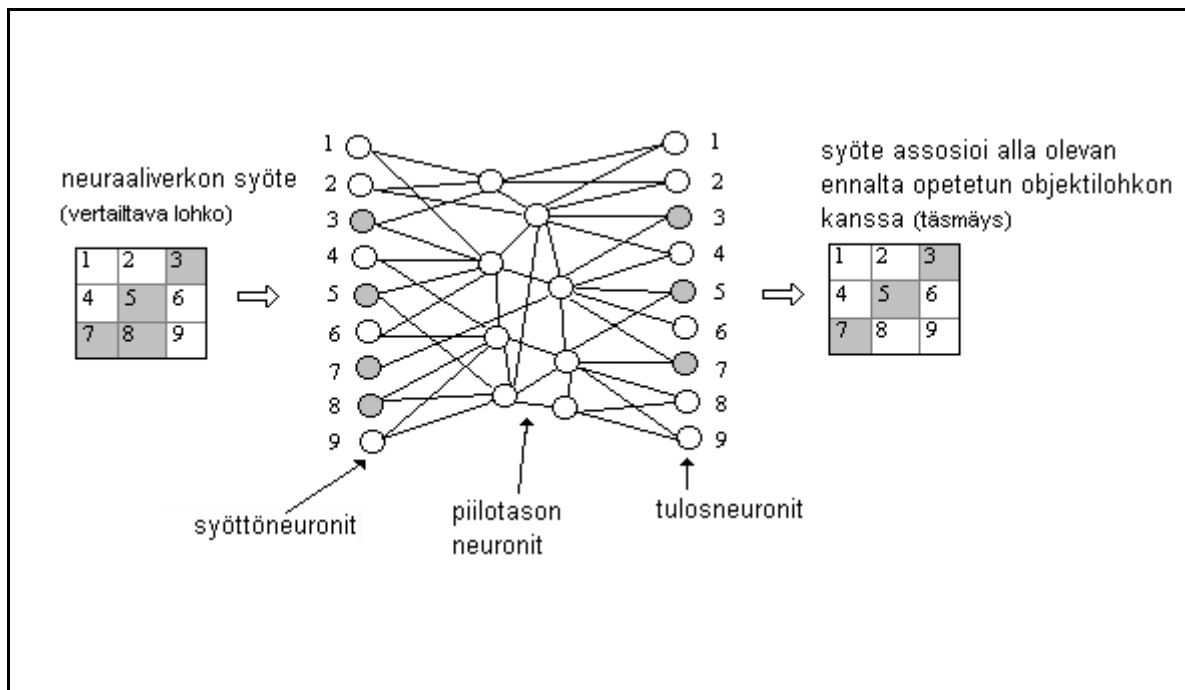
jossa V_{max} on pikseleiden mahdollisten arvojen lukumäärä, $H1$ on objektilohkon histogrammi ja $H2$ etsintäalueen lohkon histogrammi. Värikuvan tapauksessa histogrammi muodostetaan erikseen jokaiselle värikomponentille, joita jokaista vertaillaan kaavalla 4.4.

Histogrammin käyttö täsmäyksessä soveltuu hyvin tilanteisiin, joissa objektin muoto on vaikeasti määriteltävissä. Toisin sanoen objektin ääriviivat eivät ole selkeät, esimerkiksi

häiriön vaikutuksesta. Objektin täytyy kuitenkin erottua taustasta. Lisäksi äkilliset valaistusolosuhteiden muutokset tai objektin muodon muuttuminen vaikeuttavat täsmäystä histogrammin avulla. Tämä on kuitenkin varsin yleinen ongelma objektin täsmäyksessä. Histogrammin käyttö yhdessä adaptiivisen objektilohkon kanssa vähentää valaistuksen ja muodon muuttumisen ongelmaa. Tällöin jokaisen objektilohkon päivityksen yhteydessä päivitetään myös objektilohkon histogrammi.

Täsmäys neuraaliverkolla

Jos seurattava objekti tunnetaan tarkasti, voidaan täsmäyksessä hyödyntää neuraaliverkkoa. Neuraaliverkko koostuu useista toisiinsa kytketyistä laskentayksiköistä, neuroneista. Yksittäisellä neuronilla on useita syöttökanavia, aktivointifunktio ja yksi tuloskanava. Aktivointifunktio päättelee syöttötietojen perusteella, lähettääkö se syöttötietonsa tuloskanavaan vai ei. Tuloskanava johtaa johonkin toiseen neuroniin, joka tekee vastaavan laskentaprosessin. Neuraaliverkolla on syöttöneuronit, jotka lukevat tietoa ympäristöstä, piilotason neuronit, jotka suorittavat laskennan ja tulosneuronit, jotka ilmaisevat neuraaliverkon laskennan tuloksen. Laskenta etenee piilotason neuroneissa kunnes



Kuva 4.4 Esimerkki täsmäyksestä neuraaliverkolla

syöttöneuronien tulos ylittää kynnyksen joka on asetettu päätöksen muodostuksen vaatimukseksi. Täsmäyksen onnistuminen luetaan tulosneuronien arvoista. Mitä suurempi arvo, sen parempi täsmäys.

Täsmäyksessä käytettävä neuraaliverkko on niin kutsuttu assosiativinen muisti, johon voidaan tallettaa rajallinen määrä *malleja* (*pattern*). Kun neuraaliverkko saa syötteenä jonkin mallin, se ilmaisee kuinka hyvin syöte vastaa talletettuja malleja. Vastaava assosiativinen muisti voidaan toteuttaa perinteisillä menetelmillä laskemalla *Hamming - etäisyys* (*Hamming distance*), jolla tarkoitetaan sitä, minkä verran muutoksia täytyy tehdä malliin A, jotta siitä voidaan muodostaa malli B [HER92]. Tämä on sikäli mielenkiintoista, että Hamming-etäisyyden laskenta voidaan redusoida virhefunktion minimoinniksi. Tällöin voidaan todeta, että virhefunktion täsmäys voidaan toteuttaa rinnakkaislaskennalla neuraaliverkkoja hyödyntämällä.

Neuraaliverkko ei tunnista mitään ellei sitä ensin opeteta. Neuraaliverkon opettamisella tarkoitetaan esimerkkien tallettamista neuraaliverkon muistiin. Objektinseurannan kannalta tämä tarkoittaa sitä, että kun objekti ensin opetetaan neuraaliverkolle, voidaan tulosneuroneiden arvoista päätellä kuinka tarkasti objektilohkon täsmäys onnistuu opetetun objektin kanssa. Yhdelle neuraaliverkolle voi opettaa useita objekteja, mutta se kuinka monta objektia neuraaliverkko kykenee oppimaan, riippuu neuronien lukumäärästä. Neuraaliverkko ei kykene tunnistamaan muuta kuin sille opetetut objektit, joten se ilmaisee aina kuinka hyvin tarkasteltava kohde täsmää opittuihin objekteihin. Kuvassa 4.4 visualisoidaan täsmäystä neuraaliverkon avulla.

Kolmiulotteisen ympäristön tapauksessa todellisesta objektista voidaan tallettaa kaksiulotteinen projektio eri suunnista tarkasteltuna. Tällöin neuraaliverkko voi tunnistaa myös rakenteeltaan monimutkaisen objektin vaikka sen kaksiulotteinen projektio muuttuu objektin kääntyessä.

Neuraaliverkon etuna on sen kyky tunnistaa oppimansa objektit voimakkaasti häiriöllisessäkin ympäristössä. Haittoina voidaan mainita tunnistusalgoritmin suuri aikakompleksisuus, sekä lisäksi se, että objekti on tunnettava ennalta ja opetettava neuraaliverkolle.

Täsmäys painomatriisin avulla

Objekttilohko määriteltiin suorakaiteen muotoiseksi, vaikka seurattava objekti olisikin muodoltaan jotain muuta kuin suorakaide. Tästä aiheutuu taustainformaation haitallinen vaikutus objektin täsmäyksen tarkkuuteen. Eräs ratkaisu vähentää taustainformaation vaikutusta on mukauttaa objekttilohko seuraamaan mahdollisimman hyvin objektin ääri viivoja. Toteutuksen kannalta yksinkertaisempi ratkaisu on kuitenkin käyttää *painomatriisia* (*weight matrice*). Painomatriisin perusajatuksena on oletus, että objekti löytyy objekttilohkon keskeltä. Tällöin objekttilohkon keskellä olevia pikseleitä painotetaan enemmän kuin reunoilla olevia. Täsmäys painomatriisin avulla voidaan toteuttaa laskemalla *painotettu neliöllinen keskivirhe*, joka määritellään seuraavalla kaavalla:

$$\text{WMSE} = \frac{1}{N} \sum_{i=1}^N w_i (L_i - C_i)^2 \quad (4.5)$$

jossa painoarvo w_i saadaan painomatriisista. Painomatriisi voidaan määritellä siten, että painoarvo w_i vähenee euklidisen etäisyyden mukaan lohkon keskipisteestä lähtien. Keskikohdan painottaminen ei kuitenkaan aina johda parhaimpaan tulokseen. Objekttilohkon keskellä saattaa olla "reikiä", jotka sisältävät taustainformaatiota. Painomatriisin ei tarvitsekaan välttämättä painottaa lohkon keskipistettä, vaan se voidaan määritellä pikselin tarkkuudella sovelluksesta riippuen. Keskikohdan painottaminen on kuitenkin oletuksena kelvollinen, ainakin adaptiivisen objekttilohkon tapauksessa, jossa objektin muoto muuttuu seurannan edetessä.

4.3 Etsintäalueen rajaaminen

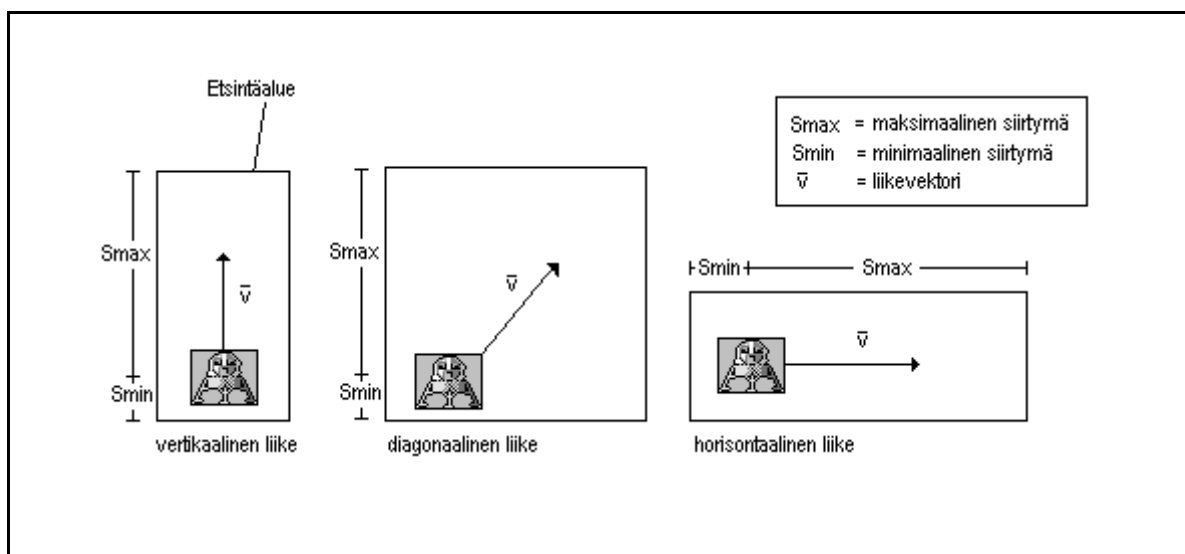
Objekttilohkoa voidaan etsiä koko kuvan alueelta. Koko kuvan läpikäyminen objekttilohkoa vertaamalla on kuitenkin runsaasti laskenta-aikaa vievä toimenpide. Laskennan tehostamiseksi etsintäaluetta pyritäänkin rajaamaan.

Objektin suurinta mahdollista siirtymää kahden perättäisen kuvan välillä kutsutaan *maksimaaliseksi siirtymäksi (maximal motion displacement)* [SEF94], joka ilmaistaan pikseleinä kuvaa kohti (pixels per frame). Vastaavasti voidaan määritellä minimaalinen siirtymä. Jos maksimaalinen siirtymä on vain muutamia pikseleitä, niin voidaan tehdä oletus, että seurattava objekti löytyy edellisessä kuvassa olevan sijaintinsa läheisyydestä. Näin etsintäalueen koko voidaan määritellä minimaalisen ja maksimaalisen siirtymän perusteella.

Liikeinformaation hyödyntäminen

Optimaalinen täsmäys etsintäalueella saadaan tekemällä *täysihaku (full search)*, jossa objektilohkoa verrataan kaikkiin mahdollisiin etsintäalueen lohkoihin. Eräs keino laskennan nopeuttamiseksi on vertailla objektilohkoa vain pieneen osaan etsintäaluetta olettaen, että kustannusfunktion palauttama arvo vähenee monotonisesti kohti parasta täsmäystä [SEF94]. Algoritmi saattaa olla nopea suurellakin etsintäalueella, mutta se ei ole optimaalinen, sillä se voi joutua paikalliseen minimiin. Tällöin algoritmi tuottaa virheellisen täsmäyksen.

Paikallisen minimi ongelma voidaan välttää pyrkimällä rajaamaan etsintäalue mahdollisimman pieneksi ja suorittamalla sitten täysihaku. Etsintäalueen rajaamisessa voidaan hyödyntää liikeinformaatiota, jota ovat objektin nopeus ja suunta. Tällöin



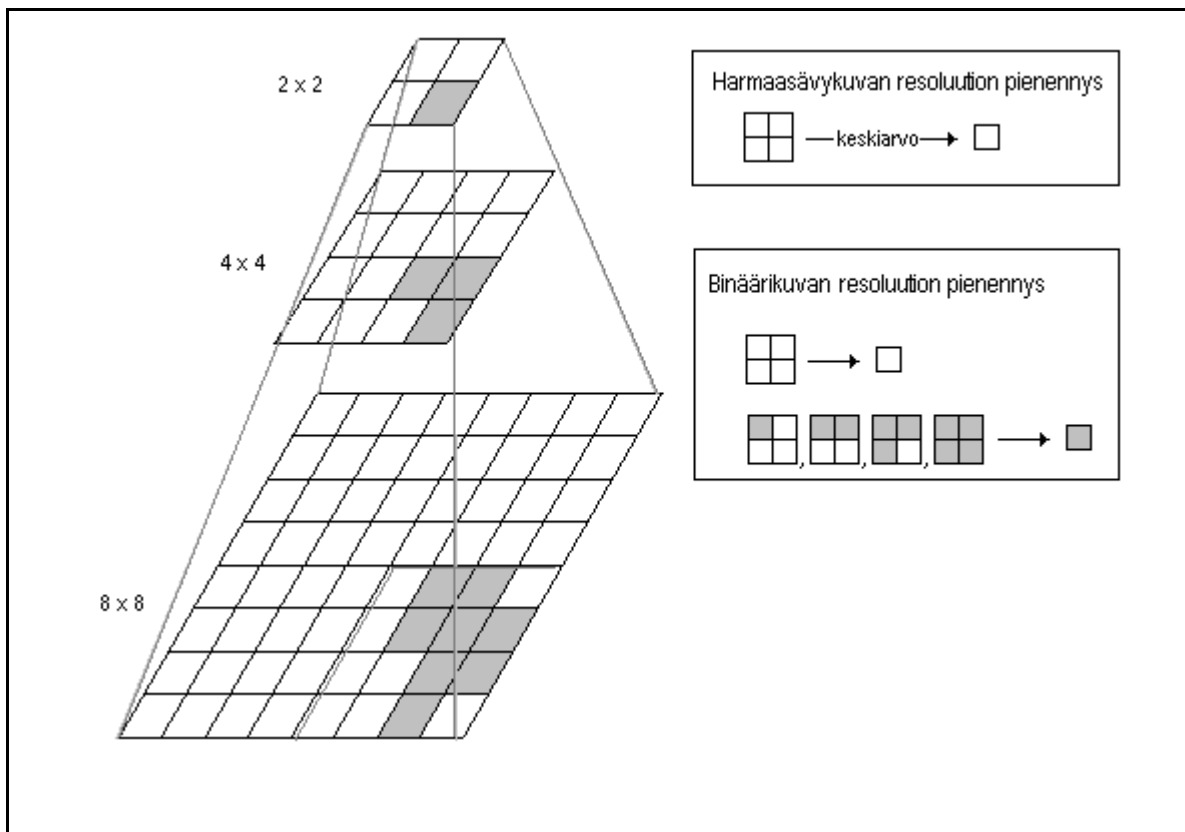
Kuva 4.5 Etsintäalue mukautuu objektin ennustetun liikkeen mukaan

etsintäalueen sijainti ja muoto voidaan määrittellä objektin ennustetun sijainnin perusteella siten, että etsintäalue laajenee objektin liikesuunnan mukaan (kuva 4.5).

Hierarkkinen liikkeen estimointi

Etsintäalueen rajauksen lisäksi laskentaa voidaan tehostaa vertailemalla objektilohkoa vain etsintäalueen "kiinnostaviin" osiin. Kiinnostavilla osilla tarkoitetaan alueita, jotka jonkin ominaisuutensa puolesta ovat soveliaampia täsmäykseen kuin toiset. Tarkoituksena on valita vain ne etsintäalueen osat joissa objektin oletetaan sijaitsevan. Eräs tapa valita edellä mainitut alueet on käyttää *hierarkkista liikkeen estimointia* (*hierarchical motion estimation*). Hierarkkisessa liikkeen estimoinnissa kuvatason resoluutiota pienennetään rekursiivisesti, kunnes eri resoluutiotasot muodostavat pyramidin [SEF94]. Kuvassa 4.6 nähdään resoluutiotasojen muodostama pyramidi.

Tämä pyramidi muodostetaan sekä tarkasteltavalle kuvalle, että referenssikuvalle. Liikevektorien muodostus aloitetaan ylimmältä tasolta, jossa resoluutio on heikoin.



Kuva 4.6 Resoluutiopyramidi hierarkkisessa liikkeen estimoinnissa

Muodostettuja liikevektoreita käytetään ohjeena siirryttäessä yhä tarkemman resoluution tasoihin, kunnes alkuperäisen tarkkuuden tason liikevektorit saadaan laskettua. Edellä kuvattu menetelmä sopii useiden lohkojen käsittelyyn ja siitä on hyötyä etenkin apertuuri-ongelman ratkaisussa, sillä liike voidaan rajata lohkon sisälle pienentämällä resoluutiota.

Yksittäisen objektin kannalta hierarkkista liikkeen estimointia voidaan soveltaa seuraavalla tavalla: aluksi objektilohkolle ja etsintäalueelle muodostetaan resoluutiopyramidit. Vertailu aloitetaan alhaisimman resoluution tasolta, siis resoluutiopyramidin huipulta. Siirryttäessä seuraavalle resoluutiotasolle, tarkastellaan enää vain edellisellä tasolla hyvin täsmänneitä alueita. Tällöin ainoastaan ne alueet, joissa objektin oletetaan sijaitsevan, käsitellään pikselitasolla. Esimerkiksi laajat taustainformaation täyttämät alueet voidaan jättää tarkastelematta yksityiskohtaisesti.

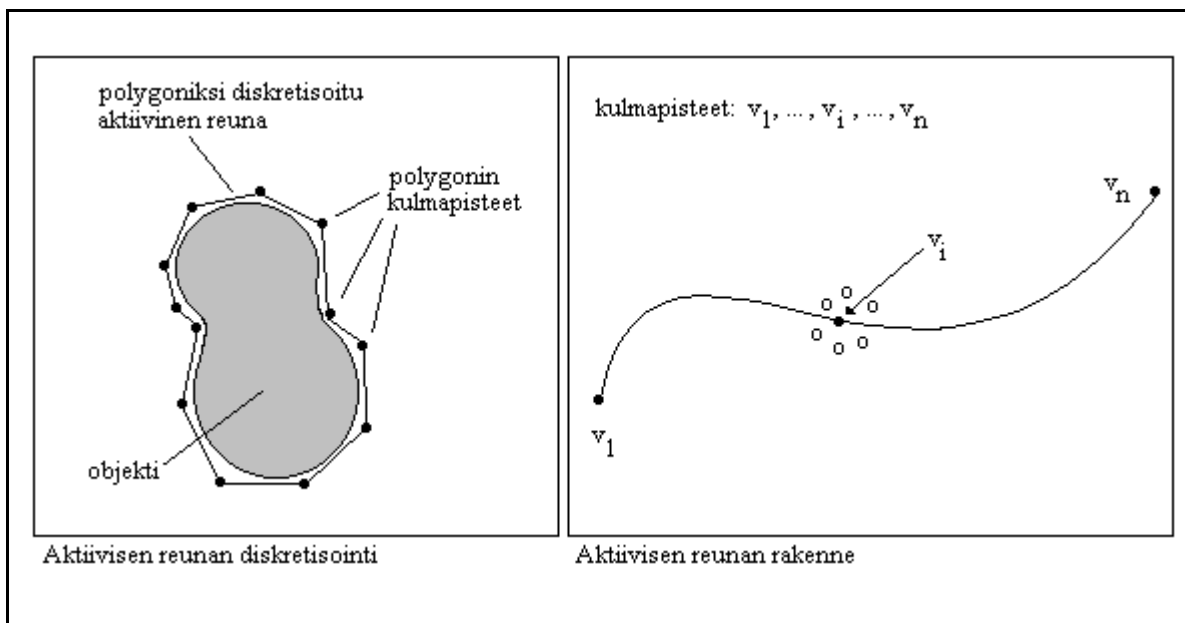
Valittaessa tarkastelun ulkopuolelle jätettäviä alueita on huomattava, että objektin reunat saattavat kadota resoluutiota pienennettäessä. Tällöin poisjätettävät alueet on valittava riittävän kaukaa objektin mahdollisesta sijaintipaikasta. Objektin sijaintipaikka voidaan arvioida vertailemalla etsintäalueen resoluutiotasoon objektilohkon vastaavaa resoluutiotasoa.

5. OBJEKTIN REUNOJEN SEURANTA

Objektin reunojen seurannassa tarkoituksena on *eristää (extract)* objekti taustastaan ääriviivojensa mukaan kuvasarjan jokaisesta kuvasta. Objektin reunojen seurantaan soveltuvia menetelmiä käytetään usein muuttuvamuotoisten objektien seurantaan. Tässä luvussa on tarkoitus tarkastella ja havainnollistaa kahta objektin reunojen seurantaan soveltuvaa lähestymistapaa, jotka ovat aktiivisen reunan menetelmä ja alueenlaajennus-tekniikka. Aktiivisen reunan menetelmässä käytetään energiamallia aliobjektien liikkeen analysoinnissa. Energialla tarkoitetaan liikkeen mallia, jota käytetään aliobjektien liikkeen määrittelyssä.

5.1 Aktiivinen reuna

Aktiivisella reunalla (active contour) tarkoitetaan *s-käyrää (spline)*, joka mukautuu tiettyyn muotoon, kun siihen vaikuttava energia minimoidaan. Aktiiviseen reunaan vaikuttava energia määritellään siten, että energia on minimissään silloin kun aktiivinen reuna kulkee seurattavan objektin reunaa mukaillen. Kirjallisuudessa aktiivista reunaa kutsutaan toisinaan myös *käärmeeksi (snake)*. Aktiivisen reunan käsite on helpompi ymmärtää, kun



Kuva 5.1 Aktiivinen reuna eli "käärme" [OLS93]

aktiivisen reunan määrittävä s-käyrä ajatellaan polygonina, jonka *kulmapisteet* (*vertex*) määrittävät käyrän muodon. Kuvassa 5.1 nähdään esimerkki objektista ja sen ympäröimästä aktiivisesta reunasta eli käärmeestä.

Aktiivisen reunan käsite voidaan karkeasti ymmärtää aliobjektien liikkeen analysointina. Tällöin aliobjekteja ovat kulmapisteet, joiden liikkeiden perusteella voidaan analysoida koko seurattavan objektin liikettä. Esimerkiksi luvussa 2 kuvattu kokonaisen laskeutuvan lentokoneen liikkeen analysointi tarkastelemalla sen eri osien liikettä tapahtuu vastaavalla periaatteella, vaikka siinä käytetty tekniikka ei perustukaan aktiiviseen reunaan.

Aktiivisen reunan energia minimoidaan kuvasarjan jokaisessa kuvassa. Objektinseurannan käynnistyessä kuvasarjan ensimmäisessä kuvassa voidaan seurattava objekti ympäröidä karkeasti, esimerkiksi käyttämällä hiirtä. Tämän jälkeen aktiivisen reunan energia minimoidaan ja reuna mukautuu objektin ääri viivojen mukaisesti. Kuvasarjan seuraavassa kuvassa käytetään alkutilanteena edellisen kuvan minimoitua aktiivista reunaa.

Aktiivisen reunan energia

Aktiivinen reuna mukautuu objektin ääri viivoihin sen mukaan, miten aktiiviseen reunaan vaikuttava energia on määritelty. Aktiivisen reunan hyödyntämisessä oleellinen osa onkin energiamallin määrittelyä. Aktiiviseen reunaan vaikuttava kokonaisenergia E_{total} voidaan määritellä seuraavalla tavalla:

$$E_{\text{total}} = E_{\text{int}} + E_{\text{image}} + E_{\text{ext}}, \quad (5.1)$$

jossa E_{int} on *sisäinen energia* (*internal energy*), E_{image} *kuvaenergia* (*image energy*) ja E_{ext} on *ulkoinen energia* (*external energy*). Jokainen energiakomponentti vaikuttaa omalta osaltaan lopullisen aktiivisen reunan muotoon. Aktiivisen reunan malli ei sinänsä rajaa miten energiakomponentit määritellään, vaan ne voidaan suunnitella sovellusten tarpeiden mukaisesti.

Aktiivisen reunan sisäinen energia toimii äkillisten muutosten rajoittajana. Sen avulla aktiivinen reuna saadaan mukautumaan valitun energiamallin mukaiseen muotoon. Fujimura et al. määrittelevät sisäisen energian neljän komponentin avulla [FUJ93]. Tällöin sisäinen energia voidaan määrittellä seuraavalla kaavalla:

$$E_{\text{int}} = E_{\text{spline}} + E_{\text{area}} + E_{\text{dist}} + E_{\text{cont}}, \quad (5.2)$$

jossa E_{spline} määrittää s-käyrän yleisen toiminnallisuuden, E_{area} aktiivisen reunan muodon karkealla tasolla, E_{dist} määrittelee aktiivisen reunan kulmapisteiden välit ja E_{cont} liikkeen ja muodonmuutoksen jatkuvuuden. Yksityiskohtaiset määritykset löytyvät viitteestä [FUJ93].

Sisäisen energian komponentin E_{area} mallina voi olla niin kutsuttu "ilmapallo"-malli, jossa aktiivinen reuna voi laajeta tai kutistua ilmapallon tavoin tasaisesti joka puolelta ilman terävien kulmien muodostumista. Komponentti E_{area} voidaan valita sen mukaan, minkälainen seurattavan objektin liikkeen ja muodon oletetaan olevan. Jos minkäänlaista oletusta ei voida tehdä, malliksi voidaan valita jokin yksinkertaisen muodon säilyttävä malli, esimerkiksi edellä mainittu "ilmapallo"-malli. Aktiivisen reunan kulmapisteiden välejä kontrolloivan sisäisen energian komponentin E_{dist} avulla voidaan varmistaa se, ettei kahden kulmapisteen välinen etäisyys kasva liian suureksi tai kutistu olemattomiin.

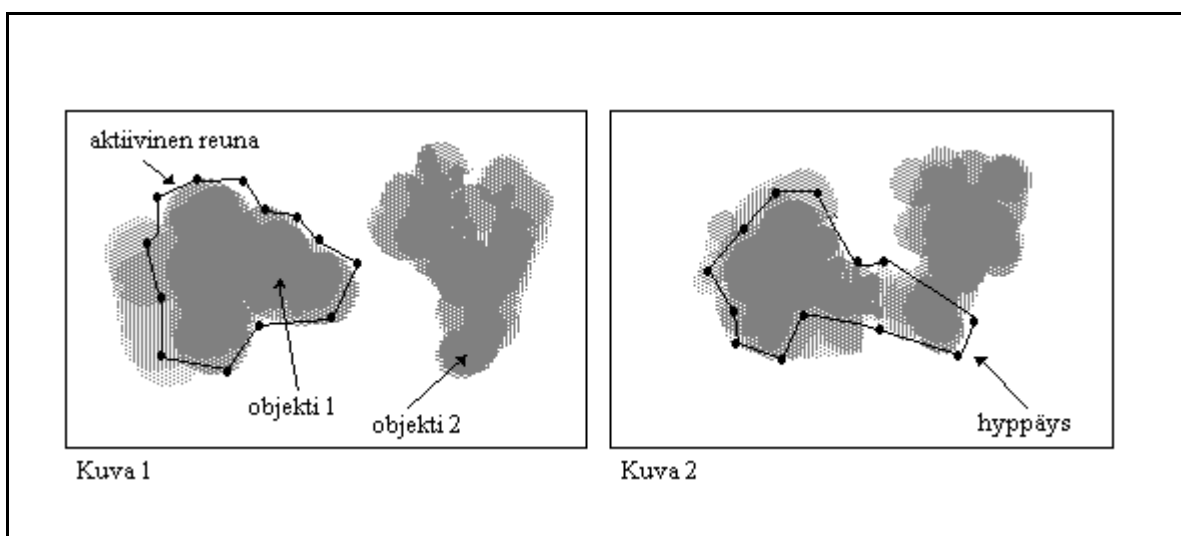
Aktiivisen reunan kuvaenergia E_{image} ohjaa aktiivisen reunan kohti kuvassa esiintyviä ominaisuuksia, kuten alueita, joissa intensiteetti vaihtuu voimakkaasti. Kuvaenergia lasketaan kunkin kulmapisteen läheisyydessä olevan paikallisen kuvainformaation perusteella. Kuvaenergia voidaan jakaa kahteen komponenttiin, jotka ovat E_{edge} ja $E_{\text{intensity}}$. Komponentin E_{edge} avulla aktiivinen reuna saadaan lähestymään kuvassa olevia ominaisuuksia. Komponentin $E_{\text{intensity}}$ avulla aktiivinen reuna hakeutuu seurattavan objektin reunoilla olevaa intensiteettiä vastaavia arvoja kohti. Tällä tarkoitetaan siis sitä, että aktiivisen reunan kohdalla oleva kuvan intensiteetti pysyy samana; olettaen tietysti että objektin intensiteetti ei muutu kuvasarjan eri kuvatasoissa.

Ulkoinen energia E_{ext} voidaan määrittellä siten, että aktiivinen reuna mukautuu tiettyjen kuva-alueiden läheisyyteen. Tätä voidaan hyödyntää silloin kun objektin ympäristö tunnetaan ja tiedetään, miten ympäristö vaikuttaa objektin liikkeeseen.

Hyppäysongelma

Aktiivinen reuna soveltuu hyvin muuttuvamuotoisten objektien seurantaan häiriöllisessäkin ympäristössä. Aktiivisen reunan ongelmana on aliobjekteja hyödyntäville objektinseurantamenetelmille tyypillinen *hyppäysongelma* (*warping*), jossa aliobjektin täsmäys tapahtuu väärän objektin kanssa tai aliobjekti täsmää jonkin ympäristön rakenteen kanssa. Tämänkaltainen väärä täsmäys on tyypillinen ongelma kaikille seurantamenetelmille. Aktiivisen reunan tapauksessa hyppäysongelma ilmenee siten, että jokin kulmapisteistä tekee hyppäyksen jonkin toisen objektin reunojen tasolle. Kuvassa 5.2 kuvataan aktiivisen reunan hyppäysongelma.

Hyppäysongelma aiheutuu lähinnä liian suuresta siirtymästä kahden kuvatason välillä ja siitä, että ympäristön häiriöllisyys tekee objektien reunat epäselviksi. Hyppäysongelmaa voidaan rajoittaa olettamalla kuvatasojen välinen liike tarpeeksi pieneksi, mutta se on usein sovelluskohtaisesti mahdotonta. Tällöin aktiivisen reunan sijainti seuraavassa kuvassa on pyrittävä jotenkin ennustamaan. Ennustus voidaan toteuttaa analysoimalla objektin



Kuva 5.2 Aktiivisen reunan hyppäysongelma

aikaisempaa liikettä ja laskemalla objektin nopeus ja suunta. Mutaation vaikutusten ennustus voi olla hankalaa. Muuttuvamuotoisten objektien liike on kuitenkin usein säännöllistä, joten niiden liikkeitä analysoimalla on mahdollista päätellä jotain objektin tulevasta sijainnista ja muodosta.

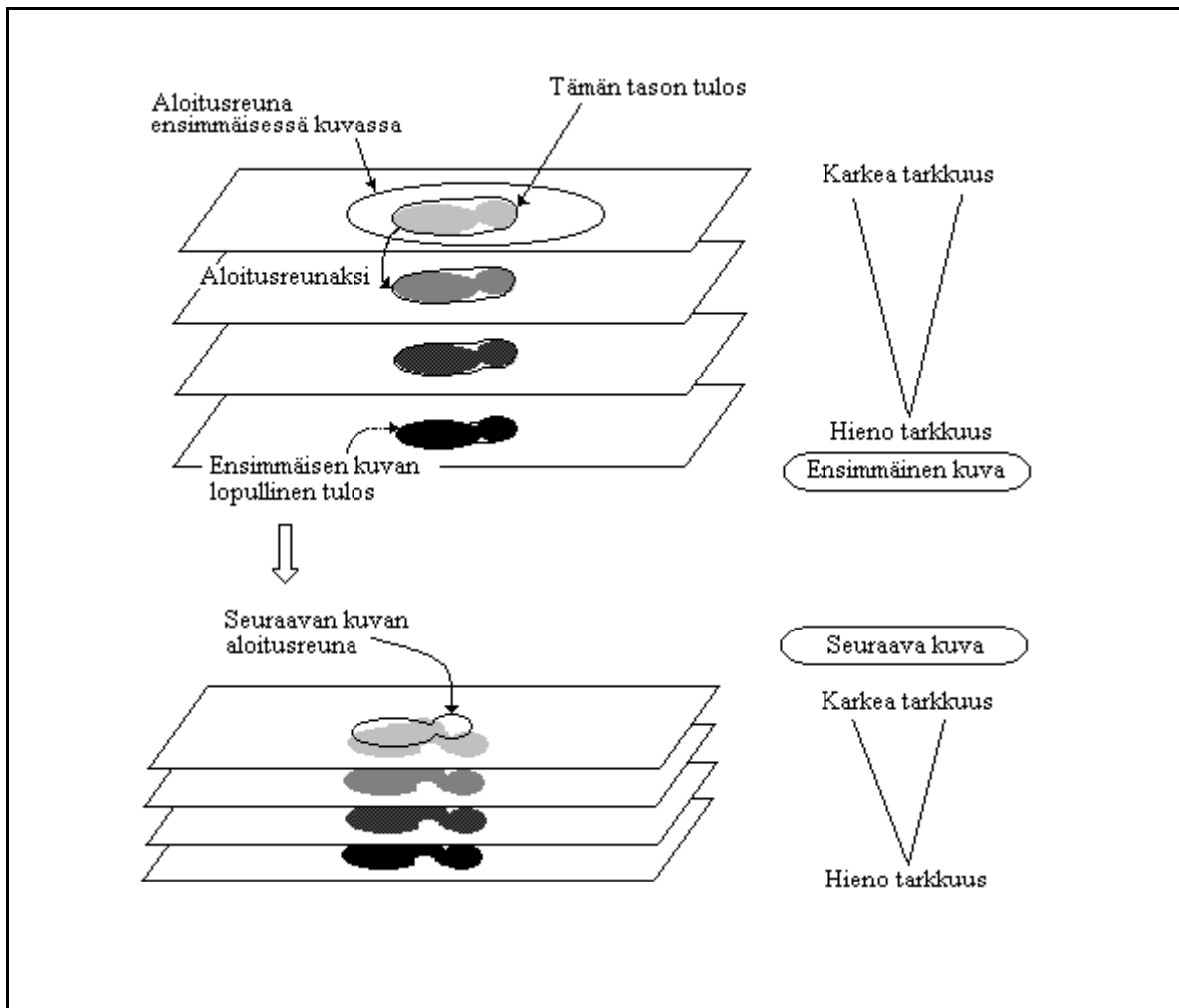
Jos seurattava objekti on suhteellisen kiinteä, voidaan sen sijainti ennustaa määrittelemällä rajoituksia objektin mutaatiolle. Tällä tarkoitetaan sitä, että objektin muoto ei saa muuttua liian paljon kahden kuvatason välillä. Tällöin kyseessä on *kiinteä aktiivinen reuna (rigid active contour)* tai *kiinteä käärme (rigid snake)* [ODI93]. Hyppäysongelman tapauksessa objektin muoto usein muuttuu voimakkaasti. Mutaatorajoitukset eivät kuitenkaan ratkaise kokonaan hyppäysongelmaa, sillä kolmiulotteinen todellinen objekti saattaa kääntyessään paljastaa aiemmin näkymättömiä rakenteitaan, jolloin objekti mutatoi voimakkaasti. [ODI93]

Hierarkkinen energian minimointi

Aktiivisen reunan energian minimointiin käytetään yleensä dynaamista ohjelmointia, sillä sen avulla voidaan saavuttaa optimaalinen reuna suhteessa käytettyyn energiamalliin [OLS93]. Dynaamisen ohjelmoinnin avulla toteutetun energian minimointitekniikan aikakompleksisuus on kuitenkin luokkaa $O(nm^3)$, jossa n on kulmapisteiden lukumäärä ja m on mahdollisten valintojen määrä kulmapisteessä [FUJ93]. Aikakompleksisuutta voi vähentää jonkin verran tinkimällä käytetyn energiamallin toiminnallisuudesta [OLS93].

Energian minimoinnin aikakompleksisuutta voidaan parantaa myös hyödyntämällä heuristisia tekniikoita, joiden tarkoituksena on lähestyä optimia tarkastelematta kaikkea informaatiota. Eräs tällainen tekniikka on *hierarkkinen energian minimointi*, jossa hyödynnetään kuvatason eri tarkkuustasoja (*multiple scales*) [FUJ93]. Hierarkkisella energian minimoinnilla on vahva analogia hierarkkisen liikkeen estimoinnin kanssa, jota käsiteltiin luvussa 4.

Kuvatasojen tarkkuustasot muodostetaan suodattamalla kuvan yksityiskohtia pois eri vahvuisilla suodattimilla. Jokaisella tarkkuustasolla suoritetaan energian minimointi



Kuva 5.3 Aktiivisen reunan hierarkkinen energian minimointi [FUJ93]

aloittaen kaikkein karkeimmasta tarkkuudesta. Jokaisen tason saavutettu energiaminimi toimii alkuarvona seuraavalla, edellistä tarkemmalla tasolla. Näin jatketaan, kunnes saavutetaan lopullinen kuvatasoa vastaava tarkkuus. Tämä taas toimii alkuarvona seuraavalle kuvatasolle. Hierarkkisen energian minimoinnin perusajatus on se, että aktiivinen reuna voi mukautua objektin ääriivuihin ilman yksityiskohtien aiheuttamia äkillisiä muutoksia objektin reunan muodossa. Kuvassa 5.3 nähdään hierarkkisen energian minimoinnin periaate.

5.2 Alueenlaajennus

Alueenlaajennuksella (*region growing*) tarkoitetaan tekniikkaa, jossa aloitusalueesta (*seed region*) tai *aloituspisteestä* (*seed point*) lähtien alueeseen lisätään alueen reunoilla olevien

pikseleiden naapuripikseleitä, kunnes alue määrittää halutun osan kuvasta. Aluetta laajennetaan *laajennusehdon* (*growing rule*) perusteella kunnes *lopetusehto* (*stopping rule*) täyttyy [FRÄ95]. Laajennusehto määrittellään yleensä siten, että alueen pikselit muodostavat jonkin samankaltaisuuden (*homogeneity*) [SCH93]. Yksinkertaisimmillaan alueen samankaltaisuus tarkoittaa alueen pikselien samaa intensiteettiä. Aluetta laajennetaan yleensä lisäämällä vain yksi pikseli kerrallaan.

Laajennusehdon täyttävät kriteerit voidaan jakaa kahteen ryhmään, jotka ovat alueen suhteen globaalit ja paikalliset kriteerit. Globaaleilla kriteereillä aluetta laajennetaan jonkin koko aluetta koskevan informaation perusteella ja vastaavasti paikallisilla kriteereillä vain tarkasteltavan pikselin naapuruston perusteella. Eräs globaali kriteeri on tarkasteltavan pikselin ja koko alueen pikseleiden keskiarvon erotus. Tällöin alueen intensiteetistä voimakkaasti poikkeavat pikselit voidaan rajata alueen ulkopuolelle. Oleellista tällöin on alueeseen kuulumisen ilmaisevan kynnysarvon valinta. Toinen globaali kriteeri voi olla se, kuinka suuri *varianssi* (*variance*) alueen pikseleihin muodostuu, jos tarkasteltava pikseli lisättäisiin alueeseen. Varianssilla tarkoitetaan pikseleiden arvojen keskimääräistä eroavaisuutta keskimääräisestä pikseliarvosta. Muodollisemmin ilmaistuna varianssi δ^2 lasketaan kaavalla:

$$\delta^2 = 1/N \sum_{i=1}^N (L_i - L_{\text{avg}})^2 \quad (5.3)$$

jossa L_{avg} on alueen L pikseleiden arvojen keskiarvo ja N on pikseleiden lukumäärä.

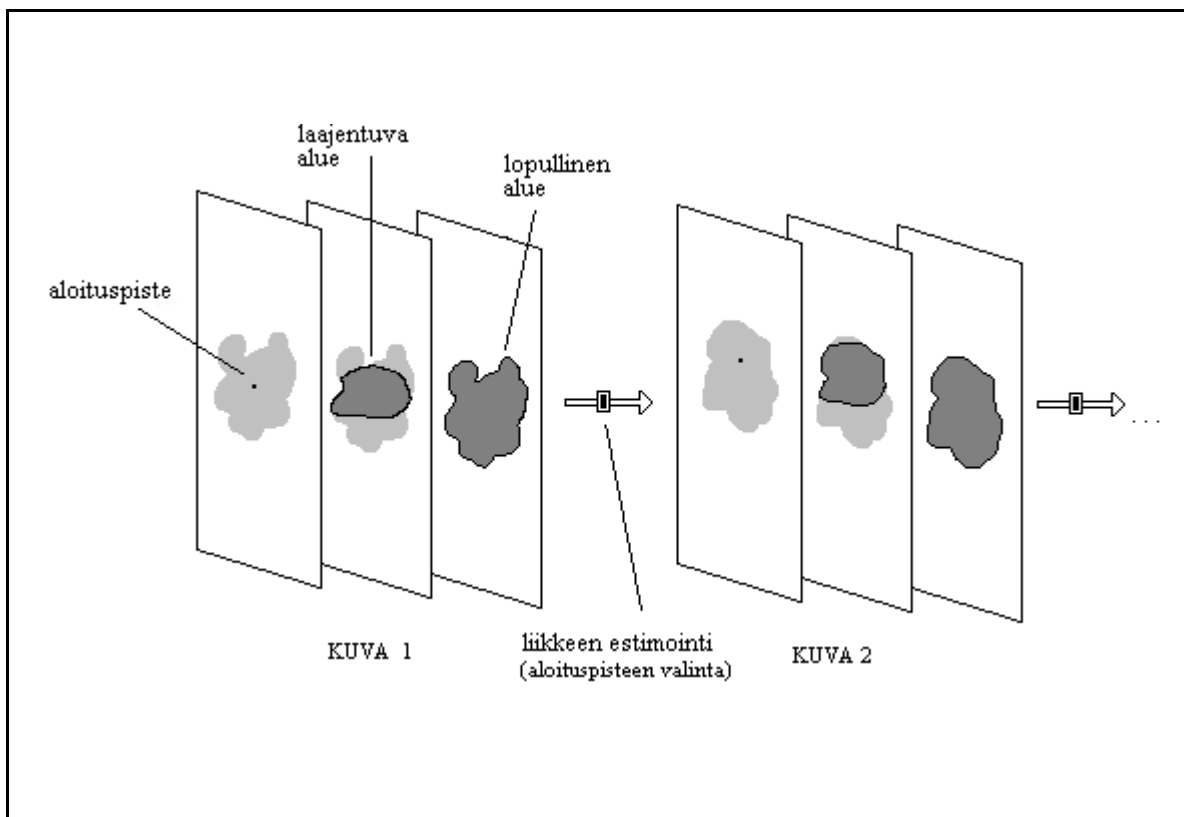
Mainittakoon, että varianssista δ^2 saadaan *keskihajonta* δ (*standard deviation*) ottamalla varianssista neliöjuuri.

Paikallisissa kriteereissä pikselin kuulumisen alueeseen päätellään tarkasteltavan pikselin naapuripikseleiden perusteella. Yksinkertaisin tapa on laskea tarkasteltavan pikselin ja sitä lähimmän alueen pikselin erotus. Ongelmana on laajennusehdon perustuminen vain yhden pikselin arvoon. Ratkaisuna tähän voidaan laajentumisehdossa hyödyntää tarkasteltavana olevan pikselin ympärillä olevia pikseleitä. Tällöin alueeseen kuulumisen kriteerinä voidaan

käyttää tarkasteltavan pikselin ja sen naapuripikseleiden varianssia. Tällöin tarkasteltava pikseli voidaan lisätä alueeseen, jos varianssi on alhainen eli tarkasteltava pikseli ja sen naapuripikselit ovat samankaltaisia. Eräs paikallinen kriteeri on pikselin *gradientti* (*gradient*), joka ilmaisee intensiteetin vaihtumiskohdan eli alueen reunan. Jos gradientti on alhainen, niin pikseli ei kuulu reunaan ja se voidaan lisätä alueeseen. Gradientti voidaan laskea suodattamalla kuvataso esimerkiksi Sobelin maskeilla. [FRÄ95]

Lopetusehto voidaan yksinkertaisimmillaan määrittellä siten, että alueen laajennus loppuu kun yhtäkään alueen reunalla olevaa pikseliä ei enää voida lisätä alueeseen. Edellistä tarkempi lopetusehdon määrittely on kuitenkin tarpeen, ainakin virheellisen laajentumisen estämiseksi. Se voi perustua esimerkiksi alueen kokoon, jolloin aluetta laajennetaan kunnes alueen koko saavuttaa oletetun objektin maksimikoon.

Alueen laajentamiseen perustuva objektinseuranta kuvasarjassa tapahtuu vastaavalla tavalla kuin aktiivisen reunan tapauksessa. Kuvasarjan ensimmäisessä kuvassa valitaan aloituspiste, jonka jälkeen alue laajennetaan. Ensimmäisen kuvan aloituspiste voidaan antaa



Kuva 5.4 Alueenlaajennus kuvasarjassa

manuaalisesti. Näin saadun alueen perusteella voidaan laskea aloituspiste kuvasarjan seuraavassa kuvassa. Aloituspisteenä voi toimia esimerkiksi alueen keskipiste. Kuvasarjan seuraavissa kuvissa onkin jo mahdollista hyödyntää objektin liikeinformaatiota. Esimerkiksi objektin translaatio on syytä laskea aloituspistettä valittaessa, ettei sitä valita objektin ulkopuolelta. Myös etsintäalue voidaan rajata liikeinformaation perusteella. Tällöin etsintäalueen rajat määrittelevät osaltaan lopetusehdon. Kuvassa 5.4 selvitetään miten alueenlaajennus toimii kuvasarjassa.

Sumea alueen laajennus

Genovan yliopistossa Italiassa on kehitetty alueenlaajennustekniikka, jossa pikseleiden kuulumisen laajennettavaan alueeseen määritellään sumeaan logiikkaan perustuen [FON93]. Siinä jokaiselle pikselille lasketaan alueeseen kuulumisen aste eli *jäsenyysaste* (*degree of membership*), jonka perusteella pikselin kuulumisen alueeseen päätellään. Tuloksena on oikeastaan sumea kartta, jonka avulla voidaan sovelluskohtaisen informaation perusteella valita kaikkein tarkoituksenmukaisin alue.

Ennen varsinaista alueen laajentamista jokaiselle etsintäalueen pikselille lasketaan jäsenyyden aste, jonka perusteella laajentuminen tapahtuu. Tämänlaista menetelmää, jossa alueen laajeneminen perustuu pikseleille asetettuun arvoon, kutsutaan *merkintäpohjaiseksi menetelmäksi* (*label-driven method*). Jäsenyyden asteen määräävät yhdessä sekä paikallinen, että globaali kriteeri. Paikallisen kriteerin perusteella tarkasteltavan pikselin jäsenyyden aste on korkea, jos tarkasteltava pikseli on riittävän samankaltainen sen jo merkittyjen naapuripikseleiden kanssa. Globaali kriteeri perustuu aloitusalueesta saatuun informaatioon. Aloitusalueen oletetaan kuuluvan objektiin. Yksinkertaisimmillaan aloitusalueen voivat muodostaa aloituspisteen naapuripikselit. Aloitusalueen avulla voidaan jäsenyyden asteita laskettaessa hyödyntää esimerkiksi objektin pikseleiden keskimääräistä intensiteettiä. [FON93]

Paikalliseen informaatioon perustuvassa jäsenyyden laskemisessa voidaan hyödyntää myös pikseleiden *paikallista kontekstia* (*local context*). Tällä tarkoitetaan sitä, että tarkkaillaan paikallisen alueen pikseleiden sijaintia suhteessa toisiinsa. Tällöin voidaan etsiä pikseleiden

muodostamia rakenteita. Eräs lähestymistapa voisi olla tekstuurin analysointiin perustuvat menetelmät [JEN93].

Kun pikseleiden jäsenyysasteet on saatu lasketuksi, voidaan alueen laajentaminen aloittaa aloitusalueesta. Alueen laajentaminen tapahtuu siten, että alueen reunoilla olevien pikseleiden jäsenyysaste kertoo laajentumisen suunnan tai sumeassa tapauksessa se kertoo laajentumisen voimakkuuden. Laajentumista voidaankin ohjalla jäsenyysasteeseen vertailtavilla kynnsarvoilla. Sumean alueenlaajentamisen suurin etu onkin mahdollisuus dynaamiseen alueen laajentamisen. Esimerkiksi seurattaessa objektia kuvasarjassa, jossa valaistusolosuhteet vaihtelevat, voidaan alueen laajentumista ohjalla valaistuksen vaikutusten mukaan.

6. OBJEKTINSEURANTAJÄRJESTELMÄ

Tässä luvussa tarkastellaan objektinseurantajärjestelmää, joka on toteutettu luvussa 3 kuvatus yleisen järjestelmämallin ja luvussa 4 kuvattujen lohkotäsmäysmenetelmien perusteella. Tarkoituksena on vertailla erilaisten täsmäysmenetelmien toimivuutta.

6.1 Järjestelmän kuvaus

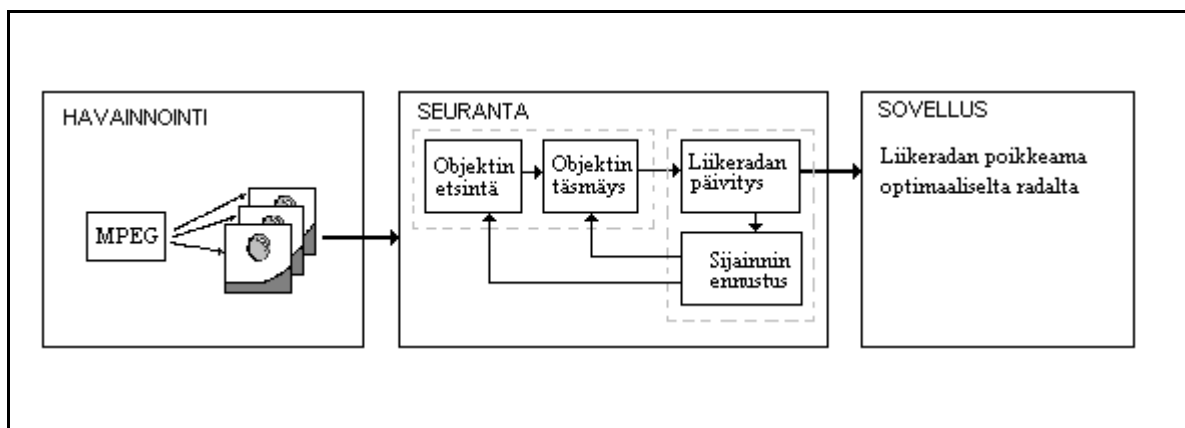
Toteutetun järjestelmän tarkoituksena on etsiä kuvasarjan jokaisesta kuvasta ennalta määritelty objekti. Järjestelmä ei siis itse havaitse seurattavia objekteja, eikä seurattavia objekteja ole kuin yksi. Järjestelmä on kirjoitettu C++ - kielellä Unix - käyttöjärjestelmässä ja se toimii X-window system - käyttöliittymän avulla. Seurantaan vaikuttavat parametrit, kuten haluttu täsmäysalgoritmi, annetaan komentorivillä ja seurannan eteneminen esitetään visuaalisesti X-ikkunassa piirtämällä suorakaide täsmätyn alueen ympärille.

Havaintoaineistona käytetään MPEG-menetelmällä tiivistettyä kuvasarjaa, jonka jokainen kuva voidaan purkaa yksitellen kuvatasoksi. Kuvataso muodostuu punaisesta, vihreästä ja sinisestä väritasosta eli RGB - tasoista. Näiden tasojen lisäksi talletetaan yksi Y-taso, joka ilmaisee kuvatason *luminanssin* (*luminance*) eli intensiteetin vaihtelun. Luminanssi lasketaan väri-informaation perusteella seuraavalla kaavalla:

$$Y = (0.30 * \text{Red} + 0.6 * \text{Green} + 0.1 * \text{Blue}) \quad (6.1)$$

Tarkoituksena on tarkastella vaikuttaako väri-informaation käyttäminen parantavasti täsmäykseen verrattuna pelkän luminanssin käyttämiseen. Kuvatasoja voidaan siis käsitellä väri-informaatioon perustuen. Kuitenkin testiympäristön teknisistä ominaisuuksista johtuen X-ikkunaan siirretään vain luminanssi-taso, joten tarkasteltava kuvataso näkyy harmaan sävyinä. Tästä huolimatta täsmäys on mahdollista suorittaa väri-informaatioon perustuen. Objektlohkon määrittelyyn käytetään samaa tietorakennetta kuin kuvatason määrittelyyn, joten vertailu on mahdollista suorittaa väritaso kerrallaan.

Järjestelmä koostuu kolmesta objektinseurantamoduulista, jotka ovat ennustus-, etsintä- ja täsmäysmoduuli. Ennustusmoduuli tuottaa ennusteen objektin mahdollisesta sijainnista objektin aikaisemman käyttäytymisen perusteella. Etsintämoduulin tehtävänä on määrittellä etsintäalue tarkasteltavassa kuvatasossa ja täsmäysmoduulin tehtävänä on määrittää seurattavan objektin sijainti etsintäalueella. Lisäksi havaintoaineiston käsittelylle ja seurannan tulosten käsittelylle on oma moduulinsa. Kuvassa 6.1 nähdään järjestelmän modulaarinen rakenne.



Kuva 6.1 Objektinseurantajärjestelmän modulaarinen rakenne

Ennustusmoduuli

Ennustusmoduuli huolehtii objektin liikkeen estimoinnista. Käytetty liikkeen estimoinnin lähestymistapa perustuu suoraviivaisen liikkeen tunnistamiseen. Moduuli laskee objektin ennustetun sijainnin liikevektorin avulla. Liikevektori lasketaan vähentämällä objektin nykyisen sijainnin koordinaateista objektin edellisen sijainnin koordinaatit. Näin laskettu liikevektori ilmaisee objektin siirtymän kahden peräkkäisen kuvatason välillä. Objektin ennustettu sijainti saadaan lisäämällä liikevektori objektin nykyiseen sijaintiin. Aikaisemmin luvun kolme kuvassa 3.2 esitettiin visuaalisesti liikevektorin muodostusta.

Tämän liikkeen estimoinnin mallina on suoraviivainen liike. Objektin oletetaan jatkavan liikettään samansuuntaisesti kuin kahden edellisen kuvatason välillä. Käytännössä kuitenkin objekti voi tehdä manöövereita eli käännöksiä, sillä suoraviivainen liike lasketaan vain kahden kuvatason välillä. Tilanne on sama kuin maanmittarilla, joka määrittelee suoraa viivaa vain kahden mittamerkin avulla. Objektinseurannan kannalta tulos on haluttu, sillä

vaikeiden manöövereitä tunnistavien algoritmien sijaan tällä yksinkertaisella liikkeen estimoinnin menetelmällä voidaan kääntyilevän objektin sijainti ennustaa jonkinlaisella tarkkuudella. Oletuksena on kuitenkin objektin tasainen liike. Esimerkiksi jyrkät suunnan muutokset tuottavat ongelmia tämänkaltaiselle algoritmilta. Sanottakoon algoritmin aikavaativuudesta, että se toimii vakioajassa eli ajassa $O(1)$. Tähän ei kuitenkaan ole otettu mukaan etsintä- ja täsmäysalgoritmien viemää aikaa.

Etsintämoduuli

Etsintämoduuli sisältää funktiot etsintäalueen valitsemiseen. Järjestelmässä on kaksi vaihtoehtoa etsintäalueen valintaan. Ensimmäisessä vaihtoehdossa etsintäalueeksi valitaan koko kuvataso ja toisessa vaihtoehdossa etsintäalue voidaan rajata ennustetun sijainnin perusteella. Kun etsintäalueena on koko kuvataso, niin täsmäys voidaan suorittaa optimaalisesti koko kuvataason suhteen. Toisin sanoen saadaan pienin mahdollinen täsmäyksen virhearvo. Tämä ei kuitenkaan ole aina paras mahdollinen tilanne, sillä objektilohko saattaa täsmätä parhaiten jonkin toisen samanlaisen objektin tai taustaan kuuluvan alueen kanssa.

Ennusteeseen perustuvassa etsintäalueen rajauksessa oletetaan, että objekti löytyy ennustetun sijainnin läheisyydestä. Tällöin etsintäalueen määrittelee suorakaide, joka ylettyy objektin nykyisestä sijainnista objektin ennustettuun sijaintiin asti. Näin ollen etsintäalue mukautuu objektin liikkeen suuntaan ja myös sen nopeuteen. Mitä nopeampi liike, sen laajempi on etsintäalue liikkeen suuntaan. Tämän lisäksi etsintäaluetta laajennetaan vielä minimaalisen siirtymän verran. Minimaalisella siirtymällä tarkoitetaan siirtymää, jonka objektin oletetaan tekevän liikevektorin ollessa nolla. Toteutetussa järjestelmässä minimaalinen siirtymä lasketaan ottamalla maksimi objektin korkeudesta ja leveydestä. Minimaalisen siirtymän lisääminen etsintäalueen reunoille antaa pienen virhemarginaalin ennustukselle. Esimerkiksi, jos ennustetun objektin liikkeen suunta on ennustettu väärin, niin objekti voi vielä löytyä etsintäalueen reunoilta. Etsintäaluetta ei kuitenkaan kannata laajentaa liikaa, sillä muuten rajaamalla saavutettu hyöty jää vähäiseksi. Moniobjektiseurannassa rajattua etsintäaluetta kutsutaan hyväksymisalueeksi. Etsintäalueen rajaamisella liikkeen ennustuksen perusteella saavutetaan myös toinen etu tarkkuuden

lisäksi. Etsintään kuuluva suoritusaika vähenee käytännössä oleellisesti verrattuna etsintään, jossa koko kuvataso käydään läpi. Algoritmin pahimman tapauksen aikavaativuus on kuitenkin edelleen $O(nm)$, jossa n on objektin pikseleiden ja m kuvataason pikseleiden lukumäärä. Pahimmassa tapauksessa etsintäalue voi olla koko kuvataso. Näin on esimerkiksi tapauksessa, jossa objektilohko tai sen liikkeen laajuus on lähes yhtä suuri kuin kuvataso. Käytännössä seurattavat objektit ovat pieniä verrattuna kuvatasoon. Tällöin etsintäalueen rajaaminen nopeuttaa huomattavasti objektin etsintää.

Täsmäysmoduuli

Täsmäysmoduulissa määritellään algoritmit joiden avulla objektilohkon täsmäys suoritetaan. Toteutetut täsmäysalgoritmit perustuvat absoluuttisen keskivirheen, neliöllisen keskivirheen, painotetun neliöllisen keskivirheen, laplace-suotimen amplitudin, histogrammin ja varianssin laskemiseen. Täsmäyksessä on mahdollista käyttää väri-informaatiota tai luminanssia. Jos käytetään väreihin perustuvaa täsmäystä, niin jokainen väritaso täsmätään erikseen ja täsmäysten tuloksista otetaan keskiarvo. Tämä poikkeaa hieman kaavan 4.3 periaatteesta, jossa täsmäys perustuu värivektorien pituuksien vertailuun.

Absoluuttinen keskivirhe tuottaa objektilohkon ja vertailtavan alueen pikseleiden keskimääräisen eroavaisuuden. Vastaavasti toimii neliöllinen keskivirhe korostamalla suuria eroavaisuuksia laskemalla erotuksen neliöllisesti. Painotettu neliöllinen keskivirhe lasketaan korostamalla tiettyjä objektilohkon pikseleitä erillisen painomatriisin avulla. Painomatriisin tarkoituksena on vähentää objektilohkossa esiintyvän taustainformaation vaikutusta. Toteutetussa järjestelmässä painomatriisi muodostetaan siten, että objektilohkon keskustaa painotetaan enemmän kuin reunoja. Matriisin keskipistettä painotetaan kertoimella kaksi. Esimerkiksi 7×7 kokoiselle objektilohkolle saadaan kaavalla 6.3 seuraava painomatriisi:

$$\text{WEIGHT MATRICE (7x7) = } \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 4 & 4 & 4 & 4 & 1 \\ 1 & 4 & 16 & 16 & 16 & 4 & 1 \\ 1 & 4 & 16 & 64 & 16 & 4 & 1 \\ 1 & 4 & 16 & 16 & 16 & 4 & 1 \\ 1 & 4 & 4 & 4 & 4 & 4 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Laplace-suotimen avulla voidaan erottaa kuvasta siinä olevat reunat. Tällöin on mahdollista käyttää täsmäyksessä hyväksi pikseleiden sijaintia suhteessa toisiinsa eli seurattavan objektin muotoa. Koko kuvataso ja objektilohko suodatetaan seuraavalla maskilla [HAV95]:

$$\text{Mask} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Tällöin tuloksena on kuvataason ja objektilohkon amplitudikentät, jossa suuri amplitudi ilmaisee reunaan. Koska Laplace-suodin antaa negatiivisen ja positiivisen amplitudin reunan kohdalla, vain positiivinen amplitudi otetaan mukaan laskuihin [HAV95]. Kun amplitudikentät on saatu muodostettua, suoritetaan täsmäys käyttäen absoluuttista keskivirhettä. Laplace-suotimen etuna on sen kohtuullinen nopeus, sillä se perustuu vain yhteen maskiin. Haittana on Laplace-suotimelle ominainen voimakas häiriöalttius, joten täsmäyksen ei voi olettaa olevan erityisen tarkka.

Histogrammiin ja varianssiin perustuvien täsmäysmenetelmien avulla voidaan täsmäyksessä käyttää hyväksi objektilohkon ja vertailtavan kuva-alueen tilastollisia ominaisuuksia. Täsmäys ei siis perustu yksittäisten pikseleiden samuuteen. Histogrammin avulla täsmäys suoritetaan siten, että jokaiselle vertailtavalle kuvataason lohkolle muodostetaan histogrammi, jota vertaillaan objektilohkon histogrammiin. Histogrammien vertailu tapahtuu neliöllisen keskivirheen periaatteella. Perusajatuksena on se, että kuvatasosta pyritään löytämään lohko, jossa pikseleiden arvojakauma on mahdollisimman lähellä objektilohkon arvojakaumaa. Varianssiin perustuva täsmäys tapahtuu laskemalla objektilohkolle ja vertailtaville lohkoille varianssi. Objektilohkon täsmäys vertailtavan kuvataason lohkon kanssa tapahtuu minimoimalla objektilohkon ja vertailtavien lohkojen varianssien erotus.

Liikkeen estimointia hyödynnetään myös täsmäyksessä. Tällöin käytetään lähimmän naapurin tekniikkaa siten, että täsmäyksen tuottamaan virheeseen lisätään vielä etäisyysvirhe. Etäisyysvirhe lasketaan objektin ennustetun sijainnin ja tarkasteltavan lohkon

välisenä välimatkana. Tämä vähentää muun muassa väärän täsmäyksen mahdollisuutta, kun näkökentässä on kaksi samankaltaista objektiä.

6.2 Mittaustuloksia

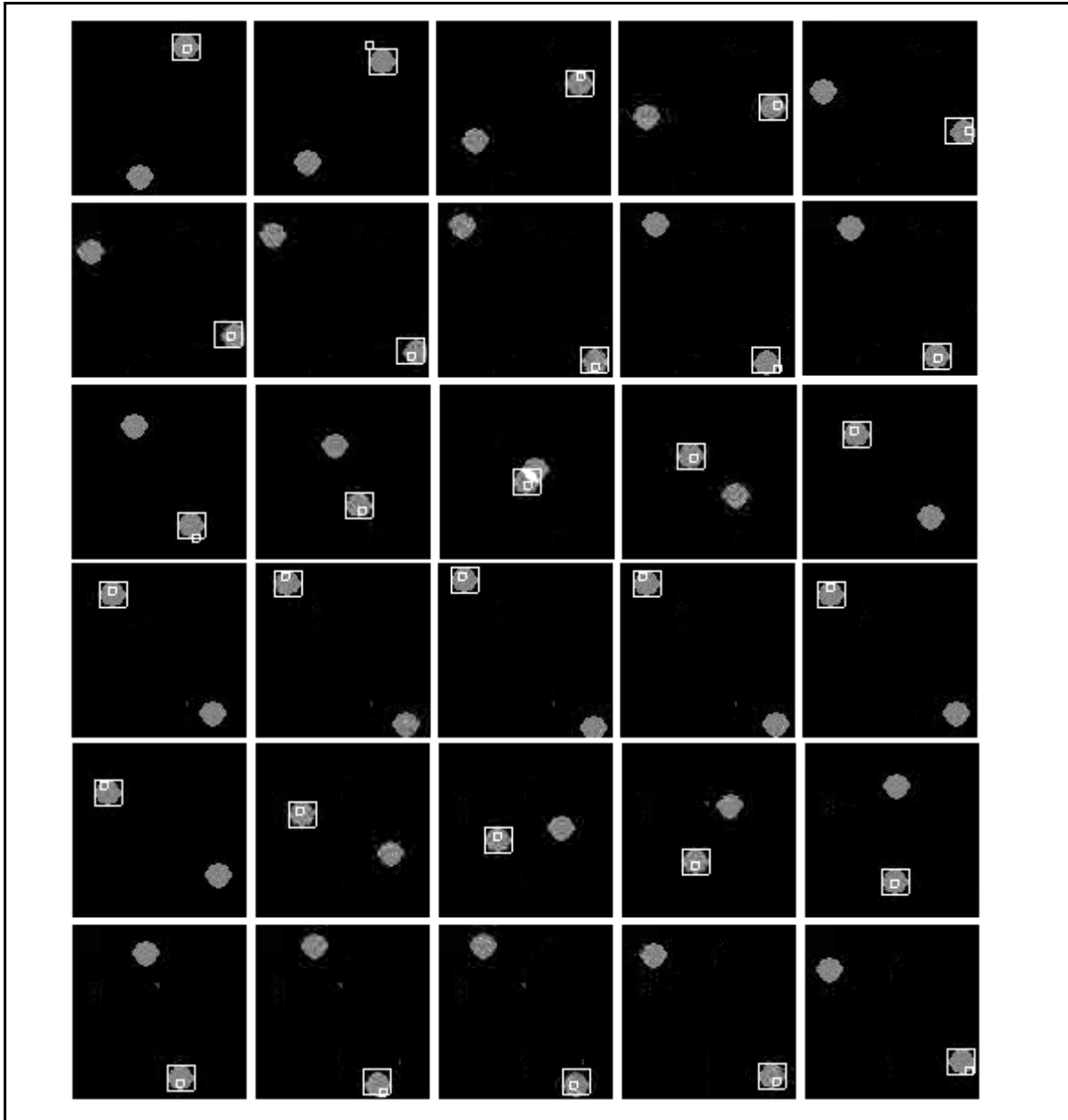
Objektinseurantamenetelmien keskinäistä paremmuutta vertaillaan soveltamalla menetelmiä kahteen kuvasarjaan. Mitattavia muuttujia on kaksi. Ensimmäinen muuttuja ilmaisee miten hyvin menetelmä kykenee seuraamaan objektiä. Tämä lasketaan vertailemalla objektin eri menetelmien avulla saatua objektin liikerataa quasi-optimaaliseen liikerataan. Tämä quasi-optimaalinen liikerata on saatu tallettamalla parhaimman täsmäyksen saavuttaneen menetelmän tuottama liikerata ja korjailemalla sitä manuaalisesti siten, että kunkin kuvatason kohdalla liikeradan koordinaatit ovat seurattavan objektin keskikohdassa. Kaikkien menetelmien liikeratoja vertaillaan näin muodostettuun liikerataan.

Järjestelmä tallettaa seurannan tuottaman liikeradan tulostamalla tiedostoon täsmätyn lohkon keskipisteen koordinaatit jokaisessa kuvatasossa. Seurannan onnistumista voi tarkkailla myös visuaalisesti. Jokaiseen kuvatasoon merkitään suorakaiteena etsintäalue, ennustettu sijainti ja täsmätty lohko. Ennustettu sijainti ilmaistaan pienellä suorakaiteella.

Toinen mittauksessa käytetty muuttuja on aika. Tarkoituksena on mitata seurannan aloituksesta sen lopettamiseen kuluva aika. Kaikki menetelmät käyvät koko kuvasarjan läpi. Aika mitataan Unix-käyttöjärjestelmän time-komentoon perustuvalla C++ - kielisellä mittaussluokalla Timer, josta enemmän viitteessä [HAV95].

Kuvasarja I

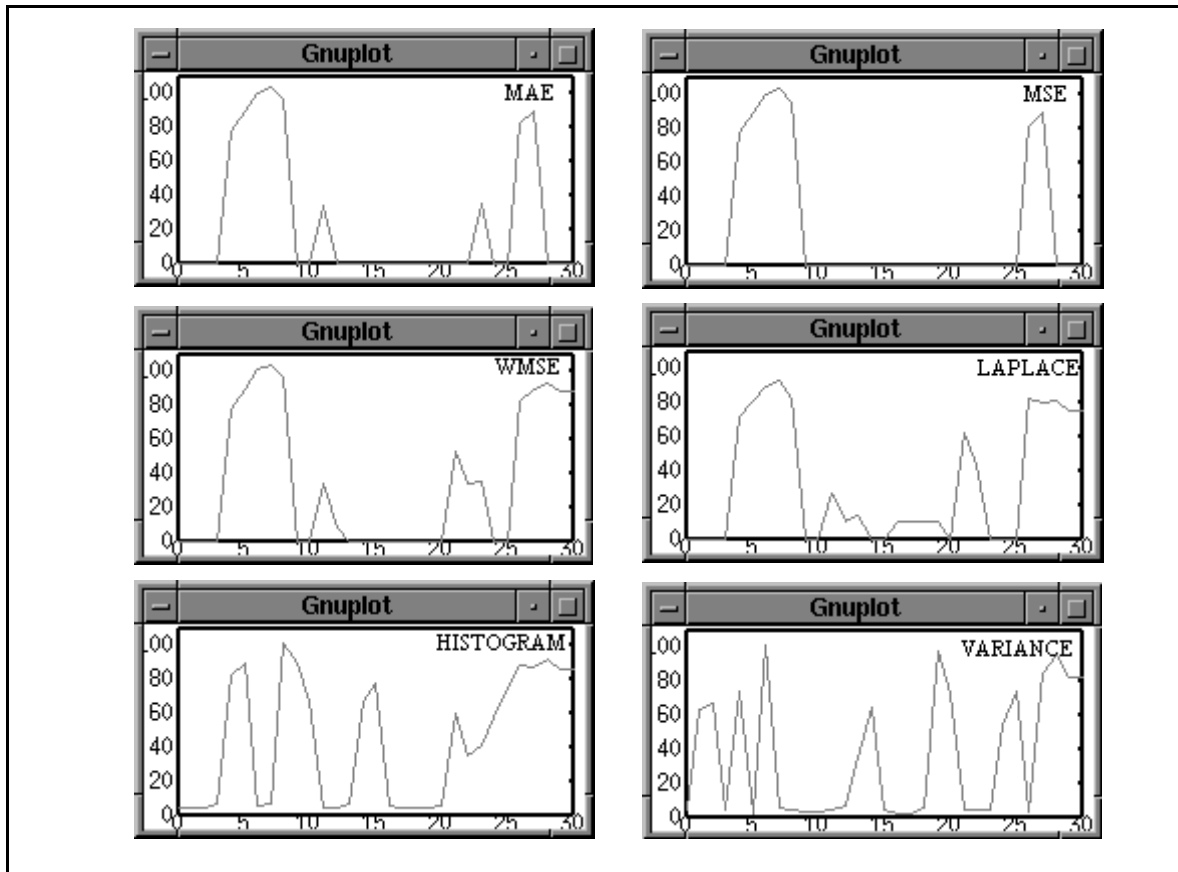
Ensimmäinen mittauksissa käytettävä kuvasarja on synteettinen kuvasarja, jossa kaksi harmaata palloa liikkuu mustalla taustalla. Objekt lohkon määrittelee pallon minimaalinen kehys. Objekti ei ole koko kuvasarjan ajan tasaisen harmaa, vaan intensiteetti vaihtelee jonkin verran. Objekt lohko määritellään staattiseksi. Kuvasarjan ensimmäisestä kuvasta valitaan lohko, joka kopioidaan objekt lohkoksi. Tämä objekt lohko ei muutu seurannan



Kuva 6.2 Kuvasarja I - objektien liikeradat risteävät

aikana. Objektilohkon valinta tapahtuu manuaalisesti ja objektilohkon sijainti kuvasarjan ensimmäisessä kuvassa ilmoitetaan järjestelmälle seuranta aloitettaessa.

Ongelmallista tässä kuvasarjassa on kahden samankaltaisen objektin esiintyminen näkökentässä. Seurannan vaikeutta lisää pallojen liikeratojen risteäminen kuvasarjan puolella välissä. Tarkoituksena on seurata vain toista palloista. Tällöin toisen, lähes identtisen pallon ilmaantuminen etsintäalueelle, tulkitaan ympäristön ilmiönä. Kuvassa 6.2 nähdään kuvasarja I ja liikerata, joka on saatu käyttämällä neliöllistä keskivirhettä.



Kuva 6.3 Seurannan onnistuminen kuvasarjassa I, kun etsintäalueena on koko kuvataso

Etsintäalue on rajattu liikkeen estimoinnin avulla. Pieni suorakaide ilmaisee lohkon keskipisteen ennustetun sijainnin ja suurempi suorakaide ilmaisee ennusteen avulla täsmätyn lohkon.

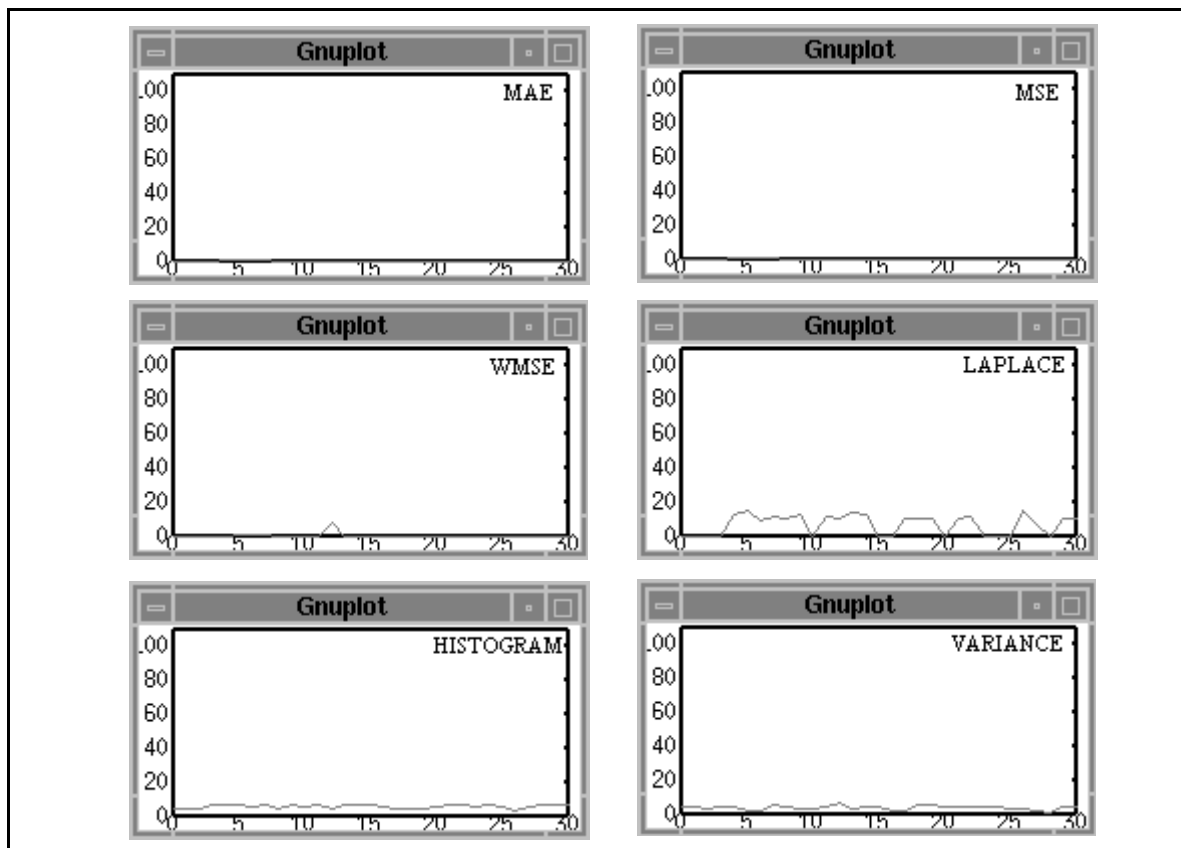
Kuvassa 6.3 tarkastellaan seurannan onnistumista jokaisen menetelmän osalta, kun etsintäalueena on koko kuvataso. Käyrät ilmaisevat poikkeamat optimaaliselta radalta. X-akseli ilmaisee kuvatason numeron. Kuvatasojen numerointi alkaa nolasta ja kuvatasoja on yhteensä kolmekymmentä. Kuten huomataan, yksikään seurantamenetelmä ei selvinnyt kunnolla kahden lähes identtisen objektien esiintymisestä. Jokainen seurantamenetelmä kyllä täsmäsi kohtalaisen hyvin objektiin, vaikkakin aika ajoin väärän objektin kanssa. Täsmäys ei epäonnistunut kertaakaan siten, että täsmäyksen kohteena olisi ollut jokin satunnainen taustaan kuuluva alue. Toisaalta toisen objektin voidaan olettaa kuuluvan seurattavan objektin taustaan.

Käyrien piikit johtuvat siitä, että objektilohko on täsmännyt väärän objektin kanssa.

Parhaiten menestyi neliöllinen keskivirhe ilman painotusta. Vertailemalla neliöllisen virheen tuottamaa tulosta absoluuttisen virheen tulokseen, voidaan todeta neliöllisen keskivirheen paremmuus.

Testien tuloksista voidaan päätellä, että koko kuvatason käyttäminen etsintäalueena ei ole käytännöllistä. Virhe kuitenkin aiheutui ainoastaan toisen samankaltaisen objektin esiintymisestä etsintäalueella. Väärän täsmäyksen ongelma poistui pikselitason virhefunktioiden osalta, kun täsmäyksessä otettiin mukaan etäisyysvirhe laskettuna objektin ennustetusta sijainnista. Tällöin seuranta antoi lähes optimaalisen liikeradan poikkeamien ollessa lähinnä pikselin luokkaa. Laplace-, histogrammi- ja varianssi - tekniikat eivät sen sijaan hyötäneet vastaavasti etäisyysvirheestä.

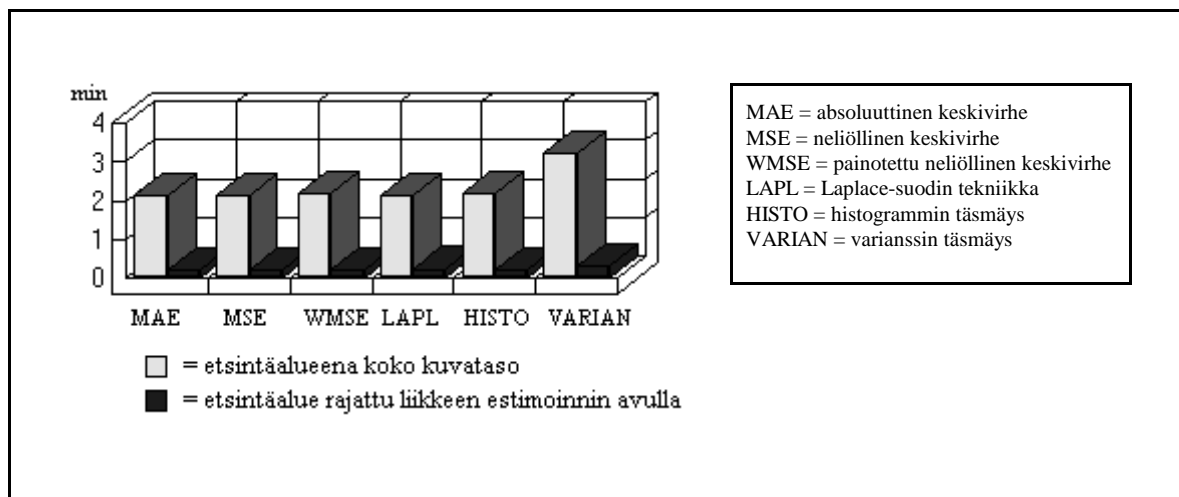
Seurannan tarkkuus parani oleellisesti, kun etsintäaluetta rajattiin liikkeen estimoinnin avulla. Yksikään seurantamenetelmistä ei täsmännyt enää toistuvasti väärään objektiin. Ainoastaan painotettu neliöllinen keskivirhe täsmäsi väärin objektien kohdatessa. Silloinkin seuraava täsmäys oli jälleen seurattavaan objektiin. Kuvassa 6.4 nähdään poikkeamat



Kuva 6.4 Seurannan onnistuminen, kun etsintäalue rajataan liikkeen estimoinnin avulla

optimaaliselta liikeradalta käyttäen liikkeen estimointia etsintäalueen rajaamiseen. Painotetun neliöllisen virheen vääriä täsmäyksiä objektien kohdatessa näkyy pienenä piikkinä kuvasarjan kuvatason 12 kohdalla. Myös laplace-suotimeen perustuva täsmäisyys seurasi kohtuullisen hyvin objektia. Laplace-tekniikan ongelma on kuitenkin täsmäyksen pieni epätarkkuus. Tällöin objektilohko täsmäsi usein hieman objektin ohi. Kertaakaan seuranta ei kuitenkaan täsmännyt varsinaisesti väärin. Sama virhemarginaali ilmeni tilastollisten menetelmien kanssa virheen ollessa hieman laplace-tekniikkaa vähäisempi.

Suoritusajat kunkin menetelmän osalta käyttäen kumpaakin etsintämetodia nähdään kuvassa 6.5. Eri tekniikkojen välillä ei ole mainittavia eroja suoritusajoissa. Varianssiin perustuvan täsmäyksen hieman muita suurempi suoritusajaksi johtuu osittain toteutuksesta.

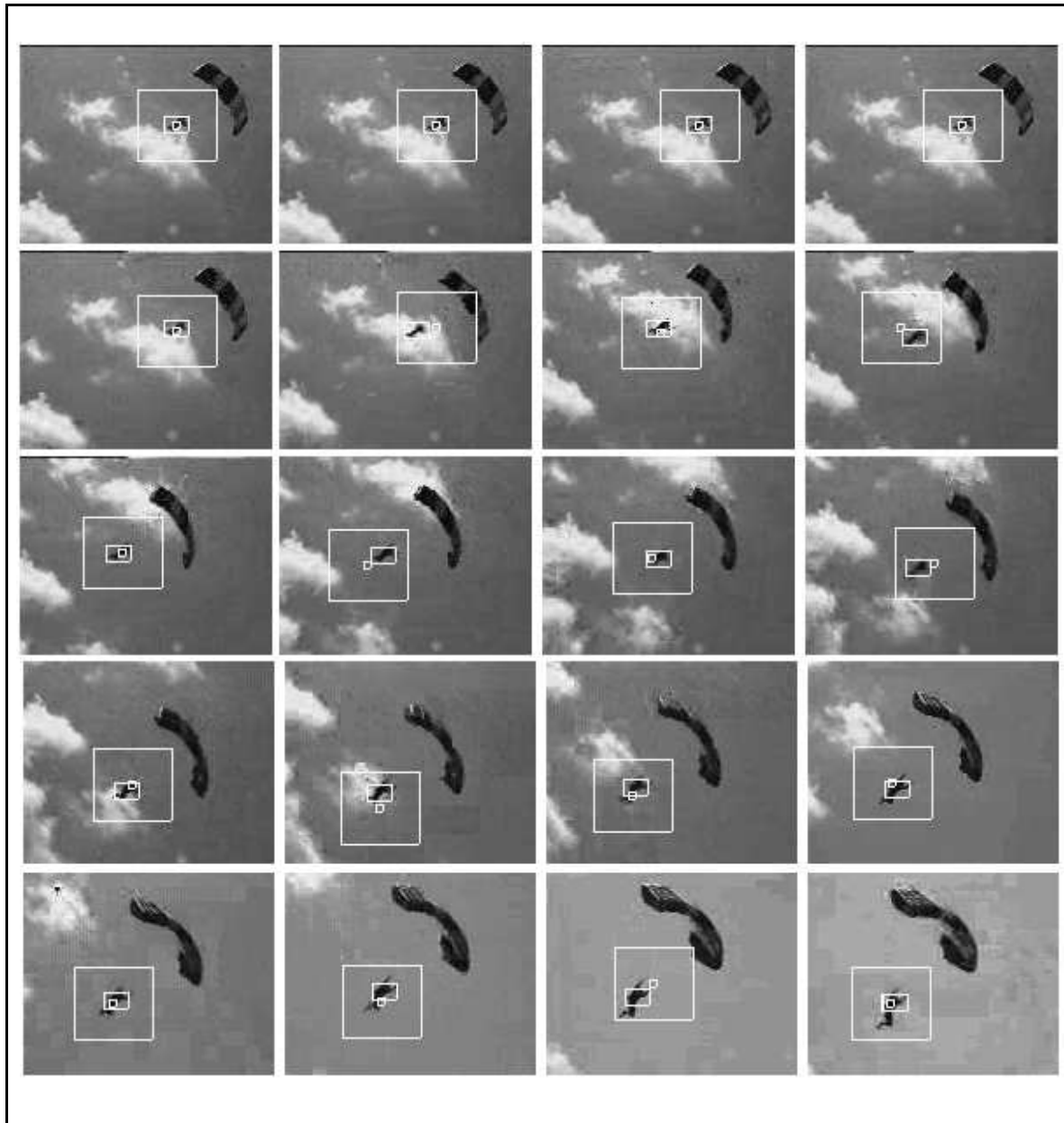


Kuva 6.5 Suoritusajojen perusteella etsintäalueen rajaaminen kannattaa

Keskimääräinen suoritusajaksi oli 2 minuuttia 30 sekuntia, kun etsintäalueena oli koko kuvataso. Keskimääräinen suoritusajaksi oli huomattavasti pienempi liikkeen estimointia hyödyntämällä. Keskimääräinen suoritusajaksi oli 30 sekuntia.

Kuvasarja II

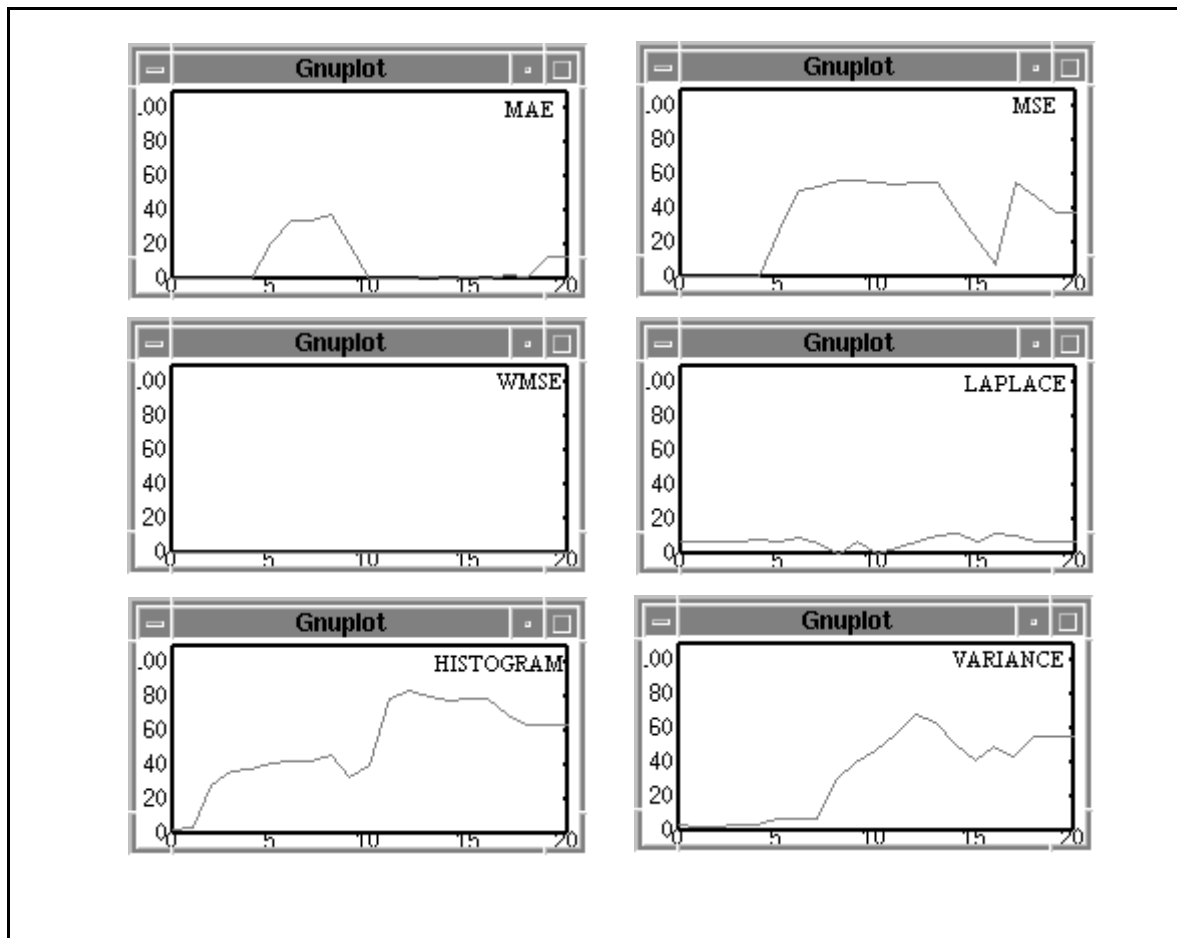
Toinen kuvasarja on seurannan kannalta haastava, sillä siinä esiintyy kolmiulotteista liikettä. Tässä kuvasarjassa laskuvarjohyppääjä laskeutuu varjollaan kohti maata. Seurattavana objektina on laskuvarjohyppääjä. Heti kuvasarjan alussa objekti liikkuu



Kuva 6.6 Laskuvarjohyppäjän seuraaminen onnistuu painotetulla MSE:llä

valkoisen pilven yli. Tällöin ongelmana on taustan voimakas muuttuminen. Tausta muuttuu muutenkin voimakkaasti kameran liikkeen vaikutuksesta. Laskuvarjohyppäjä-kuvasarja ja painotetulla neliöllisellä keskivirheellä saatu liikerata nähdään kuvassa 6.6.

Seurannan onnistumista eri menetelmillä tarkastellaan kuvassa 6.7. Absoluuttiseen neliövirheeseen perustuva seuranta suoriutuu kohtuullisesti. Kuitenkin kuvasarjan alkupuolella, kuudennessa kuvatasossa, laskuvarjohyppäjän kulkiessa valkoisen pilven yli objektilohko täsmää laskuvarjohon hyppäjän asemesta. Kun hyppäjä on kulkenut pilven yli



Kuva 6.7 Seurannan onnistuminen kuvasarjassa II

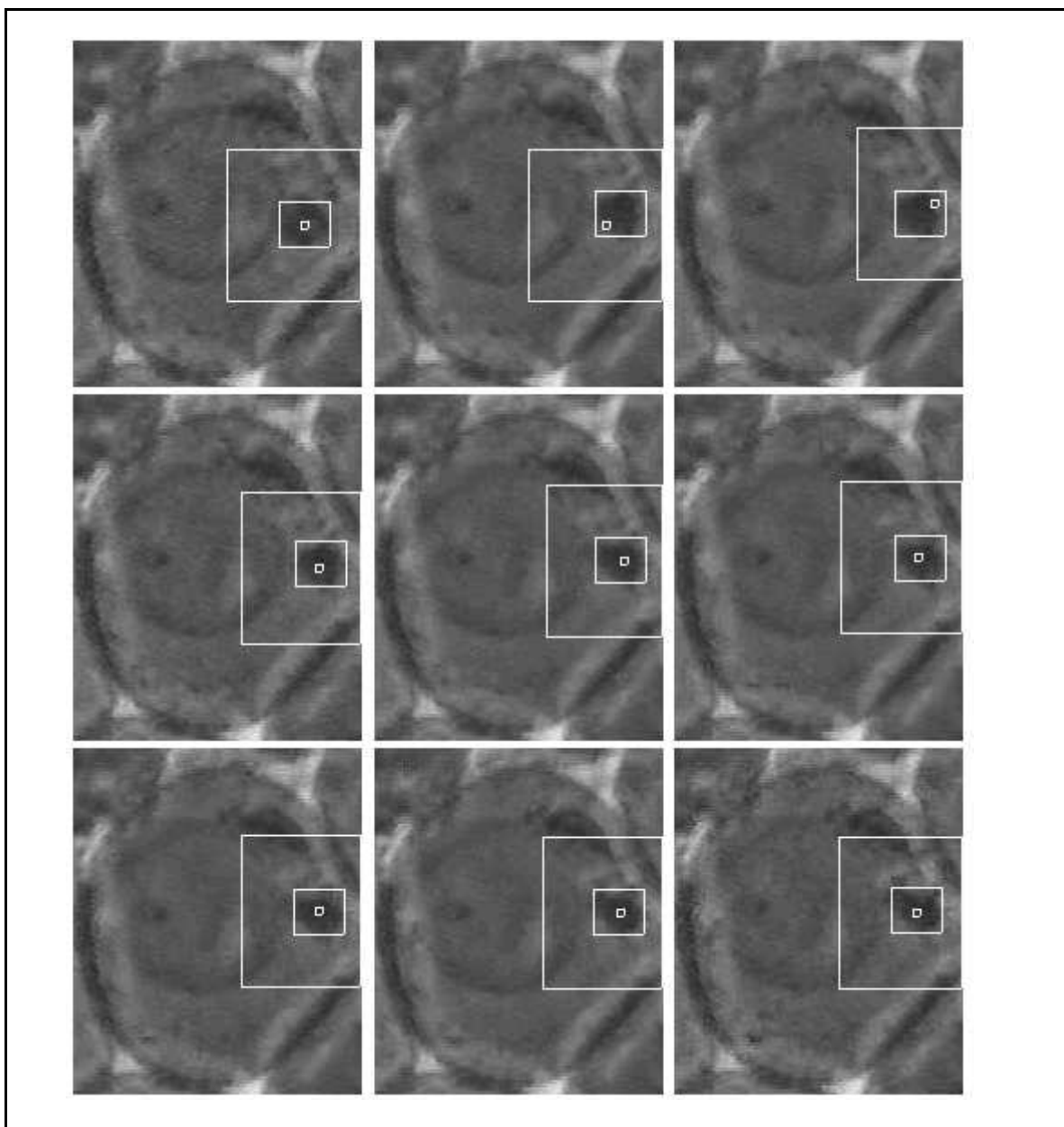
seuranta täsmää jälleen korrektisti. Sama ongelma ilmeni myös neliöllisen keskivirheen kanssa. Tällöin kuitenkin järjestelmä jäi seuraamaan laskuvarjoa täsmäämällä enää vain kerran lähelle objektia. Sekä absoluuttinen, että neliöllinen keskivirhe täsmäsivät kuvasarjan lopussa hyppääjän jalkoihin objektin keskipisteen asemesta. Tämä aiheutui staattisen objektilohkon täsmäyksestä mutatoivaan objektiin.

Tilanne parani oleellisesti käyttämällä painotettua neliöllistä keskivirhettä. Tällöin seuranta pysyi objektissa myös ongelmia tuottaneen pilven kohdalla. Hyvän tuloksen antoi myös laplace-suotimeen perustuva seurantamenetelmä, vaikka seurannassa esiintyikin pieni virhemarginaali. Itse asiassa laplace-tekniikan avulla seuranta pysyi koko ajan seurattavassa objektissa, joskin hieman sen reunoilla. Laskuvarjohyppääjän seurantaan huonosti soveltuviksi menetelmiksi osoittautuivat histogrammi ja varianssi - tekniikat. Järjestelmä kadotti hyppääjän miltei heti. Tämä johtui kuitenkin siitä, että seuranta lukittui laskuvarjoon. Mielenkiintoista seurannassa on se, että seurattava objektin liike on

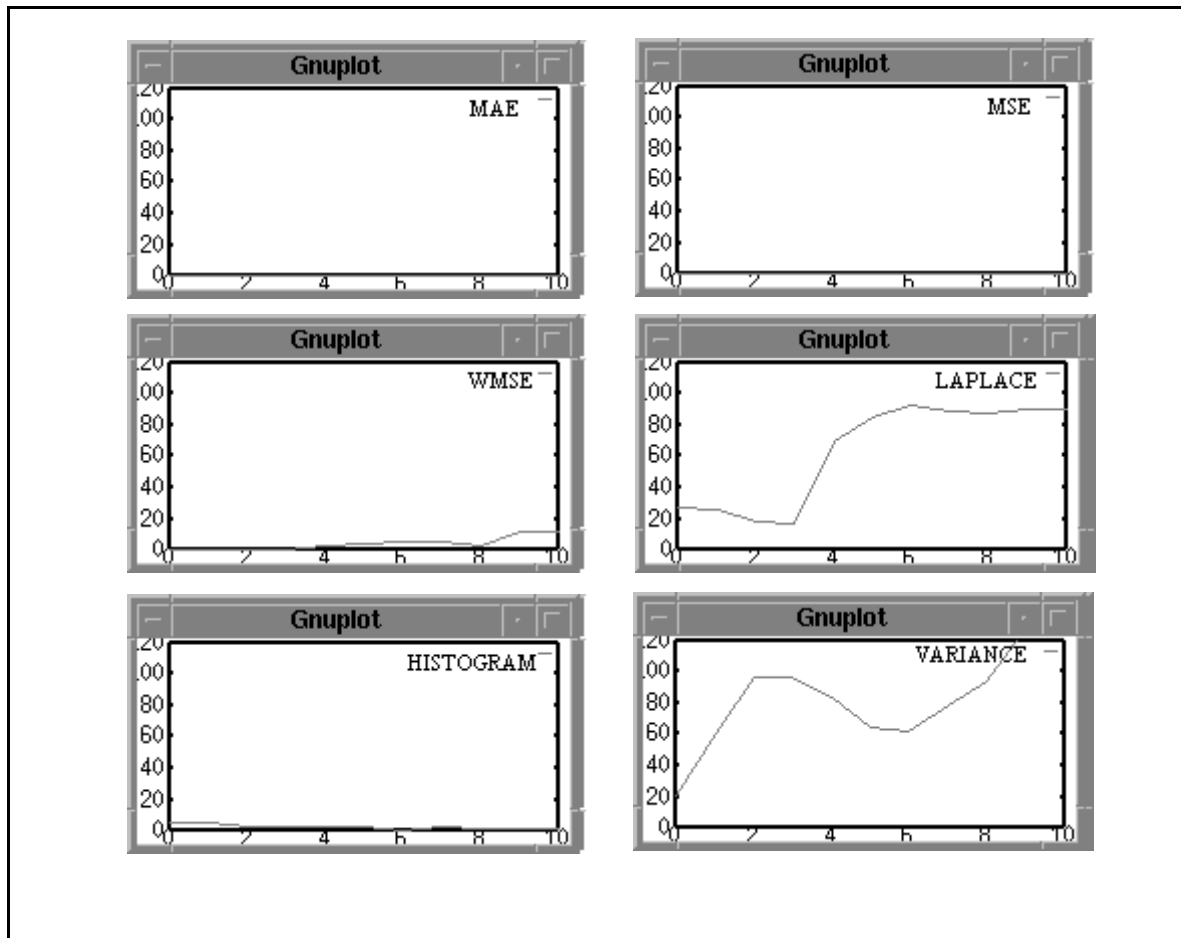
translaatiota ja mutaatiota. Mutatoivan objektin seuranta staattisen objektlohkon avulla on siis mahdollista. Tämä johtui siitä, että objekti erosi selkeästi taustasta.

Kuvasarja III

Kolmannessa kuvasarjassa tarkastellaan rotan solun osien liikkeitä. Seurattava objekti on kromatoidi, solun osa joka kuljettaa RNA-informaatiota paikkaan johon se talletetaan.



Kuva 6.8 Kromatoidin liikkeiden seuraaminen rotan solussa.



Kuva 6.9 Seurannan onnistuminen kuvasarjassa III

Kuten edellisten kuvasarjojen tapauksessa, objektilohko on tässäkin manuaalisesti irrotettu kuvasarjan ensimmäisestä kuvasta ja objektin sijainti annetaan seurannan alussa. Seurattavan objektin liike ei ole laajaa, mutta objekti mutatoi kohtalaisen vahvasti (kuva 6.8).

Kuvassa 6.9 tarkastellaan seurannan onnistumista kuvasarjassa III. Absoluuttiseen neliövirheeseen perustuva menetelmä suoriutuu täydellisesti ilman poikkeamia. Samoin neliölliseen keskivirheeseen perustuva menetelmä. Kuitenkin painotetun neliöllisen keskivirheen menetelmässä seurannan lopussa tulee poikkeamia. Tämä johtuu objektin mutaatiosta. Laplace-suotimeen perustuva menetelmä aloittaa seurannan kohtuullisesti, mutta kadottaa seurattavan objektin viidennessä kuvatasossa objektin muodon muutoksen vuoksi. Kuvasarjasta II poiketen, toinen tilastollisista menetelmistä toimii hyvin. Histogrammiin perustuvan menetelmän tulos on lähes optimaalinen. Tulos on myös varsin mielenkiintoinen, sillä virhettä on hieman vain seurannan alussa, eikä seurannan lopussa,

jossa objektin muoto on muuttunut. Tämä johtunee objektin ja taustan välisestä eroavaisuuksien muutoksesta. Varianssiin perustuva menetelmä kadottaa objektin välittömästi, sitä enää myöhemmin löytämättä.

7. YHTEENVETO JA LOPPUPÄÄTELMÄT

Objektinseurantajärjestelmän käyttämä havaintoaineisto muodostuu peräkkäisistä kuvatasoista, jotka yhdessä muodostavat kuvasarjan. Seuranta tapahtuu tarkastelemalla kuvatasosta saatavaa informaatiota. Objektinseurannan kannalta oleellista on seurattavan objektin liike, joka voi olla translaatiota, rotaatiota tai mutaatiota. Monokulaarisen eli "yksisilmäisen" havaitsijan kannalta kolmiulotteinen liike on mutaatiota, sillä objektin kaksiulotteinen muoto muuttuu objektin liikkessa. Ainoastaan kaksiulotteinen translaatio on tällöin havaitsijan kannalta puhdasta translaatiota. Objektinseuranta-sovelluksissa tärkein parametri on objektin muoto. Jos seurattava objekti on kiinteä, voidaan objekti esittää yhtenäisenä objektilohkona. Vastaavasti muuttuvamuotoisen objektin seurannassa on parempi käyttää objektin ääriviivojen seuraamiseen soveltuvia menetelmiä.

Objektinseurantajärjestelmän pääkomponentit ovat objektin etsintä ja sijainnin ennustus. Objektin etsinnässä objektin sijainti määritetään täsmämällä. Täsmäys tarkoittaa kuva-alueiden vertailuun perustuvan virhefunktion minimointia. Objektin etsinnässä käytetään sijainnin ennustusta avuksi. Liikkeen estimointiin perustuvassa sijainnin ennustuksessa objektin sijainti tarkasteltavassa kuvatasossa arvioidaan objektin aikaisemman liikkeen perusteella. Objektin etsintään voidaan käyttää lohkotäsmäystä. Lohkotäsmäyksessä objekti määritellään suorakaiteen muotoisena kuva-alueena eli objektilohkona, jota vertaillaan kuvatasoon. Lohkotäsmäyksessä täsmäyksen määrittelevä virhefunktio voi perustua pikselitason eroavaisuuksiin tai tilastollisiin eroavaisuuksiin. Tärkeimmät pikselitason virhefunktiot ovat absoluuttinen- ja neliöllinen keskivirhe. Objektilohkon osia voi lisäksi korostaa painomatriisilla. Täsmäys voi perustua myös objektin muotoon, jolloin täsmäys tapahtuu objektilohkon ja kuvatasoon gradienttikenttien avulla. Gradienttikenttä ilmaisee pikselit, joissa kuvan intensiteetti vaihtuu. Tilastollisia täsmäysmenetelmiä ovat

histogrammiin ja varianssiin perustuvat menetelmät. Histogrammi ilmaisee lohkon pikseleiden arvojakauman ja varianssi ilmaisee lohkon pikseleiden keskimääräisen eron lohkon pikseleiden arvojen keskiarvosta.

Toteutettu objektinseurantajärjestelmä perustuu lohkotäsmäykseen. Täsmäysmenetelminä ovat absoluuttinen- ja neliöllinen keskivirhe, painotettu neliöllinen keskivirhe, laplace-suotimen gradientit, histogrammien erotus ja varianssien erotus. Liikkeen estimointi perustuu suoraviivaisen liikkeen tunnistukseen. Määrittelemällä etsintäalue koko ruudun kokoiseksi seuranta oli virheellistä. Tarkkuus parani oleellisesti käyttämällä liikkeen estimoinnin tuottamaa arvioitua sijaintia etsintäalueen määrittelyyn. Myös suoritus aika lyheni. Järjestelmä kykeni seuraamaan muuttuvataustaista reaali maailmasta kuvattua kuvasarjaa laskeutuvasta laskuvarjohyppääjästä. Tämä onnistui käyttämällä täsmäykseen painotettua neliöllistä keskivirhettä ja etsintään liikkeen estimointiin perustuvaa sijainnin ennustusta. Järjestelmä kykeni myös rotan solun osan liikkeiden seurantaan.

Jatkotutkimuksen kohteena voisi olla adaptiivisen objektilohkon soveltaminen toteutettuun järjestelmään. Tämä mahdollistaisi objektin seurannan myös mutatoivan objektin osalta. Adaptiivisen objektilohkon ongelmana on lähinnä taustainformaation ilmestyminen objektilohkoon virheellisten päivitysten vuoksi. Toinen tutkimuskohde voisi olla aktiiviseen reunaan perustuvan järjestelmän toteuttaminen. Tällöin mielenkiintoista olisi mitata miten tarkasti järjestelmä kykenee seuraamaan voimakkaasti muotoaan muuttavaa objektia. Aktiivinen reuna voisi perustua aliobjekteihin, jotka vastaisivat aktiivisen reunan kulmapisteitä. Jo toteutetut lohkotäsmäystekniikat voisivat soveltua lähes sellaisinaan kulmapisteiden seurantaan. Tutkimusongelmana olisi tällöin aliobjektien keskinäisen liikkeen analysointi.

VIITELUETTELO

[ODI93] Berger Marie-Odile : "Tracking Rigid and No Polyhedral Objects in An Image Sequence", France, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp. 945-952, 1993, Tromsø, ISBN: 82-992872-0-0

[BET94] Bethel Roy E., Paras George J. : "A PDF Multitarget Tracker", IEEE Transactions on Aerospace and Electronic Systems, Vol. 30, No. 2, April 1994, ISSN: 0018-9251

[BUL93] Bulaz-Cruz J., Ali A.T., Dagless E.L. : "Real-Time Motion Detection and Tracking", UK, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 515-522, 1993, Tromsø, ISBN: 82-992872-0-0

[JEN93] Carstensen Jens Michael : "Cooccurrence Feature Performance in Texture Classification", Denmark, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp. 831-838, 1993, Tromsø, ISBN: 82-992872-0-0

[CHI94] Chin Leonard : "Application of Neural Networks in Target Tracking Data Fusion", IEEE Transactions on Aerospace and Electronic Systems, Vol. 30, No. 1, pp. 281-287, January 1994, ISSN: 0018-9251

[COH93] Cohen Isaac : "Nonlinear Variational Method for Optical Flow Computation", France, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 523-530, 1993, Tromsø, ISBN: 82-992872-0-0

[FON93] Fontana Franco, Dellepiane Silvana : "Context-Dependent Region Growing by a Label-Driven Method", Italy, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp. 1253-1260, 1993, Tromsø, ISBN: 82-992872-0-0

[FRÄ95] Fränti Pasi : "Image Processing and Compression", Lecture Notes, University of Joensuu, 1995

[FUJ93] Fujimura Kouta, Yokoya Naokazu, Yamamoto Kazuhiko : "Recognition of Echocardiographic Images by Active Contour Models", Japan, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 487-492, 1993, Tromsø, ISBN: 82-992872-0-0

[HAV95] Haven Hannu : "La programmation parallèle en traitement d'image", Projet de maîtrise, Université de Caen, 1995

[HER92] Hertz John, Krogh Anders, Palmer Richard G. : "Introduction to the Theory of Neural Computation", Lecture Notes Volume I, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, 1992, ISBN: 0-201-51560-1

[HOR86] Horn Berthold Klaus Paul : "Robot Vision", The MIT Electrical Engineering and Computer Science Series, The MIT Press, 1986, ISBN: 0-262-08159-8

[PET93] Johansen Peter : "Proposal for a Solution to the Surveillance Problem", Denmark, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 159-166, 1993, Tromsø, ISBN: 82-992872-0-0

[KIL93] Kilger M. : "Heavy Traffic Monitoring in Real-Time", Germany, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp. 1077-1076, 1993, Tromsø, ISBN: 82-992872-0-0

[KOH96] Kohler Ralf, Glazer Frank, Williams Thomas : "A Semi-Automated Film/Video System for Motion Tracking", Amerinex Applied Imaging Inc.

[KOL96] Koller Dieter : "Model-Based Object Tracking in Road Traffic Scenes", Germany, <http://www.ecrc.de/staff/koller/ModelTracking.html>, 1996

[KRI95] Krishnan Srivatsan, Raviv Daniel : "2D Feature Tracking Algorithm for Motion Analysis", Pattern Recognition, Vol. 28, No. 8, pp. 1103-1126, 1995, ISSN: 0031-3203

[KNC93] Kristensen Steen, Henrik Moller Nielsen, Henrik Iskov Christensen : "Cooperative Depth Extraction", Denmark, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 321-328, 1993, Tromsø, ISBN: 82-992872-0-0

[LEE93] Lee Chia-Hoang, Joshi Anupam : "Correspondence Problem in Image Sequence Analysis", Pattern Recognition, Vol. 26, No. 1, pp. 47-61, 1993, ISSN: 0031-3203

[GAL92] LeGall Didier J. : "The MPEG video compression algorithm", Signal Processing, Image Communication 4, pp. 129-140, 1992

[MAK93] Maki Atsuto, Uhlin Thomas, Eklundh Jan-Olof : "Phase-Based Disparity Estimation in Binocular Tracking", Sweden, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol.II, pp. 1145-1152, 1993, Tromsø,ISBN: 82-992872-0-0

[OXL94] Montera Dennis A., Rogers Steven K., Ruck Dennis W., Oxley Mark E. : "Object Tracking Through Adaptive Correlation", Optical Engineering, Vol. 33, No. 1, pp. 294-302, January 1994, ISSN: 0091-3286

[MAT94] Matteucci P., Regazzoni C. S., Foresti G. L. : "Real-time Approach to 3-D Object Tracking in Complex Scenes", Electronics Letters, Vol. 30, No. 6, pp. 475-477, March 1994

[OLS93] Olstad Bjørn, Tysdahl Harald E.: "Improving the Computational Complexity of Active Contour Algorithms", Norway, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 257-264, 1993, Tromsø, ISBN: 82-992872-0-0

[SCH93] Schutte Klamer : "Region Growing With Planar Facets", Netherlands, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp. 719-725, 1993, Tromsø, ISBN: 82-992872-0-0

[SEE96] Seeliger Oliver : "Behavior-Based Space Telerobotics And Vision", University of Maryland, USA, <http://www.cs.umd.edu/users/seeliger/robotics.html>, 1996

[SEF94] Seferidis Vassilis E., Ghanbari Mohammad : "Adaptive Motion Estimation Based on Texture Analysis", IEEE Transactions on Communications, Vol. 42, No. 2/3/4, pp. 1277-1287, 1994, ISSN: 0090-6778

[SRI90] Srinivasan M. V. : "Generalized Gradient Schemes for the Measurement of Two-Dimensional Image Motion", Biological Cybernetics, No. 63, pp. 421-431, 1990

[SWO94] Sworder, D. D., Vojak R. : "Tracking Mobile Vehicles Using a Non-Markovian Maneuver Model", Journal of Guidance, Control and Dynamics, Vol. 17, No. 4: Engineering Notes, pp. 870-872, 1994

[SOR93] Sørensen Erik, Christensen Henrik I : "Monitoring of Road Traffic", Denmark, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp. 1085-1091, 1993, Tromsø, ISBN: 82-992872-0-0

[PNG94] Tsai Ping-Tsing, Shah Mubarak, Keiter Katharine, Kasparis Takis : "Cyclic Motion Detection for Motion Based Recognition", Pattern Recognition, Vol. 27, No. 12, pp. 1591-1603, 1994, ISSN: 0031-3203

[UKA93] Ukai Masato : "Extract the Rail Vanishing Point for Camera View Control", Japan, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. I, pp. 651-658, 1993, Tromsø, ISBN: 82-992872-0-0

[WAN94] Wang T. C., Varshney P. K. : "Measurement Preprocessing Approach for Target Tracking in a Cluttered Environment", IEE Proceedings in Radar and Sonar Navigation, Vol. 141, No. 3, pp. 151-158 June 1994

[WEN93] Weng Juyang, Ahuja Narendra, Huang Thomas S. : "Optimal Motion and Structure Estimation", IEEE Transactions on Pattern Analysis and Machine Intelligence,

Vol. 15, No. 9, pp. 864-884, September 1993, ISSN: 0162-8828

[LAM93] Yuen Kelvin S. Y., Lam Lam T. S., Leung Dennis N. K. : "An Approach for Corner Detection", Hong Kong, SCIA '93, Proceedings of The 8th Scandinavian Conference on Image Analysis, Vol. II, pp.1071-1076, 1993, Tromsø, ISBN: 82-992872-0-0

[ZNG94] Zhang Zhengyou : "Token Tracking in a Cluttered Scene", Image and Vision Computing, Vol. 12, No. 2, pp. 110-120, March 1994, ISSN: 0262-8856

[ZHU94] Zhou B., Bose N. K. : "A Unified Approach to Data Association in Multitarget Tracking", Automatica, Vol. 30, No. 9, pp. 1469-1472, 1994, ISSN: 0005-1098