

# Estimering av spedbarns bevegelser ut fra videodata

**Svein Arne Nasset**

Master i teknisk kybernetikk  
Oppgaven levert: Juni 2008  
Hovedveileder: Geir Mathisen, ITK  
Biveileder(e): Øyvind Stavadahl, ITK



# Oppgavetekst

Det foreligger videoopptak av ca. 100 spedbarns spontane bevegelser. For å kunne stille en tidligst mulig diagnose i forhold til nevro-motoriske sykdommer, ønsker en å analysere barnas bevegelser og klassifisere disse som hhv. "normale" eller "avvikende".

I denne oppgaven skal du finne fram til algoritmer som kan følge utvalgte kroppsdelers bevegelser i videobildene, slik at disse bevegelsene senere kan analyseres.

1. Gjør en utredning om eksisterende algoritmer for bevegelsesfølging i videodata.
2. Konstruer eller tilpass en eller flere algoritmer til det foreliggende datasettet for å følge hender, føtter, hode og bryst/kropp på en mest mulig robust måte. Valg av algoritme skal grunngis ut fra den konkrete anvendelsen.
3. Implementer og prøv ut algoritmen(e) på det foreliggende datasettet.
4. Så langt tiden strekker til, tallfest algoritmen(e)s robusthet i forhold til varierende lysforhold, motivets farge og tekstur, oppløsning, beregningstid m.v.

Besvarelsen skal bestå av en rapport og en DVD-plate med programkode etc.

Oppgaven gitt: 07. januar 2008

Hovedveileder: Geir Mathisen, ITK



## Forord

Denne rapporten er resultatet av masteroppgaven min ved utdannelsen for master i teknisk kybernetikk ved NTNU i Trondheim.

Rapporten inneholder en beskrivelse av ulike metoder for å representere et bildeutsnitt, og hvordan disse metodene kan brukes i en algoritme for målfølging i bildesekvenser. Algoritmene er tilpasset for å kunne følge et spedbarns bevegelser av hode, føtter, hender og bryst. Dette blir gjort for å kunne detektere om barnet har utviklet Cerebral Parese.

Jeg synes denne masteroppgaven har vært artig å jobbe med. Ikke bare byr den på interessante faglige og tekniske utfordringer, men også det menneskelige aspektet ved oppgaven gjør at den har vært spennende.

Jeg vil rette en stor takk til min veileder Øyvind Stavadahl (1. amanuensis, NTNU) for inspirerende og hyggelige møter, og til Ole Christian Eidheim (PhD-student, NTNU) for ideer og diskusjoner. Også takk til Marion for hennes kjærighet og oppmuntrende smil.

Trondheim, mai 2008  
Svein Arne Nettet



## Sammendrag

For å kunne gi best mulig behandling av nevromotoriske sykdommer hos barn, er det viktig at diagnosen stilles så tidlig som mulig. Derfor pågår det et forskningsprosjekt der man studerer barns sporadiske bevegelser når de er to til fire måneder gamle. Disse bevegelsene har vist seg å kunne gi en indikasjon på om barnet er normalt eller avvikende.

Det foreligger en database med videoopptak av barn som ligger på ryggen og kan bevege seg uten hindringer. Ved å studere disse opptakene kan en spesialist innen fysioterapi kunne gi en diagnose på Cerebral Parese med relativt god sikkerhet. Målet med forskningsprosjektet er derfor å erstatte fysioterapeuten med et dataprogram som kan analysere videostrømmen på samme måte.

Denne masteroppgaven går ut på å kunne posisjonere hender, føtter, hode og bryst hos spedbarn i videosekvenser. Dette skal gjøres uten bruk av optiske markører festet på barnet. Posisjonsestimatene kan i videre arbeid brukes til å gjøre en analyse av barnets bevegelser, og kunne klassifisere de som normale eller avvikende.

Det er utviklet og testet fire forskjellige måter å utføre målfølgingen på. Disse baserer seg på histogram, spatiogram, co-occurrence matriser og cosinusvinkelen mellom bildeutsnitt.

Det er gjort tester som har resultert i vellykkede følginger av barnets hode, hender, føtter og bryst. De forskjellige representasjonsmetodene har vist seg å ha styrker og svakheter som gjør at de kan kombineres til å gi en mer robust algoritme. Direkte billedsammenligning med cosinusvinkel er best når objektet som skal følges ikke endrer form og farge i vesentlig stor grad. Dette gjelder for hodet og brystet. Spatiogram-metoden og co-occurrence metoden er gode når objektet varierer form og farge, og i kombinasjon med flere referansemoteller har de vist seg å danne et godt grunnlag for robust følging av hender og føtter.

For å gjøre algoritmen mer robust er det foreslått alternative metoder som kan utvikles i framtiden. Ved å registrere kanter og konturer er det sannsynlig at følgingen vil fungere bedre.

# Innhold

<b>Sammendrag</b> .....	<b>i</b>
<b>Innhold</b> .....	<b>ii</b>
<b>Figurer og tabeller</b> .....	<b>iv</b>
<b>1 Introduksjon</b> .....	<b>1</b>
1.1 Cerebral Parese.....	1
1.2 GMA - General Movements Assesement.....	1
1.3 Prosjektets forhistorie.....	2
1.4 Oppgavebeskrivelse .....	2
1.5 Generell målfølgning i bildesekvenser.....	3
1.6 Rapportens oppbygning.....	6
<b>2 Metoder</b> .....	<b>7</b>
2.1 Pre-prosessering av videostrømmen.....	7
2.1.1 Kontrast og lysstyrke .....	8
2.1.2 De-interlacing .....	8
2.2 Histogram .....	10
2.3 Spatiogram .....	12
2.4 Co-occurrence matriser .....	13
2.5 Vinkelmetoden .....	14
2.6 Algoritme for målfølgning .....	14
2.6.1 Basert på histogrammer .....	16
2.6.2 Basert på spatiogram .....	16
2.6.3 Basert på co-occurrence matriser.....	17
2.6.4 Basert på vinkelmetoden.....	18
2.6.5 Valg av antall farger .....	18
2.6.6 Målfølgning med modelloppdatering .....	21
2.6.7 Målfølgning med flere referansemodeller .....	27
2.7 Sammenligning av metodene .....	28
2.8 Fjerning av høyfrekvent støy.....	32
2.9 Applikasjon for analyse av metodene .....	33



<b>3</b>	<b>Resultat og analyse .....</b>	<b>37</b>
3.1	Kjøretidsanalyse .....	37
3.2	Støyfjerning .....	38
3.3	Følging av hender .....	39
3.3.1	Metodene hver for seg .....	39
3.3.2	Metodene sammen .....	41
3.4	Følging av føtter .....	43
3.5	Følging av hode .....	45
3.6	Følging av bryst .....	47
<b>4</b>	<b>Konklusjon, diskusjon og videre arbeid .....</b>	<b>50</b>
4.1	Resultater .....	50
4.2	Diskusjon .....	50
4.3	Forslag til forbedringer .....	51
4.3.1	Datagrunnlag .....	51
4.3.2	Nye metoder .....	51
	<b>Bibliografi .....</b>	<b>54</b>
	<b>Appendiks A Grafer fra følging av hender .....</b>	<b>57</b>
	<b>Appendiks B Grafer fra følging av føtter .....</b>	<b>63</b>
	<b>Appendiks C Grafer fra følging av hode .....</b>	<b>65</b>
	<b>Appendiks D Grafer fra følging av bryst .....</b>	<b>67</b>
	<b>Appendiks E Plott av co-occurrence modelloppdatering .....</b>	<b>69</b>
	<b>Appendiks F Programkode og videoer .....</b>	<b>71</b>

## Figurer og tabeller

Figur 1: Barn med sensorer tilkoblet.....	2
Figur 2: Eksempel på bruk av histogramutjevning .....	8
Figur 3: Interlacing i datagrunnlaget.....	9
Figur 4: Gråtoneskala delt i 8 grupperinger .....	10
Figur 5: RGB fargerom delt i 2x2x2 .....	10
Figur 6: Gråtone testbilde.....	11
Figur 7: Histogram, 256 grupperinger.....	11
Figur 8: Histogram, 8 grupperinger.....	11
Figur 9: Spatiogram illustrasjonsbilde .....	12
Figur 10: Co-occurrence illustrasjonsfigur .....	13
Figur 11: Utsnitt for beregning av co-occurrence matrise .....	13
Figur 12: Co-occurrence matrise fra området i figur 11 .....	13
Figur 13: Co-occurrence matrise diagonalt snitt.....	14
Figur 14: Søkeområdet for målfølgingsalgoritmen.....	15
Figur 15: Illustrering av søkeområdet.....	15
Figur 16: Filtreringsmatriser som funksjon av $\sigma$ .....	17
Figur 17: Effektiv utregning av nabo co-occurrence matriser .....	18
Figur 18: Grafer for målfølgning ved forskjellig valg av antall farger .....	20
Figur 19: Co-occurrence modelloppdatering for hvert bilde .....	21
Figur 20: Modelloppdatering av co-occurrence metoden, kun ved gode treff.....	22
Figur 21: Vinkelmetoden med modelloppdatering .....	23
Figur 22: Prinsipp for å oppdatere et posisjonsestimert .....	24
Figur 23: Posisjonsestimert ved bruk av modelloppdatering .....	24
Figur 24: Posisjonskandidater fra de forskjellige metodene .....	25
Figur 25: Optimalt valg av posisjonskandidater .....	26
Figur 26: Tidsserie ved bruk av optimalt valg av posisjonskandidater.....	26
Figur 27: Justering av posisjon etter følgning av feil fot .....	27
Figur 28: Referanseobjekter ved bruk av målfølgning med flere referansemodeller .....	28
Figur 29: Bilder som har likt histogram .....	29
Figur 30: Bilder som har innbyrdes lik vinkel .....	30
Figur 31: Vinkelmetoden mister følgning av armen .....	30
Figur 32: Spektrum til barnets virkelige bevegelser .....	32
Figur 33: Welch-spektrum .....	32
Figur 34: Yule-Walker-spektrum .....	32
Figur 35: Karakteristikk for et Butterworth lavpassfilter.....	33

Figur 36: Input og output til følgingsalgoritmen.....	34
Figur 37: Skjerm bilde av applikasjonen.....	34
Figur 38: ULM diagram for algoritmen .....	36
Figur 39: Oversikt over hvordan funksjonene er relatert til hverandre.....	36
Figur 40: Spektrum av filtrert posisjonsestimater .....	38
Figur 41: Tidsplot av posisjonsestimater med og uten filter .....	38
Figur 42: Metodenes posisjonsestimater ved bilde nr 197 .....	40
Figur 43: Tidsserie høyre hånd, metodene hver for seg.....	41
Figur 44: Posisjon i y-retning forfølging av venstre hånd .....	42
Figur 45: Korrelasjon i y-posisjon for venstre hånd .....	42
Figur 46: Tidsserie forfølging av venstre fot .....	43
Figur 47: Oversikt over hvor ofte de forskjellige referansemodellene er brukt.....	44
Figur 48: Følging av venstre fot, avvik fra virkelig posisjon.....	45
Figur 49: Forskjellen på hode- og ansiktsfølging.....	46
Figur 50: Fortløpende avvik fra virkelig posisjon forfølging av ansikt .....	46
Figur 51: Søkeområdet og posisjonsverdier forfølging av bryst.....	48
Figur 52: Okklusjon vedfølging av bryst .....	48
Figur 53: Resultat for bevegelsesdeteksjon.....	52
Figur 54: Kantdeteksjon brukes for å finne venstre hånd .....	52
Figur 55: Rød-grønn fargefordeling i hele bildet (a) og i områder med hud (b).....	53
Figur 56: Resultat av huddeteksjon.....	53
Tabell 1: Oppsummering av resultatene for co-occurrence ved forskjellige valg av $\sigma$ .....	17
Tabell 2: Oppsummering av valg av antall farger.....	19
Tabell 3: Oppsummering av modelloppdatering for co-occurrence metoden.....	22
Tabell 4: Fordeler og svakheter ved de forskjellige metodene .....	31
Tabell 5: Beskrivelse av komponentene i applikasjonen .....	35
Tabell 6: Kjøretidsanalyse, ms .....	37
Tabell 7: Oppsummering av resultatene forfølging av hender.....	41
Tabell 8: Oppsummering målfølging med og uten posisjonskorrigering .....	43
Tabell 9: Gjennomsnittlig avvik fra virkelig posisjon for venstre fot.....	44
Tabell 10: Oppsummering av resultater forfølging av føtter .....	44
Tabell 11: Oppsummering av ansiktsfølging .....	47
Tabell 12: Oppsummering av resultatene forfølging av bryst.....	49

# 1 Introduksjon

## 1.1 Cerebral Parese

Cerebral Parese (CP) er den vanligste nevrologiske årsaken til bevegelseshemming hos barn og unge (Norsk Barnelegeforening, 2008). Sykdommen skyldes en utviklingsforstyrrelse eller en skade i hjernen som har oppstått i fosterlivet, under fødselen eller i løpet av de første leveårene. Det mest karakteristiske symptomet er motoriske vansker i varierende grad, men andre tegn på hjerneskade som for eksempel lærevansker, forsinket tale og språkutvikling og epilepsi kan også være til stede.

I følge Norsk Barnelegeforening viser undersøkelser at omkring 2 per 1000 nyfødte utvikler CP. Det betyr ca. 120 barn per årskull i Norge. Spesielt barn som er født før termin er utsatt for å utvikle CP (Med-World, 2002).

Diagnostikk av CP er primært en klinisk oppgave. Bekreftelse av endelig diagnose baserer seg ofte på flere observasjoner og undersøkelser over tid. De fleste barn får stilt diagnosen før ett års alder, men ved uklare symptomer kan det ta noe lenger tid.

Behandling av CP er en tverrfaglig oppgave og det er viktig å starte opp så snart mistanken foreligger.

## 1.2 GMA - General Movements Assesement

Når barn er to til fire måneder gamle utvikler de bevegelser som er spontane, men karakteristiske (fidgeties). Friske barn utvikler fidgeties etter 9-15 uker. Disse bevegelsene er små, sirkulære og elegante bevegelser av hode, bryst og lemmer, og forekommer når barnet er våken og ligger på ryggen uforstyrret (Prechtl, 1997). Bevegelsene er varierte, men følger et fast mønster. Forstyrrelser i dette mønsteret, og ensformige bevegelser, er derimot en indikasjon på at barnet kan ha CP. General Movements Assesement (GMA) er en klinisk metode for undersøke om barn er i faresonen for å ha CP (Prechtl, 2001). Metodene baserer seg på å observere barnets bevegelser, og ut i fra dette kunne gi en subjektiv diagnose.

Siden GMA analysen kun gir en subjektiv betraktning av barnets bevegelser, er det knyttet usikkerhet til hvor nøyaktig diagnosen kan stilles.

### 1.3 Prosjektets forhistorie

Lars Adde ved Seksjon for Fysioterapi på St. Olavs Hospital i Trondheim startet prosjektet på slutten av 90 tallet. Han ville filme og måle bevegelsene hos spedbarn for å si noe om sannsynligheten for CP. Øyvind Stavdahl (SINTEF/NTNU) bidro med instrumentering, og i 2000 var det første laboratoriet klart. Systemet ble videre utviklet, og i 2003 hadde man et operativt system. Dette bestod av et kamera og et sensorsystem for registrering av barnets bevegelser. Sensorene er basert på magnetisme, og gir posisjon i tre dimensjoner 25 ganger i sekundet. Disse sensorene ble festet på barnets hender, føtter, hode og bryst.



**Figur 1: Barn med sensorer tilkoblet**

I tillegg til en klinisk undersøkelse av barnets bevegelser, er det gjort forsøk på å basere diagnostikken på posisjonsdata fra sensorene. Dette arbeidet pågår fremdeles.

Som et alternativ til de magnetiske sensorene, er det forsøkt å finne posisjonsdata direkte på grunnlag av videofilmen. Staal (2006) bruker fargemarkører festet til barnets hender, føtter, hode og bryst for å finne posisjonene.

### 1.4 Oppgavebeskrivelse

Det er tilgjengelig en database med ca 100 videoopptak av spedbarn, som vist i figur 1. Dette er konfidensielt materiale, og filmene er derfor kun tilgjengelig på en PC uten nettverkstilknytning. I tillegg er all data kryptert, slik at det kun er masterstudenten som har tilgang til filene. Dette er gjort av sikkerhetshensyn, i tilfelle utstyret skulle bli stjålet fra kontoret.

Sensorene som er beskrevet i kapittel 1.3 baserer seg på magnetisme, og gir posisjon i tre dimensjoner. Svakheter ved dette systemet er at det er kostbart, det krever tid å montere det på barnet og det består av mange ledninger og elektronikk. Det er gjort videoopptak av barna med dette sensorsystemet tilkoblet. Ungene ligger på ryggen og kan bevege seg fritt mens de blir filmet ovenfra. Det er derfor interessant å utvikle et system som kan følge barnets bevegelser ut i fra videofilmen. Oppgaven er dermed å utvikle algoritmer som kan følge bevegelsene til barnets hender, føtter, hode og bryst.

## 1.5 Generell målfølgning i bildesekvenser

Det å kunne følge objekter i en videosekvens er en viktig vitenskap innenfor mange forskjellige bruksområder.

- Overvåkningssystemer for trafikk, bygninger og utearealer (Siebel, 2001) (Siebel, 2003) (Ohya et al., 2002)
- Sensorløsninger for industriautomatisering (Lange, 2003)
- Målfølgning for militære formål (Haker et al., 2001), (Hammon, 2002)
- Følgning av personer for bruk i videokonferanser (Fung et al., 2001)
- Følgning av idrettsutøvere som grunnlag for statistikk (for eksempel hvor langt har en fotballspiller løpt i en kamp) (Choi et al., 1996) og (Iwase og Saito, 2003)

Et bevegelsesfølgningssystem skal kunne gi posisjonen til et eller flere bestemte objekt i en strøm av bilder. Utfordringen er å kunne følge objektene selv om de beveger seg raskt i forhold til bilderate, endrer form på grunn av rotasjon, endrer fargesammensetning på grunn av lysforhold eller når det delvis blir gjemt bak andre objekter.

Det er vanlig å gjøre noe pre-prosessering av bildematerialet før objektets posisjon blir bestemt. Man kan endre kontrast- og lysverdier slik at klassifiseringen av objektet blir enklere. For å representere et objekt kan det benyttes forskjellige metoder, og man må velge en metode som passer best til sitt formål.

For å klassifisere bilder brukes forskjellige teknikker (Kyuheon, 1999);

- fargeinformasjon brukes for å bestemme hvilke farger som er dominante innenfor en gitt region
- teksturanalyse blir gjort for å bestemme bildets karakteristiske struktur
- kantanalyse for å finne objektets form

## **Segmentering**

Rasmussen (1996) viser en segmenteringsmetode, der man ved bruk av filtrering og segmentering av et bilde vil sitte igjen med det området man ønsker å følge ("blob"). Her brukes en metode der man segmenterer bildet på bakgrunn av fargen til det området som man ønsker å følge. Denne metoden gir gode resultater hvis objektets farge skiller seg tydelig ut fra bakgrunnenes farger. Du (1995) bruker også denne teknikken, men her blir bildet gjort om fra RGB – fargerom til et endimensjonalt fargerom med de mest brukte distinkte fargene, og noen gråtoner. Til sammen utgjør dette 52 forskjellige fargeverdier. Dette reduserer beregningstiden betraktelig, noe som førte til at metoden kunne benyttes på datidens roboter.

Segmentering på grunnlag av farger har blitt brukt i tidligere utgaver av dette prosjektet. (Staal, 2006). Her har ungene blitt utstyrt med fargebånd på hender, føtter, hode og bryst, og en segmenteringsalgoritme har funnet posisjonene ved hjelp av disse fargemarkørene.

## **Histogrammer**

Histogram er en vanlig metode å benytte hvis man skal bruke farger som sammenligningsgrunnlag mellom to bilder. Zivkovic (2004) bruker et histogram for å representere et objekt. Objektet avgrenses med en ellipse, og piksler nærme sentrum blir gitt en høyere vekt i histogrammet, enn piksler langs randen av objektet. På denne måten viser Zivkovic en representasjonsmetode som tar hensyn til at objektet kan endre form og størrelse. Comanici (2003) bruker en tilsvarende metode sammen med en mean-shift algoritme for å oppnå god ansiktsfølging. Zhao (2007) baserer seg også på bruk av histogrammer, men her blir histogrammene utvidet til å inneholde data fra bildet sett i mange fargerom. Birchfield (1998) bruker histogrammer sammen med kantinformasjon for å følge menneskehoder i en bildesekvens.

## **Spatiogrammer**

Birchfield (2005) viderefører teknikken med histogrammer til å inneholde mer informasjon. Det han kaller et spatiogram er et andre ordens histogram, som i tillegg til å finne mengden av de forskjellige fargene også finner fargenes massesenter i bildet, samt fargenes varians i posisjon. På den måten viser han at spatiogrammet er en bedre måte å representere et objekt på, enn ved å kun benytte et histogram. Metoden lagrer mer informasjon om hvordan objektet ser ut, og man får et bedre sammenligningsgrunnlag mellom objektmodellen og de forskjellige objekt kandidatene. I 2007 testet Birchfield metoden opp mot andre følgingsalgoritmer basert på fargehistogrammer, log-Gabor histogrammer, Haar wavelet histogrammer, og kantorieringshistogrammer. Konklusjonen her er at spatiogram-metoden er bedre enn alle de andre metodene. Conaire (2007a) viderefører spatiogram-metoden, og innfører en bedre måte å sammenligne to spatiogram på, der sammenligningsfunksjonen gir en klarere indikasjon på et godt treff. Videre brukes denne metoden i parallell for å skape en robust algoritme basert på flere enkeltstående spatiogramalgoritmer (Conaire, 2007b).

### **Co-occurrence matriser**

En co-occurrence matrise er en god måte å beskrive teksturen i et bilde (Borchani, 1997). En GLCM (Grey Level Co-occurrence Matrix) beskriver hvor ofte kombinasjoner av fargeverdier forekommer som to nabopiksler. Kyuheon (1999) bruker en slik matrise til å regne ut forskjellige egenskaper for et bildeutsnitt; *maximum probability, uniformity, contrast, correlation, variance, sum of squares, inverse difference moment*. Disse egenskapene blir satt sammen til en egenskapsvektor, og kan videre brukes som sammenligningsgrunnlag for to bildeutsnitt. Hseu (1999) bruker co-occurrence matrisene direkte som sammenligningsgrunnlag mellom to bilder. Her beregnes korrelasjonen mellom bildepikslene og korrelasjonen mellom elementene i co-occurrence matrisen for å finne det utsnittet som er mest likt et referanseutsnitt. Hammon (2002) kombinerer co-occurrence matriser med bakgrunnseliminering for å forbedre ytelsen til målfølgingsalgoritmen. Xu (2004) bruker co-occurrence matrisens egenskaper sammen med et kalmanfilter for å oppnå gode følgingsegenskaper. I den implementasjonen benyttes kun entropien i co-occurrencematriksen som sammenligningsgrunnlag mellom to påfølgende bilder i en videosekvens.

### **Konturer og kanter**

Et bilde i videostrømmen blir analysert for å finne konturer og kanter. Dette er områder der fargeintensiteten endres raskt over en kort pikselavstand i bildet, noe som oppstår i overganger mellom et objekt og en bakgrunn. Objektets kanter definerer en kontur, og det er denne konturen man forsøker å finne i de neste bildene. Prewer et al. (1997) bruker denne metoden til å utvikle en algoritme for å følge objekter i en videosekvens. Zhu et al. (2006) benytter kantdeteksjon sammen med en bevegelsesmodell for å estimere posisjonen til objektet.

### **Optical flow**

Lucena et al. (2003) bruker optical flow sammen med et partikkelfilter i en algoritme for målfølgning. Denne teknikken gjør at man kan følge en piksels bevegelse gjennom en sekvens med bilder. Optical flow kan ikke brukes alene på grunn av manglende nøyaktighet, men metoden kan kombineres med observasjonsmodeller og korrelasjonsanalyse for å bedre følgingsresultatene.

### **SMOG**

Wang et al. (2006a, 2006b, 2007) konstruerer et sammenligningsmål basert på Spatial-color Mixture of Gaussians (SMOG). Dette er en utvidelse av histogrammet, og inneholder i tillegg informasjon om fargenes posisjoner i bildet. På denne måten er metoden ganske tilsvarende spatiogrammet. Men der spatiogrammet kalkulerer statistiske egenskaper for hver farge, vil SMOG algoritmen kun kalkulere statistiske egenskaper for hver parameter i modellen.



## 1.6 Rapportens oppbygning

Denne rapporten er delt i fire hovedseksjoner.

Del 1 inneholder en introduksjon til oppgaven og hva som har blitt gjort i prosjektet tidligere. Det gis også en generell innføring i forskjellige metoder for målsøking i bildesekvenser.

Del 2 inneholder en mer detaljert beskrivelse av de metodene som er benyttet i dette prosjektet. Det gis også en forklaring på hvordan disse metodene kan brukes i en algoritme for målføling gjennom en sekvens med bilder. Applikasjonen som er utviklet blir overfladisk beskrevet, og det presenteres litt enkel signalbehandling av posisjonsestimatene.

Del 3 presenterer de resultatene som er oppnådd, mens del 4 konkluderer og diskuterer, samt kommer med forslag til forbedringer og videre arbeid.

## 2 Metoder

Dette kapitlet beskriver de metodene som er brukt, og hvordan disse er satt sammen til en algoritme for målfølgning. Kapitlet inneholder også litt om hva som er gjort av pre-prosessering av datagrunnlaget.

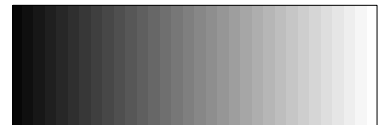
### 2.1 Pre-prosessering av videostrømmen

Et bilde er representert som en 2D flate med mange punkter. Disse kalles piksler, og representerer fargen i dette punktet i bildet. Det er vanlig å indeksere et bilde med x- og y-koordinater, der origo er øverst i venstre hjørne.

Fargene i et bilde kan være representert på mange måter, men de vanligste er:

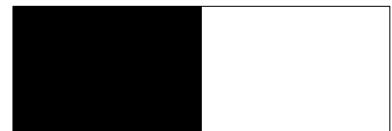
#### Gråtonebilde

Hver piksel har en verdi der minste verdi indikerer svart og høyeste verdi tilsvarer hvit, og mellom dette er det en jevn overgang.



#### Binærbilde

Hver piksel har verdien 0 (svart) eller 1 (hvit). Det er vanlig å bruke denne representasjonen til segmentering av bilder, og målobjektet blir en "blob" (1), og resten er svart.



#### Fargebilde

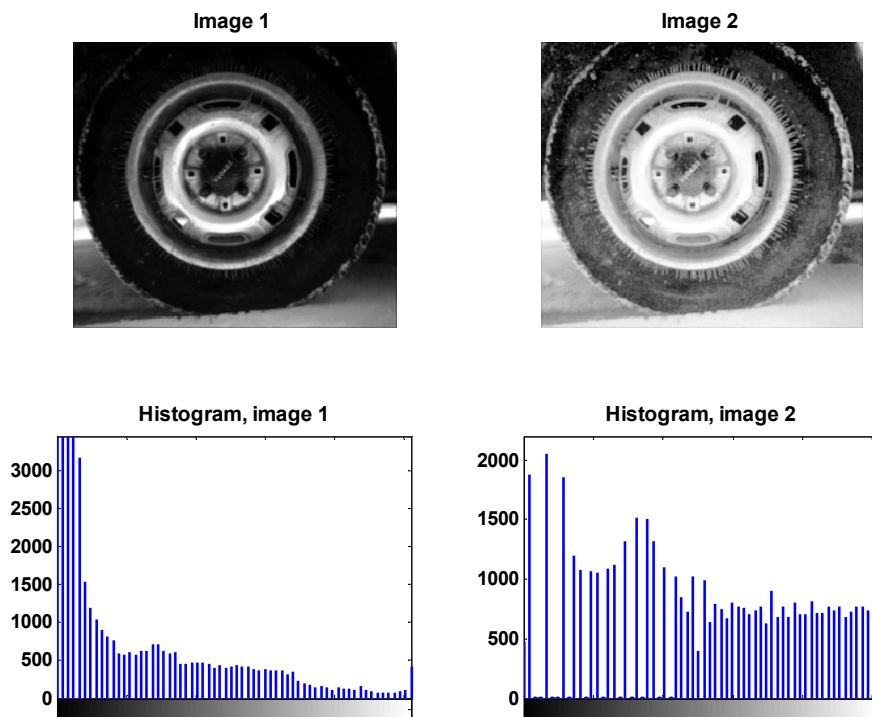
Hver piksel består av en vektor som representerer forskjellige elementærfarger. Ved å kombinere verdier for disse forskjellige fargene kan pikselen få en farge som består av en vektet blanding av elementærfargene. Pikselen får dermed en farge ut i fra et gitt fargerom. Dette kan være RGB (Red,Green,Blue), HSL (Hue, Saturation, Lightness) CMYK (Cyan, Magneta, Yellow, Key (svart)).



Et 8 bit RGB bilde har dermed  $2^8 \cdot 2^8 \cdot 2^8 \approx 16,7$  millioner forskjellige farger.

### 2.1.1 Kontrast og lysstyrke

For å bedre kontrasten i bildene, kan det benyttes en metode for å jevne ut histogrammet til bildet. Dette går ut på å endre fargefordelingen i bildet slik at histogrammet for bildet blir så flatt som mulig. Dermed vil man lettere kunne skille mellom mørke og lyse flater. Figur 2 viser hvordan histogramutjevning er brukt for å øke kontrasten i bildet.



Figur 2: Eksempel på bruk av histogramutjevning

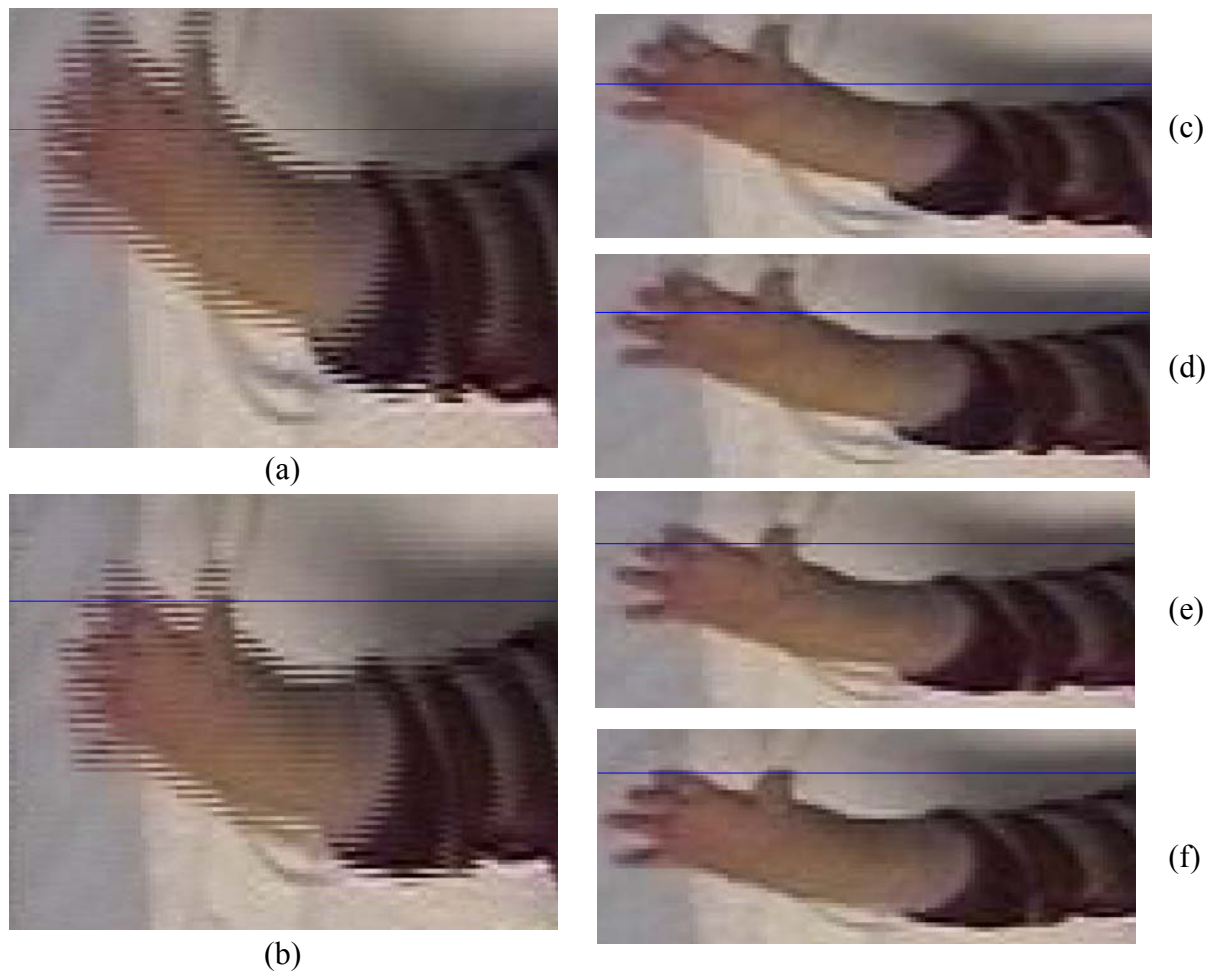
MATLAB har en innebygget funksjon for histogramutjevning.

Histogramutjevning blir benyttet vedfølging av barnets hender.

### 2.1.2 De-interlacing

Da filmene ble tatt opp, ble det bestemt at de skulle digitaliseres med Motion JPEG og en kompresjonsverdi på 85 %. Dette medfører at man ikke lenger har tilgang til kompresjonsfrie stillbilder, men bilder i JPEG format med 85 % komprimering. I tillegg er filmene tatt opp med *interlacing* istedenfor *progressive scan*. Interlacing er en teknikk som brukes i videosystemer for å doble bilderaten uten å doble båndbredden. Det gjøres med at kun annenhver linje i bildet blir tegnet på skjermen av gangen. Det vil si at et komplett bilde består av partallslinjene fra et bilde tatt ved tidspunkt  $t$  og oddetallslinjene fra et annet bilde tatt ved tidspunkt  $t + \Delta t$ . Dette er vanskelig å oppdage når en filmsekvens vises på TV, men

når bildene blir digitalisert oppstår det synlige forvrengninger i bildet. I områder med bevegelse kan man se at bildet består av to delbiler. Se figur 3.



**Figur 3: Interlacing i datagrunnlaget**

I venstre kolonne i figur 3 vises to stillbilder (a) og (b), og man kan se at bildene har forvrengninger siden de består av to delbilder som er fordelt på annenhver linje. I høyre kolonne er delbildene vist hver for seg. Figuren viser at to stillbilder i serie består av fire bilder tatt på forskjellige tidspunkt. Den horisontale streken står på fast posisjon i alle bildene, og man kan se at hånden har beveget seg mellom alle bildene.

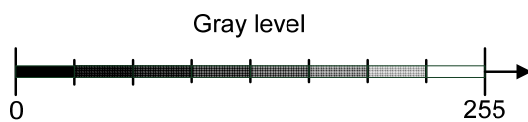
Det finnes forskjellige metoder for å løse problemet med interlacing [44]

- Interpolere  
Det regnes ut et gjennomsnitt for de to delbildene. Resultatet vil ha full oppløsning, men vil bli uskarpt der det har vært bevegelse.
- Linjedobling  
Man bruker kun oddetallslinjene, og gir disse dobbel bredde ved å kopiere de over i partallslinjene.
- Halvere  
Oddetallslinjene og partallslinjene trekkes ut til to separate bilder. På denne måten halveres oppløsningen, mens bilderaten dobles.

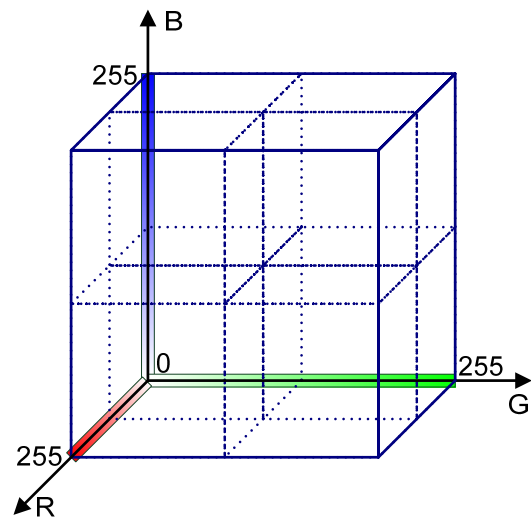
I dette prosjektet vil begge delbildene benyttes for å utnytte all informasjonen i bildene, og oppnå dobbel bilderate.

## 2.2 Histogram

Et histogram er en framstilling av fordelingen av fargene i et bilde, der hver søyle i histogrammet viser hvor mange piksler som har den gitte fargen. Hvis en piksel i et gråtonebilde er representert ved 8 bit, vil pikselen kunne ha 256 forskjellige fargenivåer. Et fargebilde i 8 bit RGB vil derfor kunne inneholde  $256^3 \approx 16,7$  millioner forskjellige farger. For å redusere dimensjonen på histogrammer kan man dele fargeverdiene inn i forskjellige grupperinger, og telle antall forekomster i hver av disse. Hvis man velger å ha åtte grupperinger i et gråtonebilde, vil hver bås inneholde fordelingen fra 32 forskjellige fargeverdier. En slik inndeling er vist i figur 4. I RGB fargebilder vil antall mulige farger bli lik tre ganger antall grupperinger for hver av elementærfargene. Figur 5 viser et RGB fargerom som er delt inn i åtte forskjellige farger.



Figur 4: Gråtoneskala delt i 8 grupperinger



Figur 5: RGB fargerom delt i 2x2x2

Et histogram er normalisert dersom alle elementene summerer seg til 1. Et slikt histogram beregnes på følgende måte:

$$h = [n_1 \quad n_2 \quad \cdots \quad n_{b-1} \quad n_b] \quad (1)$$

$$n_b = \frac{1}{N} \sum_{i=1}^N \delta_{ib} \quad (2)$$

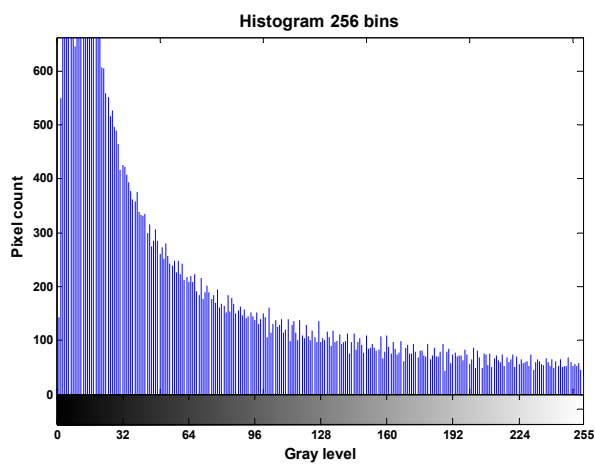
der  $N$  er antall piksler i bildet, og  $\delta_{ib} = 1$  dersom fargeverdien tilhører gruppering  $b$ , og  $\delta_{ib} = 0$  ellers.

Histogrammet er rotasjonsuavhengig, det vil si at metoden ikke tar hensyn til posisjonen til fargene i bildet.

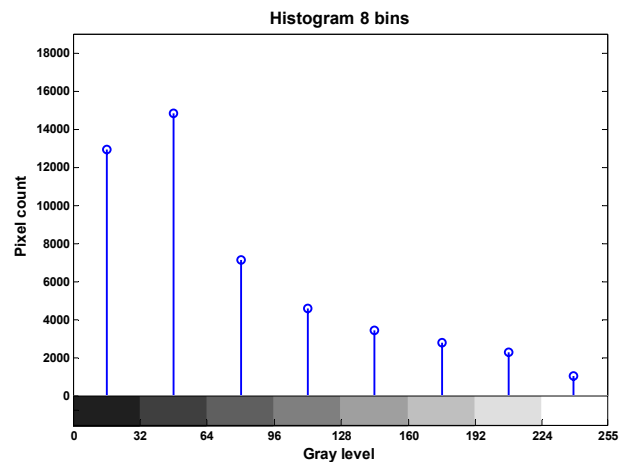


**Figur 6: Gråtone testbilde**

Figur 6 viser et gråtonebilde. Histogrammet for dette er vist i figur 7, og figur 8 viser histogrammet når man deler bildet opp i 8 fager.



**Figur 7: Histogram, 256 grupperinger**



**Figur 8: Histogram, 8 grupperinger**

## 2.3 Spatiogram

Et spatiogram er et histogram som også inneholder informasjon om fargenes plassering i bildet. I tillegg til å telle hvor mange ganger en farge er representert, beregnes også gjennomsnittsposisjonen  $\mu_b$  og variansen  $\Sigma_b$  til de forskjellige fargeverdiene. Dermed benytter metoden mer statistikk enn et histogram. I motsetning til histogrammet er spatiogrammet rotasjonsavhengig.

Et spatiogram beregnes på følgende måte:

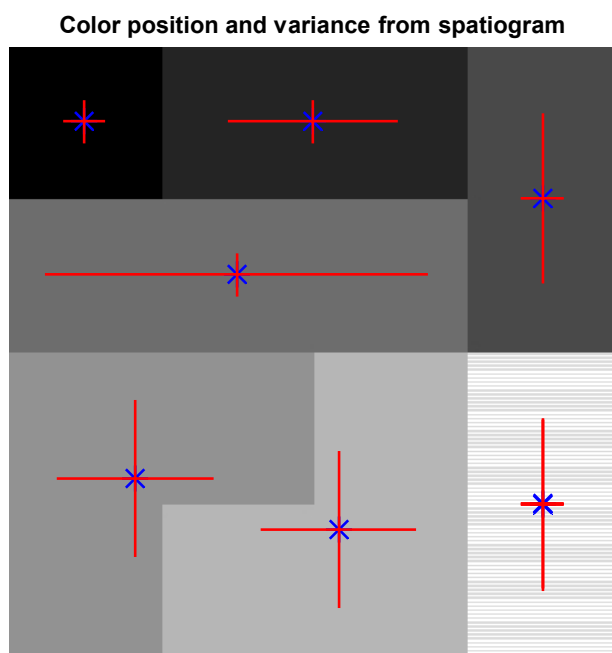
$$s = \left[ \begin{bmatrix} n_1 & \mu_1 & \Sigma_1 \end{bmatrix} \cdots \begin{bmatrix} n_b & \mu_b & \Sigma_b \end{bmatrix} \right] \quad (3)$$

$$n_b = \frac{1}{N} \sum_{i=1}^N \delta_{ib} \quad (4)$$

$$\mu_b = \frac{1}{\sum_{i=1}^N \delta_{ib}} \sum_{i=1}^N x_i \delta_{ib} \quad (5)$$

$$\Sigma_b = \frac{1}{\sum_{i=1}^N \delta_{ib}} \sum_{i=1}^N (x_i - \mu_b)(x_i - \mu_b)^T \delta_{ib} \quad (6)$$

Figur 9 viser hvordan spatiogrammet lagrer massesentrene til fargene. Bildet består av åtte fargenivåer, og gjennomsnittsposisjonen for hver farge er vist i figuren. Variansen til fordelingene av de forskjellige fargene i x og y retning er også vist på figuren. Histogramdelen av spatiogrammet er beskrevet i kapittel 2.2.



**Figur 9: Spatiogram illustrasjonsbilde. Figuren viser massesenteret og variansen til fargene**

## 2.4 Co-occurrence matriser

En co-occurrence matrise  $M$  er en tabell som viser hvor ofte forskjellige kombinasjoner av pikselverdier opptrer som nabopikslar (vertikalt, horisontalt eller på skrå) i et bilde. For eksempel vil  $M_{15,25}$  være lik antall ganger pikslar med fargeverdi 15 og 25 opptrer som nabopikslar. Merk også at  $M$  er symmetrisk siden  $M_{i,j} = M_{j,i}$ . Matrisen har dimensjon lik antall fargenivåer i bildet. Hvis bildet skaleres ned i antall farger (beskrevet i kapittel 2.2), vil co-occurrence matrisen få tilsvarende dimensjon.

$$I = \begin{bmatrix} 2 & 4 & 2 \\ 2 & 1 & 3 \\ 4 & 2 & 3 \end{bmatrix} \quad M = \begin{array}{c} \begin{matrix} & 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{bmatrix} 0 & 4 & 2 & 2 \\ 4 & 2 & 3 & 5 \\ 2 & 3 & 1 & 1 \\ 2 & 5 & 1 & 0 \end{bmatrix} \end{array}$$

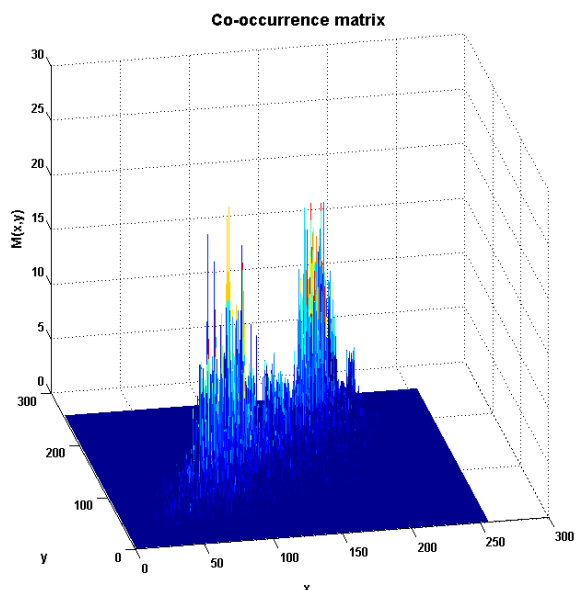
Figur 10: Co-occurrence illustrasjonsfigur. Bildeutsnitt I og co-occurrence matrise M

Co-occurrence metoden er rotasjonsuavhengig da den kun ser på den parvise fargesammensetningen mellom to pikslar.

En co-occurrence matrise har som regel de største verdiene langs diagonalen. Verdiene her kommer fra de stedene i bildet der to nabopikslar er identiske, mens verdiene i områder rundt diagonalen kommer fra områder der fargene er ganske like mellom to nabopikslar. Siden fargene i to nabopikslar ofte er ganske like, er det normalt å anta at man får størst verdier langs diagonalen. Figur 13 viser et diagonalt snitt av en co-occurrence matrise.



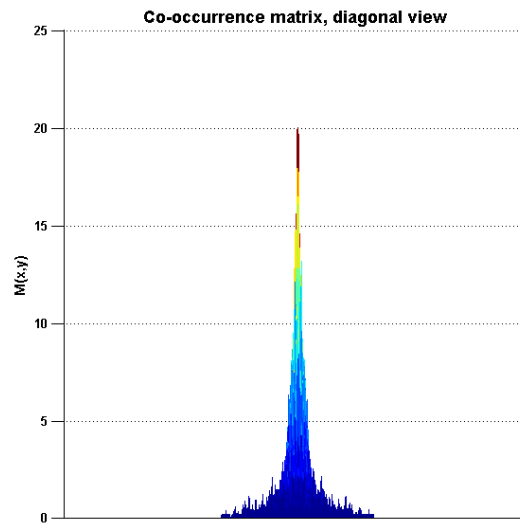
Figur 11: Utsnitt for beregning av co-occurrence matrise



Figur 12: Co-occurrence matrise fra området i figur 11



Figur 12 viser en slik co-occurrence matrise for området vist i figur 11. Matrisen i figur 12 er regnet ut ved å summere co-occurrence matrisene for hver av fargene RGB i bildet.



Figur 13: Co-occurrence matrise diagonalt snitt

## 2.5 Vinkelmetoden

Ved å betrakte bildeutsnittene som vektorer, kan man sammenligne to utsnitt direkte ved å beregne cosinusvinkelen mellom vektorene ved ligning (9). På denne måten vil man i første bilde opprette en referansevektor med dimensjon lik antall piksler i utsnittet. I de påfølgende bildene vil man lete etter bildeutsnitt som ligner på referanseutsnittet.

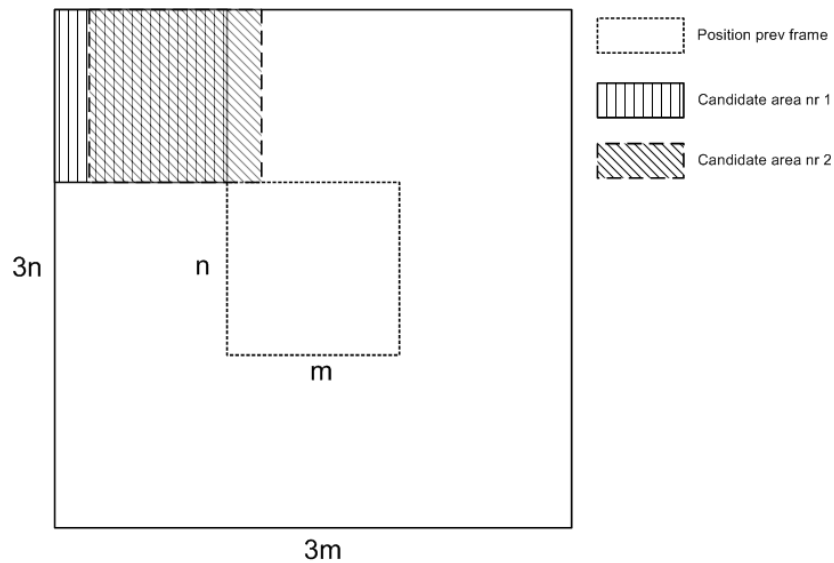
## 2.6 Algoritme for målfølging

Algoritmen for målfølging kan beskrives på følgende måte:

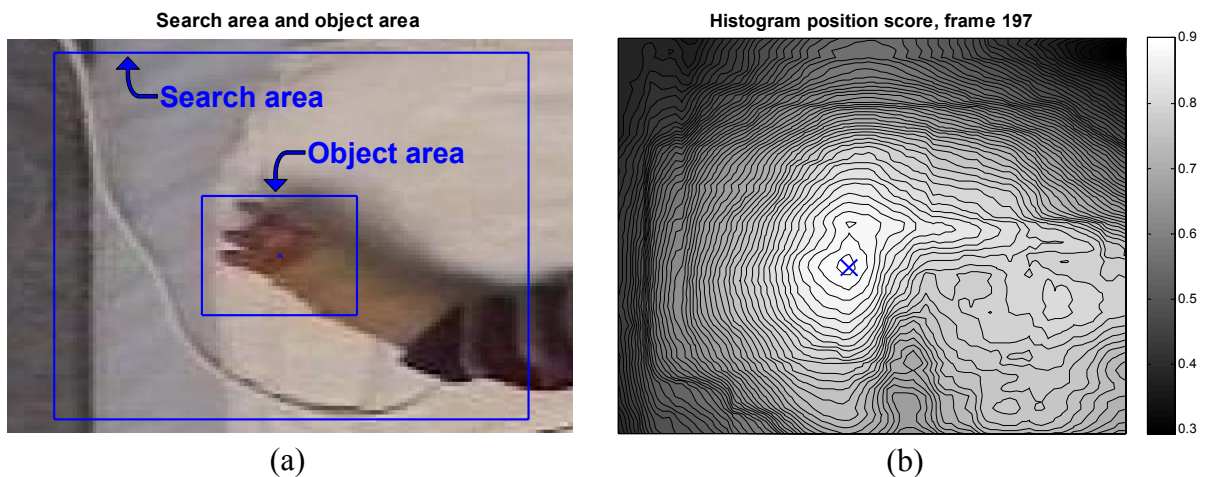
- I første bilde velges et rektangulært område bestående av  $n \times m$  piksler. Dette området omslutter det man ønsker å følge, og det beregnes et histogram, et spatiogram, en co-occurrence matrise og et bildeutsnitt for området. Disse vil fungere som referansemodeller for de forskjellige metodene gjennom hele sekvensen.
- I påfølgende bilde vil det gjøres et søk over et område som har høyde og bredde tre ganger så stort som det opprinnelige objektet,  $3n \times 3m$ .
- Det store søkeområdet vil deles opp i mange små overlappende områder på størrelse med det opprinnelige objektet, der hvert av disse områdene er forskjøvet én pikselrad i forhold til sin nabo. Totalt vil søkeområdet inneholde  $(2n+1)(2m+1) = 4mn + 2(n+m) + 1$  kandidat områder. Dette tilsvarer ca 3000 kandidater ved et bildeutsnitt på  $25 \times 30$  piksler.

Søkeområdet er valgt så stort for å sikre at objektet befinner seg innenfor området.

- For hver objektkandidat blir det beregnet et histogram, et spatiogram, en co-occurrence matrise og et bildeutsnitt. Disse brukes til å evaluere sammenligningsfunksjonen for de forskjellige metodene. Funksjonene er beskrevet nærmere i kapittel 2.6.1 – 2.6.4
- For hver metode antas objektet å ligge i det området med høyest sammenligningsverdi. Dette vil være sentrum for søkeområde i neste bilde. Metodene kjøres parallelt, og har hvert sitt søkeområde og hver sin beste kandidat.



Figur 14: Søkeområdet for målfølgingsalgoritmen



Figur 15: Illustrering av søkeområdet. (a) viser søkeområdet, og det området som antas å inneholde objektet. (b) viser hvordan Bhattacharyya-koeffisienten endrer seg i søkeområdet.

Pseudokode for algoritmen:

```

for all frames f
  for all trackingMethods t
    max=-1
    candidates = findCandidates(bestCandidate(f-1,t))
    for all candidates i
      score = matchScore(candidates(i), objectModel(t))
      if score > max
        bestCandidate(f,t) = max = candidates(i)

```

## 2.6.1 Basert på histogrammer

Histogrammer kan være modeller for det området som man ønsker å følge. Dette gjøres ved at man velger et område som dekker objektet, og lager et histogram av dette. I neste bilde leter man etter et område som et histogram som ligner referansehistogrammet. Denne algoritmen er beskrevet i kapittel 2.6. For å sammenligne to histogrammer benyttes Bhattacharyya-koeffisienten (Bhattacharyya, 1943):

$$\rho = \sum_{b=1}^B \sqrt{n_b n_{b,ref}} \quad (7)$$

Bhattacharyya-koeffisienten er et lengdemål, og vil ligge i område [0 1], der 1 tilsvarer maksimal likhet.

## 2.6.2 Basert på spatioqram

Conaire et al. (2007) viser at et spatioqram  $s = (n, \mu, \Sigma)$  kan sammenlignes med et annet referanse-spatioqram  $\hat{s} = (\hat{n}, \hat{\mu}, \hat{\Sigma})$  på følgende måte

$$\rho = \sum_{b=1}^B \sqrt{n_b \hat{n}_b} \left( 8\pi \left| \Sigma_b \hat{\Sigma}_b \right|^{1/4} N\left(\mu_b; \hat{\mu}_b, 2(\Sigma_b + \hat{\Sigma}_b)\right) \right) \quad (8)$$

der

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{2\pi\Sigma^2}} e^{-\frac{(x-\mu)^2}{2\Sigma^2}}$$

representerer en normalisert Gaussisk funksjon evaluert i x.

$\rho$  vil være et tall i området [0,1] der 1 tilsvarer eksakt likhet, og 0 tilsvarer ingen likhet.

Dette vil fungere som sammenligningsfunksjon for spatioqram-metoden i algoritmen som er beskrevet i kapittel 2.6.

### 2.6.3 Basert på co-occurrence matriser

En co-occurrence matrise kan benyttes som en referansemodell for et område som man ønsker å følge. I bildesekvensen kan man da søke etter områder som har co-occurrence matriser som ligner på modellens matrise.

For å bedre sammenligningsgrunnlaget, vil det lønne seg å benytte en referansematrise som er litt glattere enn den opprinnelige. På denne måten vil metoden bli mindre ømfintlig for små variasjoner mellom matrisene. Glattingen gjøres ved at et element i matrisen settes til en vektet sum av seg selv og dens naboer. Vektene er regnet ut fra en Gaussisk flate med topp i sentrumspikselen, og én piksel radius. Det er testet med forskjellige verdier for standardavviket til den Gaussiske flaten.

$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$
0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01
(a)	(b)	(c)
$\sigma = 1.5$	$\sigma = 2.5$	$\sigma = 5$
0.09	0.12	0.09
0.12	0.15	0.12
0.09	0.12	0.09
(d)	(e)	(f)

Figur 16: Filtreeringsmatriser som funksjon av  $\sigma$

	Mean	Std
$\sigma = 0.5$	8.08	6.77
$\sigma = 0.75$	6.78	5.31
$\sigma = 1$	6.72	5.61
$\sigma = 1.5$	6.60	5.58
$\sigma = 2.5$	6.56	5.67
$\sigma = 5$	6.41	5.28

Tabell 1: Oppsummering av resultatene for co-occurrence ved forskjellige valg av  $\sigma$

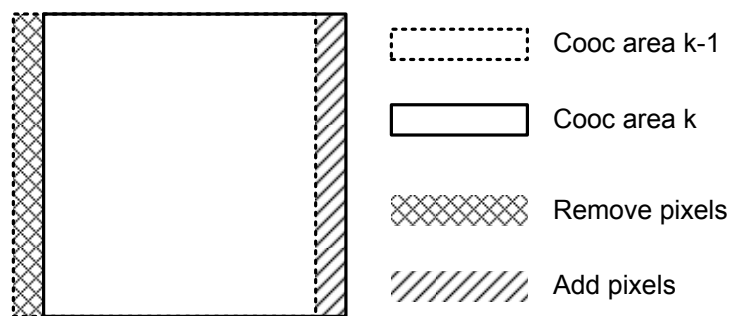
Gjennom en testsekvens er det registrert avviket mellom metodens posisjonsestimat den faktiske posisjonen til målobjektet. Tabell 1 oppsummerer sekvensen ved å vise gjennomsnittlig avvik fra virkelig posisjon, samt standardavviket for hvor mye dette avviket varierer i tidsserien. Her ser man at det er standardavvik lik fem som gir best resultat. Dette tilsvarer en lik vektning av alle elementene, som vist i figur 16 (f).

To co-occurrence matriser,  $M$  og  $\hat{M}$ , kan sammenlignes ved å regne ut cosinusvinkelen mellom dem. Matrisene re-arrangeres til vektorer med dimensjon  $\text{antallElementer} \times 1 = n \times 1$ . Deretter kan man beregne cosinusvinkelen i det  $n$ -dimensjonale rommet ved:

$$\rho = \cos(\theta) = \frac{M \cdot \hat{M}}{|M| |\hat{M}|} = \frac{\sum_{i=1}^n m_i \hat{m}_i}{\sqrt{\sum_{i=1}^n m_i^2 \sum_{i=1}^n \hat{m}_i^2}} \quad (9)$$

$\rho$  vil nå være et tall i området  $[-1,1]$ , der 1 tilsvarer best likhet, og -1 tilsvarer minst likhet.

I algoritmen for målfølgning regnes det ut mange co-occurrence matriser for overlappende områder. Det betyr at datagrunnlaget for to nabo-områder er svært likt, og at man kun trenger å modifisere den forrige co-occurrence matrise for å finne matrisen for det gjeldende området.



Figur 17: Effektiv utregning av nabo co-occurrence matriser

## 2.6.4 Basert på vinkelmetoden

Vinkelmetoden bruker cosinusvinkelen mellom bildeutsnittene for å sammenligne to bilder. Ligningen blir dermed slik som ligning (9), men med bildeutsnittene direkte som vektorer i ligningen.

## 2.6.5 Valg av antall farger

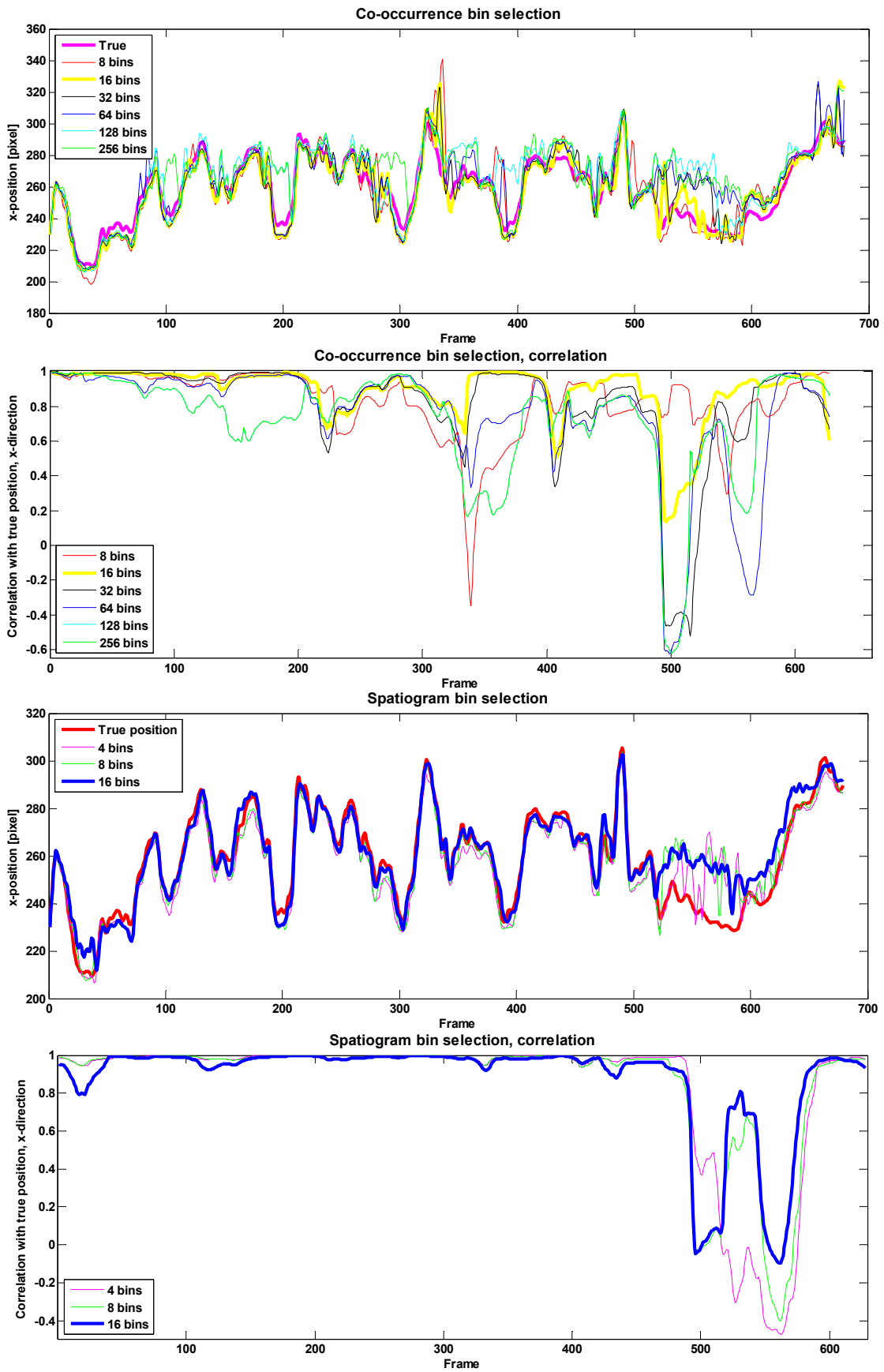
Det er gjort tester med forskjellig valg av antall farger i bildet for co-occurrence metoden og spatiogram-metoden. Målfølgningen har tatt utgangspunkt i samme referanseområde, og posisjonsestimatene er filtrert slik som beskrevet i kapittel 2.8.

For hvert bilde gjennom hele måleserien er det regnet ut pikselstanden mellom metodens posisjonsestimat, og den virkelige posisjonen. I tabell 2 vises gjennomsnittlig avvik fra virkelig posisjon, samt standardavviket for hvor mye dette avviket varierer i tidsserien. Begge metodene er best ved å redusere antall farger til 4096, og ved bruk av histogramutjevning.

Bins	Colors	Spatioqram				Co-occurrence			
		HistEq		Not histEq		HistEq		Not histEq	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
4	64	6,15	5	6,09	5,03	59,42	48,3	31,61	35,12
8	512	6,01	5,62	6,88	5,88	8,24	7,67	10,94	11,21
16	4 096	<b>5,97</b>	<b>5,63</b>	7,06	6,04	<b>6,72</b>	<b>5,61</b>	7,97	7,57
32	32 768	-	-	-	-	7,8	6,97	8,08	7,06
64	262 144	-	-	-	-	9,17	8,22	8,81	8,03
128	2 097 150	-	-	-	-	10,95	9,09	11,19	9,39

**Tabell 2: Oppsummering av valg av antall farger**

Merk at det ikke er kjørt tester på spatioqram ved flere farger enn 4096, da dette tar uforholdsmessig lang tid. Se mer om kjøretidsanalyse i kapittel 3.1.



Figur 18: Grafer for målfølgning ved forskjellig valg av antall farger

## 2.6.6 Målfølging med modelloppdatering

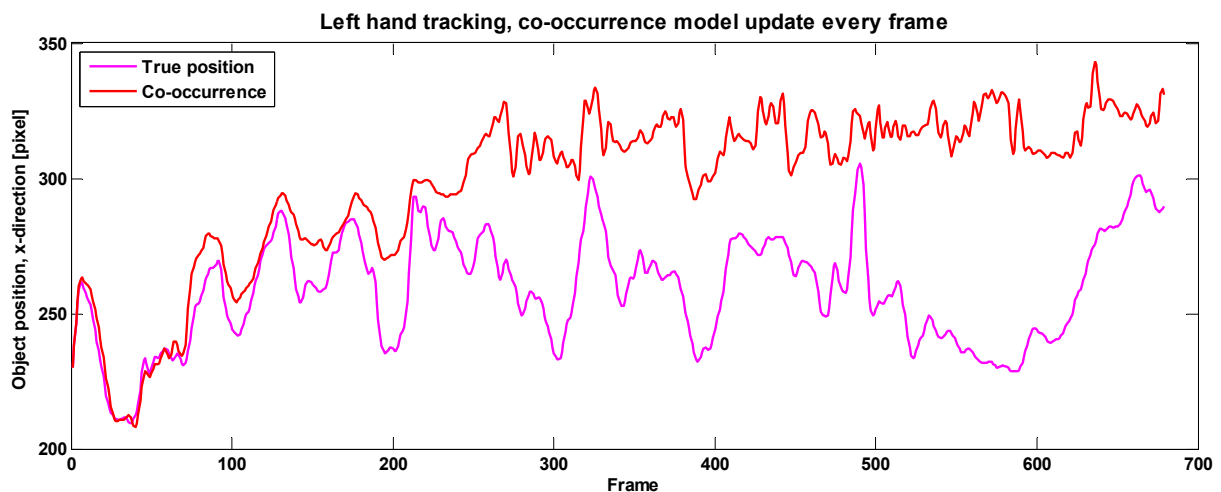
Det er testet om oppdateringer av modellreferansene underveis i bildesekvensen vil gi et positivt utslag på målfølgingen. Dette kapittelet er delt i to deler, der den første delen tar for seg modelloppdatering for metodene hver for seg, mens den andre deler foreslår måter metodene kan kombineres på for å bedre målfølgingen.

### Individuell modelloppdatering

#### Co-occurrence

Referansemodellen for co-occurrence metoden kan oppdateres på to måter som er forklart i de neste avsnittene.

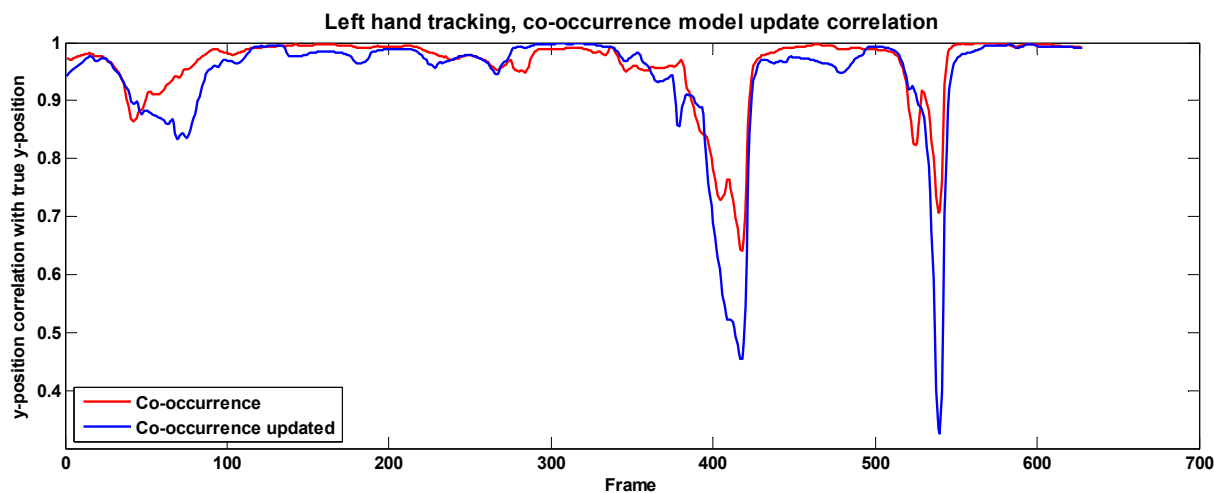
Man kan opprette en ny referansemodell for hvert bilde. Dette medfører at den beste objekt kandidaten blir brukt som referanse for påfølgende bilde. Tester har vist at denne metoden vil være god i kort tid, men vil medføre drift på sikt. Som man ser av figur 19 mister algoritmen objektet etter 80 bilder.



Figur 19: Co-occurrence modelloppdatering for hvert bilde

Et annet alternativ for modelloppdateringen er å ta vare på informasjonen fra alle tidligere kandidater. Siden sammenligningsfunksjonen for co-occurrence matrisene kun baserer seg på vinkelen, kan man oppdatere referansemodellen ved å summere den med en god kandidatmodell. Hvis matrisene er helt identiske vil vektoren kun bli lenger, mens retningen er den samme. Hvis de derimot er litt ulike vil vinkelen få en liten dreining i retning mot den nyeste co-occurrence matrisen.





**Figur 20: Modeloppdatering av co-occurrence metoden, kun ved gode treff**

Som man kan se i figur 20 vil ikke metoden med modeloppdatering forbedre resultatene. Begge metodene klarer å følge målet, men metoden med oppdatering varierer mer, og gir derfor dårligere korrelasjon med virkelig posisjon. I tillegg er det også en fare for drift med bruk av denne oppdateringsmetoden. På grunn av dette er det valgt å ikke bruke denne måten for modeloppdatering.

Gjennom hele tidsserien er det registrert avstanden mellom posisjonsestimaten og de virkelige posisjonene. Tabell 3 oppsummerer disse resultatene. Her ser man at det er minst gjennomsnittlig avvik ved å ikke benytte noen form for modeloppdatering.

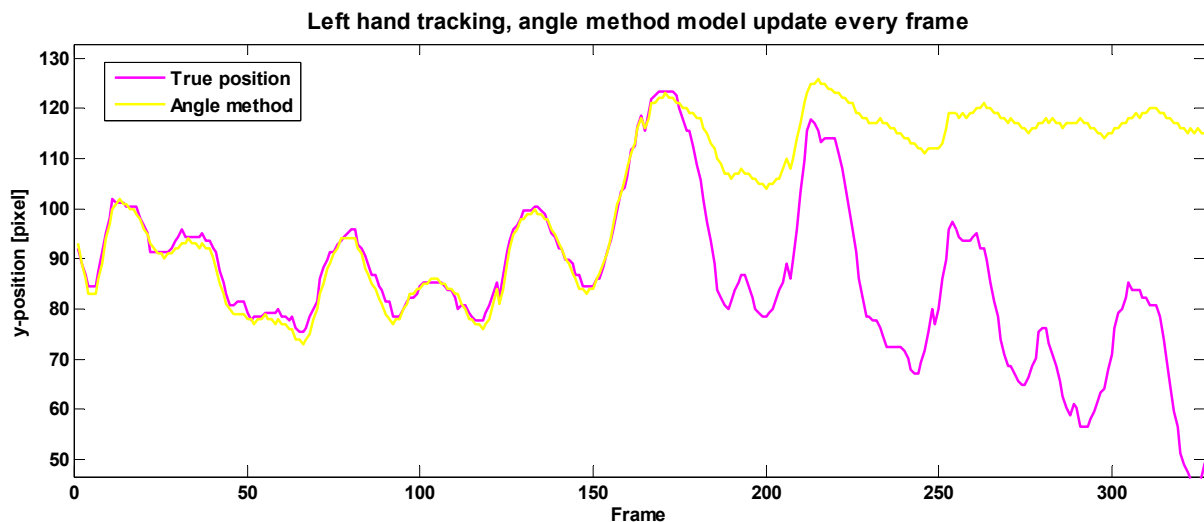
Se flere plott i appendiks E.

Method	Mean	Std
Regular	6,42	5,28
Model update on good hits	8,12	7,44
Model update on every frame	43,27	26,30

**Tabell 3: Oppsummering av modeloppdatering for co-occurrence metoden**

### Vinkelmetoden

Man kan også oppdatere modellen for vinkelmetoden underveis i sekvensen. Dette gjøres ved å sette referanseutsnittet til det beste posisjonsestimaten for hvert bilde. Det vil si at området som fikk høyest sammenligningsverdi i gjeldene bilde, brukes som referansemodell for neste bilde.



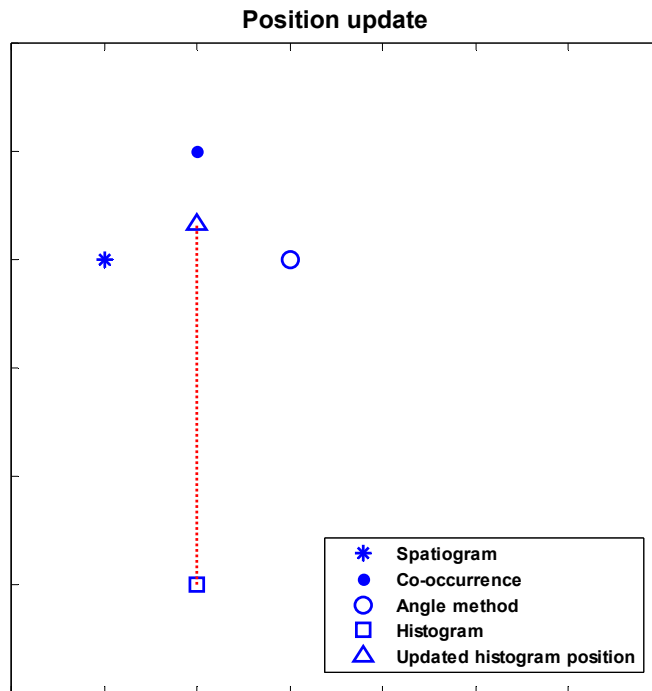
**Figur 21: Vinkelmetoden med modelloppdatering**

Svakheten ved denne metoden er at systemet ikke vil ha noen hukommelse, da den kun sammenlignes med forrige bilde. Som man kan se i figur 21 er resultatet at posisjonsestimateret drifter vekk fra korrekt posisjon etter en stund.

## **Felles modelloppdatering**

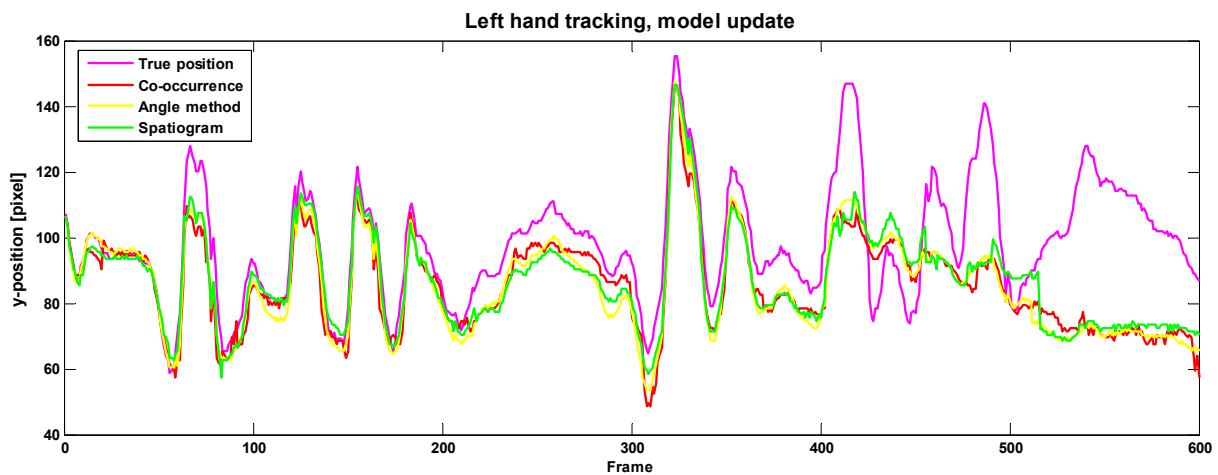
### **Ny modell**

For å unngå at posisjonsestimaterne fra de forskjellige metodene sklir fra hverandre, kan man kombinere de på en slik måte at de kan dra hverandre inn igjen. Ideen er at hvis avstanden fra et posisjonsestimater til hver av de andre er over dobbelt så stor som avstanden mellom de andre, antar man at estimateret er feil. Figur 22 illustrerer dette prinsippet. Metodens objektposisjon blir satt til å være gjennomsnittsposisjonen blant de resterende, og det blir laget en ny referansemodell for metoden i denne posisjonen.



**Figur 22: Prinsipp for å oppdatere et posisjonsestimert**

Det er gjort tester med denne formen for modelloppdateringer. Resultatet vises i figur 23, og man kan se at posisjonsestimertene følger hverandre, men drar hverandre ut av kurs etter en stund.



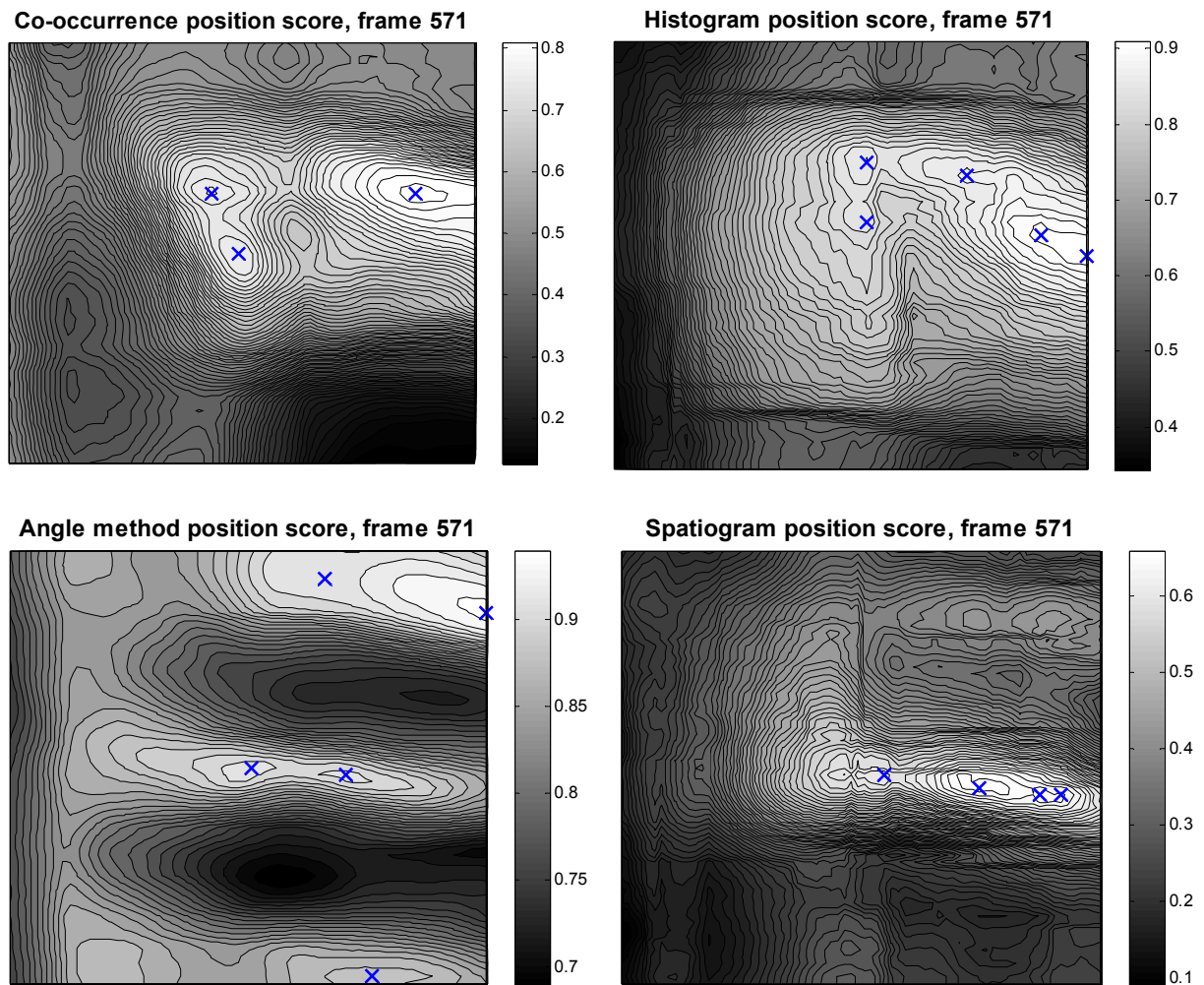
**Figur 23: Posisjonsestimert ved bruk av modelloppdatering**

### Minste omkrets

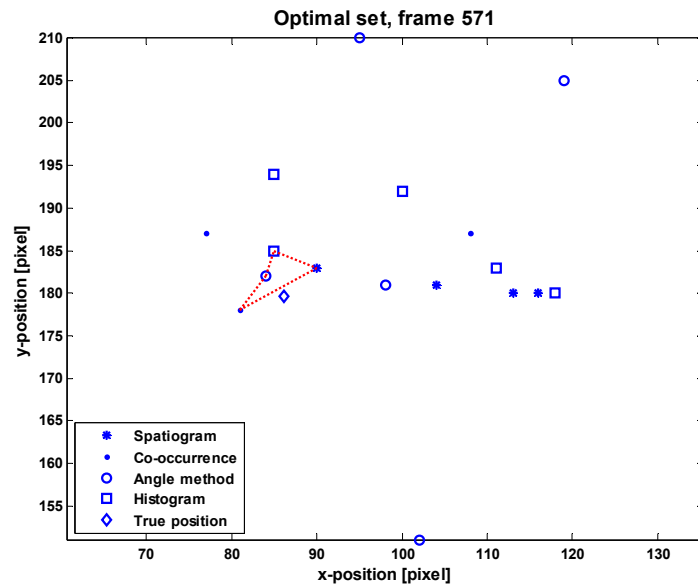
I kapittel 2.6 ble posisjonen for en gitt metode bestemt til å være der sammenligningsverdien var høyest. Dette kan gi et galt estimert hvis fordelingen er forholdsvis jevn. I figur 42 fra kapittel 3.3.1 ser man at fordelingen har en klar og entydig topp, og at det derfor er naturlig å anta at objektet befinner seg der. Men hvis vi ser på resultatene fra bilde 571, ser vi at det er flere lokale maksimumspunkt i fordelingen av posisjonscore. En mulig posisjonskandidat er

et lokalt toppunkt som har verdi større enn 90 % av maksimalverdien. I figur 24 ser man kandidatene ved de forskjellige metodene.

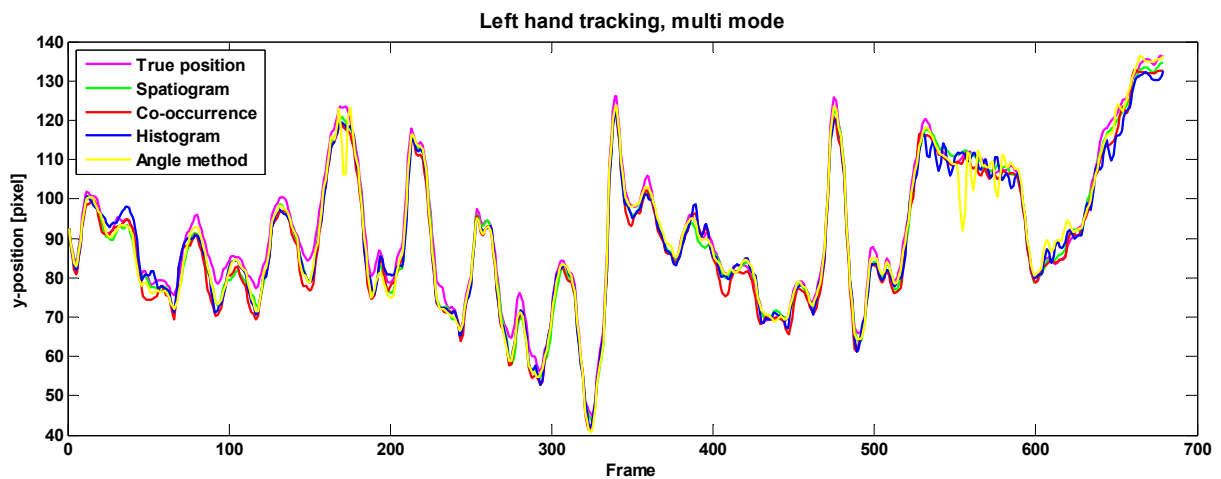
Metoden for minste omkrets går ut på å velge kandidater fra de forskjellige representasjonsmetodene slik at de danner område med så liten omkrets som mulig. Figur 25 illustrerer dette. Det er med andre ord ikke sikkert at det er posisjonsestimater med høyest verdi som blir valgt.



Figur 24: Posisjonskandidater fra de forskjellige metodene



Figur 25: Optimalt valg av posisjonskandidater



Figur 26: Tidsserie ved bruk av optimalt valg av posisjonskandidater

Figur 26 viser at denne måten å kombinere posisjonsestimaterne på, gjør at alle metodene klarer å følge objektet gjennom hele sekvensen. Se kapittel 3.3.2 for mer detaljerte resultater ved bruk av denne metoden.

Pseudokode for følgingialgoritme med posisjonsoptimalisering:

```

for all frames f
  for all trackingMethods t
    max=-1
    candidates = findCandidates(bestCandidate(f-1,t))
    for all candidates i
      score = matchScore(candidates(i), objectModel(t))
      if score > max
        bestCandidate(f,t) = candidates(i)
    bestCandidate(f, all) = combineMethods(bestCandidate(f, all))

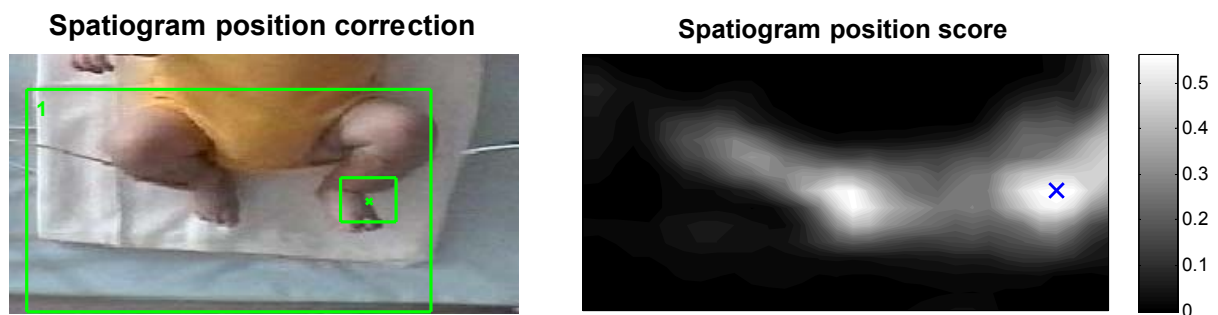
```

## 2.6.7 Målfølging med flere referansemodeller

Føttene kan være utfordrende å følge siden de endrer form, ligner på hverandre, endrer farge på grunn av skygger, beveger seg mye og kan bevege seg over hverandre. Dette vil stille ekstra krav til algoritmen for at den skal bli robust nok. Derfor er det utviklet ekstra funksjonalitet til algoritmen når den skal brukes til å følge føttene.

### Føttene ligner på hverandre

Høyre og venstre fot har ofte lik farge og form, og hvis de beveger seg mot hverandre kan algoritmen hoppe over på feil fot når de igjen beveger seg fra hverandre. Som en løsning på dette problemet utvides søkeområdet hvis det er mistanke om at feil fot følges. Kriteriet for at dette skal skje, er at høyre fot befinner seg i venstre halvplan, og omvendt. Da vil søkeområdet finne begge føttene, og deretter hoppe tilbake på den opprinnelige foten. Dette er uavhengig om sammenligningsverdien til feil fot var høyest.

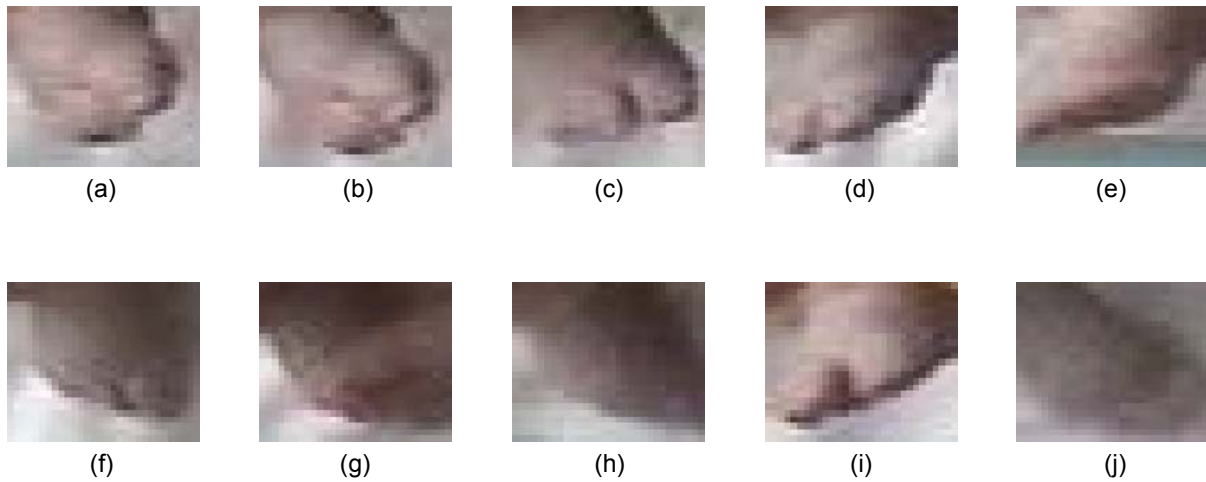


Figur 27: Justering av posisjon etterfølging av feil fot

Figur 27 viser en situasjon der spatiogram-metoden skal følge foten til høyre. Underveis i sekvensen har ungen hatt føttene sammen, og da de beveget seg fra hverandre igjen, valgte algoritmen å følge foten til venstre. Dette ble oppdaget ved at posisjonsestimatet befant seg i venstre halvplan. Dermed utvidet søkeområdet seg, og algoritmen forkaster posisjonsestimatet lengst til venstre, og velger det til høyre.

### Føttene endrer form

Siden føttene ofte endrer farge og form underveis i sekvensen, vil algoritmen feile hvis den ikke finner foten i søkeområdet. Som en løsning på dette er det lagt inn flere referansemodeller for en fot i algoritmen. I starten av sekvensen blir brukeren bedt om å markere foten på ti forskjellige bilder fordelt over hele sekvensen. På denne måten blir det lagret referansemodeller for foten i forskjellige former og farger. Underveis i sekvensen vil algoritmen gjøre et søk blant alle disse modellene for hvert fjerde bilde. Dette fungerer ved at samtlige objektkandidater blir sammenlignet med alle modellene, og man velger å bruke den modellen som gir høyest sammenligningsverdi.



**Figur 28: Referanseobjekter ved bruk av målfølging med flere referansemodeller**

Pseudokode for algoritmen:

```

for all frames f
  if mod(f, 4) = 0
    for all trackingMethods t
      max=-1
      candidates = findCandidates(bestCandidate(f-1,t))
      for all candidates i
        for all objectModels m
          score = matchScore(candidates(i), objectModel(t,m) )
          if score > max
            modelRef(t) = m
            max = score

    for all trackingMethods t
      max=-1
      candidates = findCandidates(bestCandidate(f-1,t))
      for all candidates i
        score = matchScore(candidates(i), objectModel(t, modelRef(t) ) )
        if score > max
          bestCandidate(t) = candidates(i)

bestCandidate(f, all) = combineMethods(bestCandidate(f, all))

```

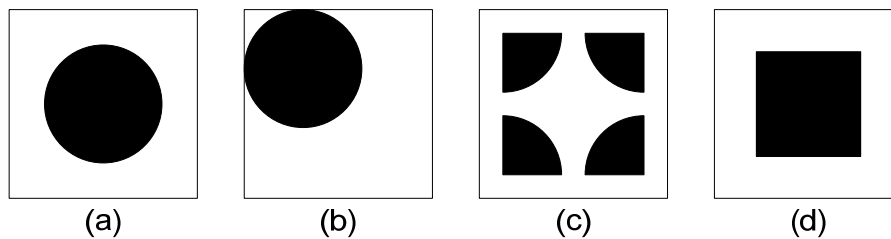
## 2.7 Sammenligning av metodene

I kapittel 2.2 – 2.5 presenteres det forskjellige metoder for objektrepresentasjon og målfølging i bildesekvenser. Dette kapittelet skal gi en oppsummering av metodene, samt påpeke fordeler og svakheter ved metodene, slik de blir benyttet i det foreliggende datasettet. Det vil også gies en begrunnelse for valg av metode til de forskjellige målobjektene.

Valg av algoritme er avhengig av det objektet som skal følges, og hvordan objektet antas å endre seg underveis i sekvensen. Hendene og føttene kommer til å bevege seg mye, samt endre farge og form, mens hode og brystet vil ligge mer i ro og ikke endre utseende i vesentlig grad.

## Histogram

Denne metoden baserer seg på den statistiske fargefordelingen i bildet. Metoden teller forekomstene av hver farge eller fargegruppe, og bruker dette som sammenligningsgrunnlag for to bildeutsnitt. Figur 29 viser fire bildeutsnitt som har lik fordeling av svart og hvitt, og dermed likt histogram, men som er forskjellige på mange andre måter. Histogrammetoden vil tro at disse bildene er identiske. Dette fører til at metoden ikke fungerer tilfredsstillende når man skal følge objekter som ikke skiller seg klart ut med fargen, og der mange objekt kandidater har forholdsvis lik fargefordeling.



Figur 29: Bilder som har likt histogram

## Spatioqram

Spatioqrammet er en utvidelse av histogrammet, der man også tar med fargenes massesenter og varians i bildeutsnittet. Dette gjør at spatioqram-metoden blir vesentlig bedre enn histogrammetoden, og metoden klarer å skille bildene i figur 29 fra hverandre. Selv om (a), (c) og (d) har identisk histogram og massesenter for fargene, er variansen forskjellig i dem alle. Dette gjør at metoden i alle sammenhenger vil være bedre og mer robust enn histogrammetoden, samt at den klarer å definere et referanseutsnitt på en mer detaljert måte.

På grunn av begrenset regnekraft og minne i datamaskinen, må man redusere antall farger i bildet fra  $256^3$  til  $16^3$  for at beregningene ikke skal ta for lang tid. Dette gjør at metoden ikke egner seg til å følge områder med forholdsvis lik farge i hele utsnittet. Da er det fare for at mesteparten av fargene havner i samme fargegruppe, og metoden blir dermed følsom for endringer i lysintensitet. På grunn av dette vil metoden ikke egne seg til å følge ansikt og bryst, men vil være godt egnet til bruk på hender og føtter.

## Co-occurrence

Co-occurrence metoden er basert på fordelingen av forskjellige kombinasjoner av fargeintensitet mellom to piksler som er ved siden av hverandre. En piksel har åtte nabopiksler, og ved å beregne co-occurrence matrisen på grunnlag av alle disse blir metoden rotasjonsuavhengig.

Ved bruk av fargebilder blir co-occurrence matrisen beregnet i hvert fargelag for seg (for RGB bilder), og disse tre matrisene blir referansegrunnlaget for et bildeutsnitt. Reduksjonen ved å kun beregne matrisen i R, G og B delen hver for seg er gjort på grunn av begrenset minnekapasitet. Hvis matrisen skulle vært beregnet for alle tenkelige fargekombinasjoner i et 8 bit RGB bilde, ville den fått  $256^6$  elementer. Dette tilsvarer ca 260 TB data. Selv ved å

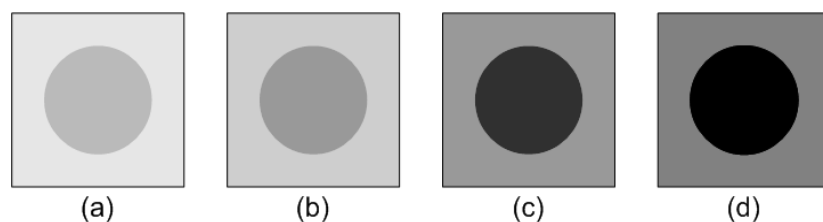


redusere antall farger i bildet til  $16^3$ , vil matrisen kreve 16 MB, en størrelse som fortsatt er lite håndterbart å gjøre beregninger med.

Co-occurrence metoden egner seg godt når objektet kan roteres underveis i sekvensen, men samtidig beholder samme form og samme bakgrunn. Dette gjelder i en viss grad for hender og føtter. Men siden metoden ikke klarer å skille objektkandidater godt fra hverandre når søkeområdet inneholder mange like farger, vil ikke metoden fungere tilfredsstillende ved følging av hode og bryst.

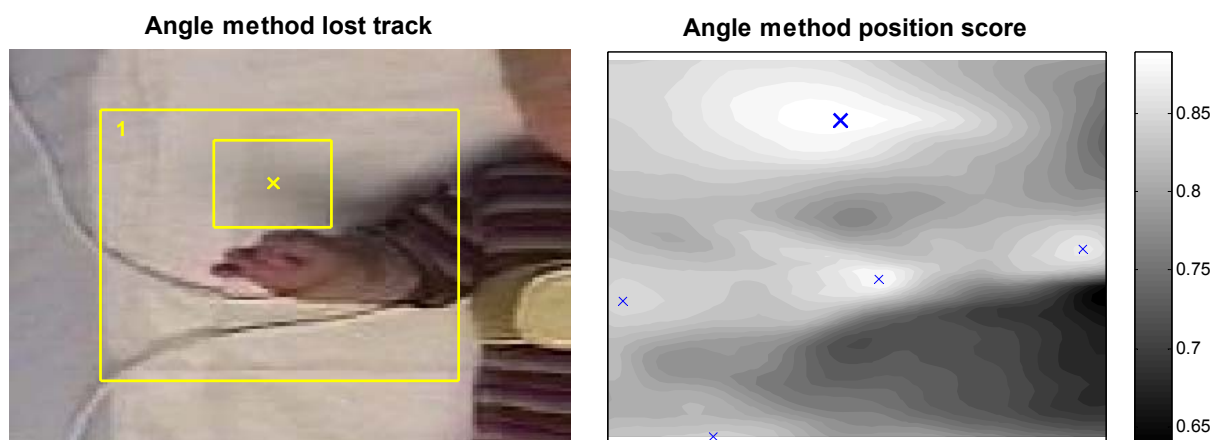
### Vinkelmetoden

Vinkelmetoden bruker bildeutsnittet direkte ved å beregne cosinusvinkelen mellom utsnittet og et referanseutsnitt. Denne vinkelen vil være i et vektorrom som har dimensjon lik antall piksler i utsnittet, der en cosinusvinkel på 1 tilsvarer best mulig likhet, og -1 tilsvarer minst mulig likhet. Siden vinkelen er avhengig at vektorelementenes innbyrdes forhold, kan flere mulige kandidater gi lik sammenligningsverdi. I figur 30 har alle sirklene dobbel så mørk farge som bakgrunnen. Dette gjør at forholdet mellom sirkelen og bakgrunnsfargen er konstant for alle bildene, og dermed vil de innbyrdes cosinusvinklene bli 1.



Figur 30: Bilder som har innbyrdes lik vinkel

Figur 31 viser problematikken med skyggeområder som ligner på objektreferansen. 160 bilder ut i sekvensen har hånden endret seg så mye at skyggens fargefordeling ligner mer på referansen enn det hånden gjør.



Figur 31: Vinkelmetoden mister følging av armen

Selv om metoden fungerer dårlig når objektet endrer form, fungerer den bra når objektet har nesten konstant form og farge gjennom hele sekvensen. Dette er tilfelle for bryst og ansikt, og her er metoden bedre enn de andre. Fordelen med vinkelmetoden framfor de andre metodene er at den ikke reduserer antall farger i bildet.

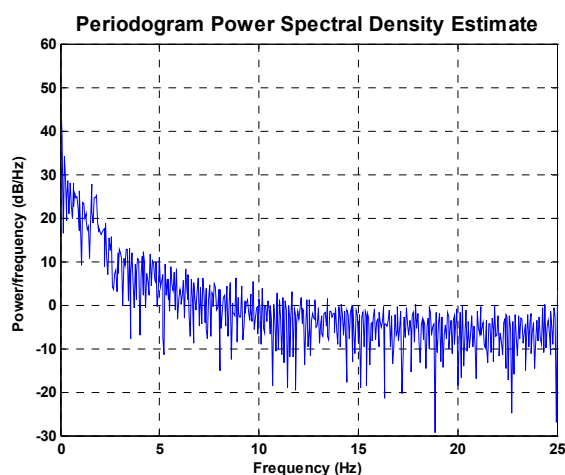
Tabell 4 oppsummerer fordelene, svakheter og bruksområde ved de forskjellige metodene.

	Fordeler	Svakheter	Bruksområde
Histogram	+ Rotasjonsuavhengig + Intuitiv metode + Rask beregningstid	– Lite grunnlag for sammenligning – Mange kandidater kan ha tilnærmet lik sammenligningsverdi – Følsom for endring av lys	Hender Føtter
Spatioqram	+ Basert på 1. og 2. ordens statistikk + Gir signifikant best score på riktig objekt	– Lang beregningstid – Ikke rotasjonsuavhengig – Følsom for endring av lysforhold	Hender Føtter
Co-occurrence	+ Rotasjonsuavhengig + Teksturklassifisering + Basert på 2.ordens statistikk	– Tar hvert fargerom for seg – Kan finne områder som gir høyere score uten at det er objektet	Hender Føtter
Vinkelmetoden	+ Rask beregningstid + God når objektets form endres lite	– Dårlig når objektet endrer form – Kan finne områder som gir høyere score uten at det er objektet – Ikke rotasjonsuavhengig	Hode Bryst

**Tabell 4: Fordeler og svakheter ved de forskjellige metodene**

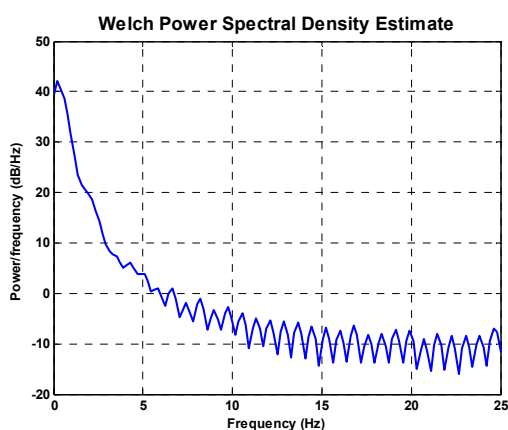
## 2.8 Fjerning av høyfrekvent støy

Posisjonsestimatene fra de forskjellige metodene inneholder mye høyfrekvent støy. Med dette menes at estimatene ikke treffer helt korrekt posisjon i hvert bilde, men varierer rundt. Dette gjør at trajektorien for posisjonsestimatet ikke blir så jevn som den virkelige posisjonstrajektorien. For å bedre dette kan estimatene filtreres i et lavpassfilter. For å finne ut hvilke frekvenser som er aktuelle å filtrere bort, er det gjort en analyse for å finne hvilke frekvenser ungens bevegelser har. Gjennom en videosekvens er håndens posisjon manuelt registrert, og figur 32 viser spektrumet til dette signalet.

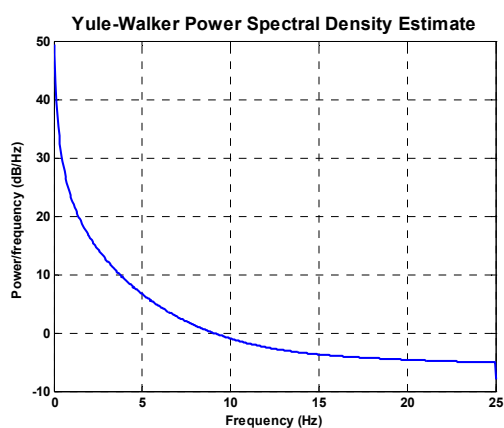


Figur 32: Spektrum til barnets virkelige bevegelser

Dette gir et støyfylt plot, og det kan være vanskelig å se spekteret. For å se spekteret uten så mye støy kan man benytte et Welch-spektrum (Welch, 1967) eller et Yule-Walker-spektrum (Stoica et al., 1997).



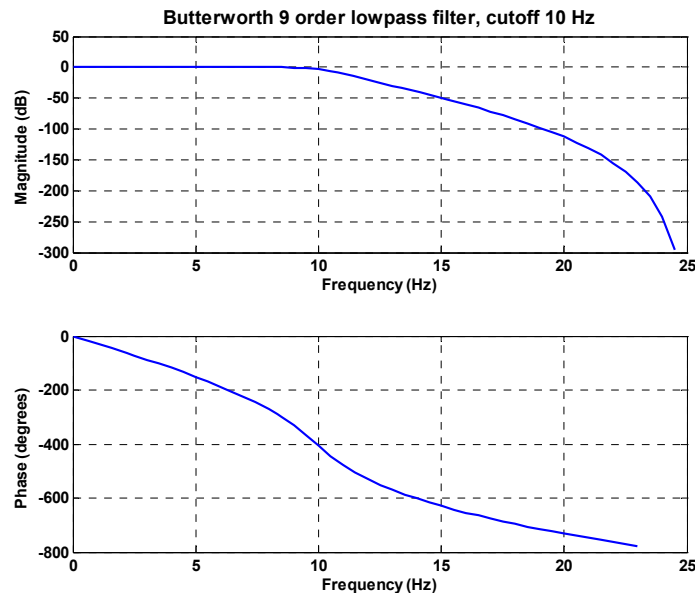
Figur 33: Welch-spektrum



Figur 34: Yule-Walker-spektrum

Man kan nå se at det er lite bevegelser med frekvens over 10 Hz, noe som stemmer bra for mennesker. Mann et al. (1989) har gjort undersøkelser med bevegelser for dagligdage

formål, og resultatet viser at bevegelsene ligger stort sett rundt 1 Hz, og 75 % under 5 Hz. Dette, i kombinasjon med egne resultater, viser at man kan filtrere vekk alle signalene over 10Hz. Til dette er det benyttet et Butterworth lavpassfilter. Dette filteret er karakterisert ved maksimal flat respons i passbåndområdet, og sterk dempning ellers.



Figur 35: Karakteristikk for et Butterworth lavpassfilter

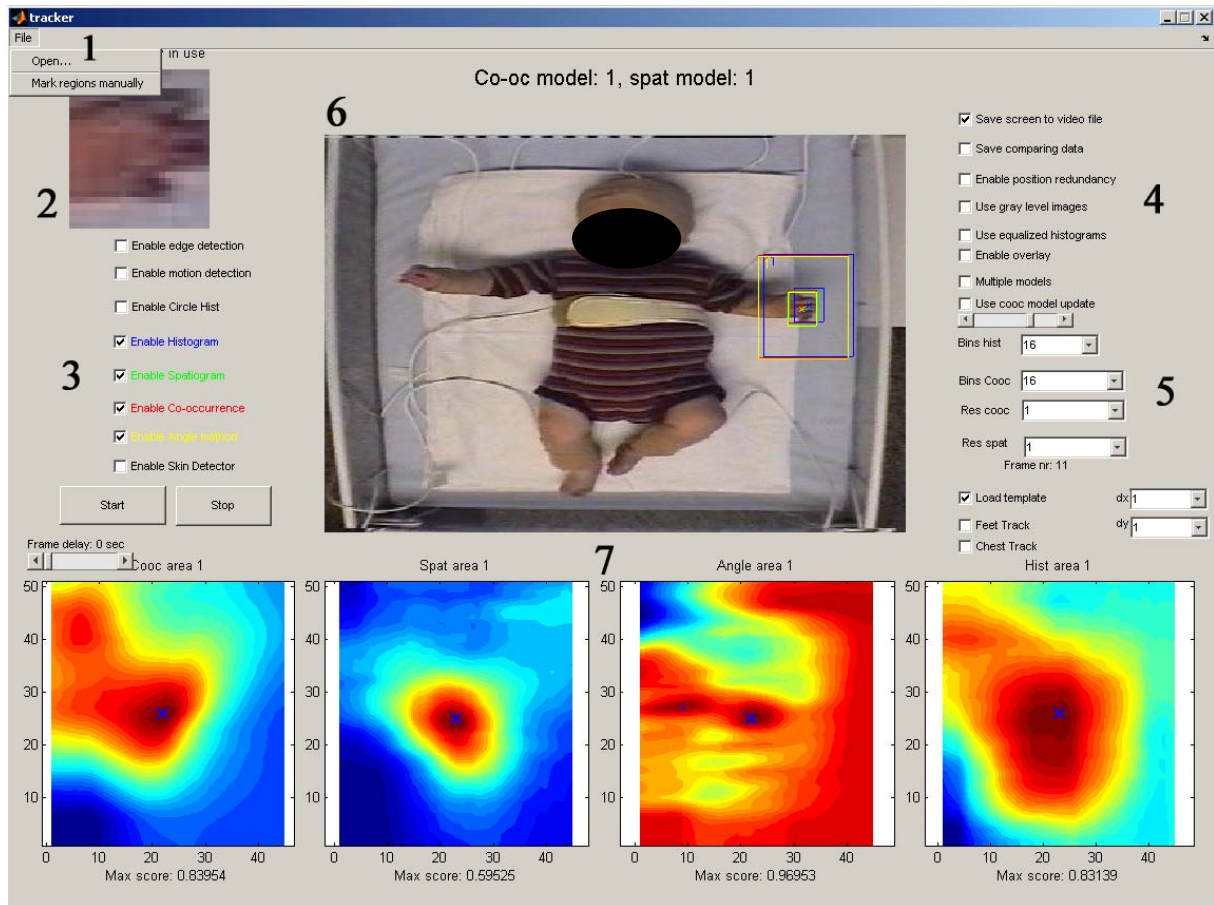
## 2.9 Applikasjon for analyse av metodene

Det er utviklet et program for testing og analyse av metodene som er beskrevet i kapittel 2. Programmeringen er utført i MATLAB R2007b, med Image Processing Toolbox.

Applikasjonen er basert på et grafisk brukergrensesnitt, som vist i figur 37. Her kan brukeren bestemme hvilket datagrunnlag som skal brukes. Det kan enten være en videosekvens, eller en mappe med stillbilder. Brukeren kan også sette parametere for følgingsalgoritmen, se nærmere forklaringer i tabell 5. Når algoritmen er ferdig, vil resultatet med posisjonstrajektoriene bli skrevet til en fil. Hvis brukeren ønsker det, blir det også lagret en videosekvens der man kan se posisjonsestimatene underveis i følgingen.



**Figur 36: Input og output til følgingsalgoritmen**

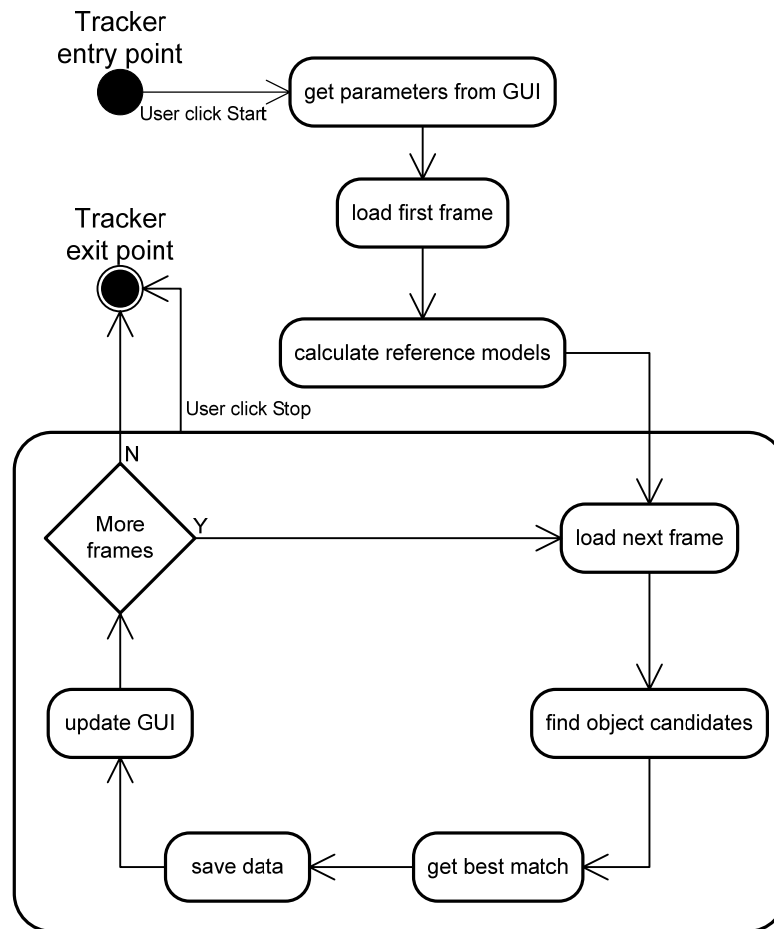


**Figur 37: Skjerm bilde av applikasjonen**

Område	Felt	Beskrivelse
1	Open	Velger hvilket datagrunnlag som skal brukes. Enten en video (avi) eller en mappe med stillbilder.
	Mark Regions	Går gjennom gjeldene bildesekvens, og brukeren markerer objektets lokasjon for hvert bilde. Posisjonene blir til slutt lagret i en fil.
2	Bilde	Viser det bildeutsnittet som blir brukt som referansemodell.
3	Enable edge detection	Aktiverer algoritmen for kantdeteksjon.
	Enable motion detection	Aktiverer algoritmen for bevegelsesdeteksjon.
	Enable circle hist	Aktiverer metoden for histogrammer basert på sirkelkoordinater. Ikke fullført.
	Enable histogram	Aktiverer histogrammetoden.
	Enable spatiogram	Aktiverer spatiogram-metoden .
	Enable co-occurrence	Aktiverer co-occurrence metoden.
	Enable angle method	Aktiverer vinkelmetoden.
	Enable skin detektor	Aktiverer algoritmen for huddeteksjon.
	Start	Starter algoritmen for målfølgning. Se figur 38 for nærmere beskrivelse av algoritmen.
	Stop	Stopper algoritmen for målfølgning.
	Frame delay	Gir brukeren muligheten til at algoritmen skal ta en pause mellom hvert bilde. Dette gjøres hvis man vil studere skjermbildet.
4	Save screen to video file	For hvert bilde lagres skjermbildet. Dette settes inn i en videosekvens. Se digitalt vedlegg for eksempel.
	Save comparing data	Lagrer sammenligningsdata for hvert bilde.
	Enable position redundancy	Aktiverer metoden for optimalt valg av posisjonskandidater.
	Use gray level images	Konverterer fargebilder til gråtonebilder.
	Use equalized histograms	Benyttter algoritmen for histogramutjevning på datagrunnlaget.
	Enable overlay	Brukes til sladding av videosekvensen. En svart firkant blir plassert der musa befinner seg..
	Multiple models	Aktiverer algoritmen som bruker flere modellreferanser.
	Use cooc update	Aktiverer metoden for å oppdatere co-occurrence matrisen ved gode treff. Grensen for godt treff settes her.
5	Bins hist	Velger bins for histogram- og spatiogram-metoden.
	Bins cooc	Velger bins for co-occurrence metoden.
	Res cooc	Velger søkeoppløsning for co-occurrence metoden. Hvis 1 så søkes det i hver kandidat, hvis 2 søkes annenhver kandidat osv.
	Res spat	Velger søkeoppløsning for spatiogram-metoden.
	Frame nr	Viser bildenummeret i sekvensen.
	Load template	Brukes for å velge samme søkeobjekt som forrige søk.
	Feet track	Aktiverer spesialalgoritme for følgning av føtter.
	Chest track	Aktiverer spesialalgoritme for følgning av bryst.
	dx	Setter søkeområdet i x-retning. 1 tilsvarer objektets x-størrelse +- 1*objektets x-størrelse, 0.5 tilsvarer +-0.5*objektets x-størrelse osv.
	dy	Setter søkeområdet i y-retning.
6	Bilde	Viser oversiktsbildet av ungen og de forskjellige metodenes søkeområde og objektområde.
7	Bilder	Viser sammenligningsverdi for de forskjellige metodene.

**Tabell 5: Beskrivelse av komponentene i applikasjonen**

Figur 38 illustrerer hvordan følgingsalgoritmen fungerer, og figur 39 viser hvordan funksjonene bygger på hverandre.



**Figur 38: ULM diagram for algoritmen**

GUI (tracker.m)					
getBestPositionCombination					
getBestSpatMatch		getBestHistMatch		getBestCoocMatch	getBestAngleMatch
getSpatioqram	compareSpatioqrams	getHistogram	compareHistograms	getCoocMatrix	getCosineAngle

**Figur 39: Oversikt over hvordan funksjonene er relatert til hverandre**

## 3 Resultat og analyse

I dette kapitlet presenteres de resultatene som er oppnådd ved følgende av hode, hender, føtter og bryst hos et utvalg av spedbarn. Metodene som er brukt er beskrevet i kapittel 2.

### 3.1 Kjøretidsanalyse

Det er gjort kjøretidsanalyser på de forskjellige metodene. Det er målt hvor lang tid det tar å opprette data for en objektkandidat, samt å sammenligne denne kandidaten med referanseobjektet. Analysen er gjort i MATLAB, og det er benyttet 8 bit fargebilder for alle metodene. Tabell 6 oppsummerer resultatene.

Bins	Co-occurrence	Spatioqram	Angle method	Histogram
4	1.18	0.29	0.27	0.19
8	1.18	0.41	0.27	0.20
16	1.18	1.43	0.27	0.34
32	1.26	10.45	0.27	1.17
64	1.32	101.2	0.27	9.30
128	1.59	1805	0.27	73.1
256	5.7	-	0.27	-

Tabell 6: Kjøretidsanalyse, ms

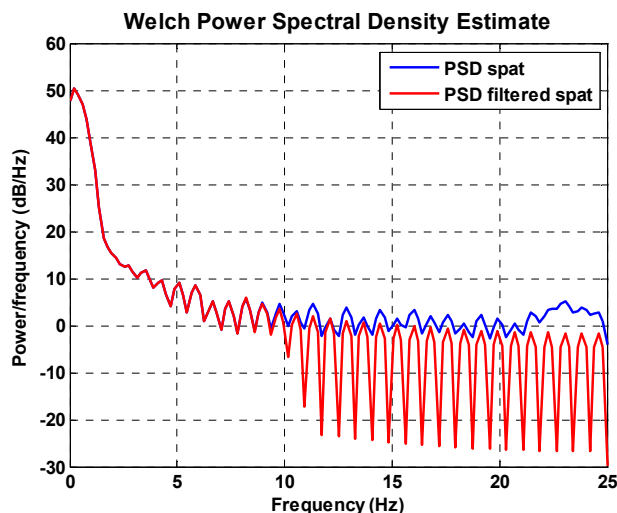
Man kan her se at vinkelmetoden er uavhengig av hvor mange farger man deler bildet inn i. Dette kommer av at metoden går ut på å regne ut vinkelen mellom to vektorer, og dimensjonen på disse vektorene er uavhengig av antall farger i bildet.

Siden co-occurrence metoden regner ut co-occurrence matrisene for R- G- og B-lagene hver for seg, er ikke denne metoden like følsom for antall farger som histogrammetoden og spatioqram-metoden. Datamengden for disse metodene øker kubisk med antall fager i hvert fargelag, og ved å benytte 256 farger i hvert lag vil datamaskinen gå tom for minne før den har klart å regne ut alle histogramverdiene.



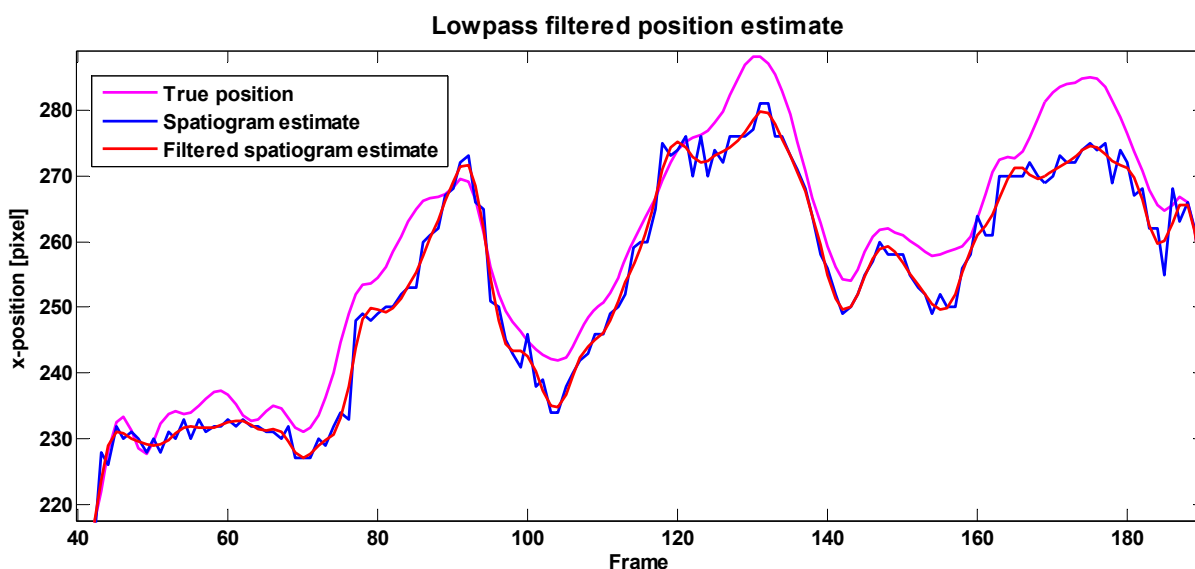
## 3.2 Støyfjerning

Ved å sende signalene gjennom filteret beskrevet i kapittel 2.8, vil man bli kvitt noe av den høyfrekvente støyen på posisjonsestimatene. Filteret demper signaler over 10 Hz.



Figur 40: Spektrum av filtrert posisjonsestimat

Figur 40 viser endringen på frekvensspektrumet til estimatene fra spatiogram-metoden ved bruk av filteret. Man ser her at frekvenser over 10 Hz er dempet, mens de under 10 Hz er uendret. Dette kommer også fram i figur 41, der man ser at det filtrerte signalet blir mye glattere, og ligner mer på den virkelige posisjonstrajektorien.



Figur 41: Tidsplot av posisjonsestimat med og uten filter

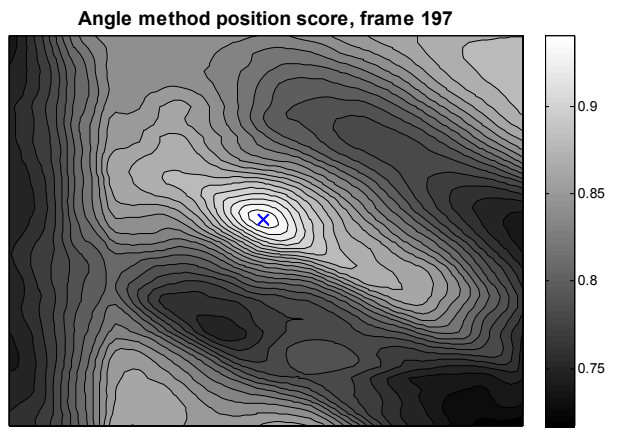
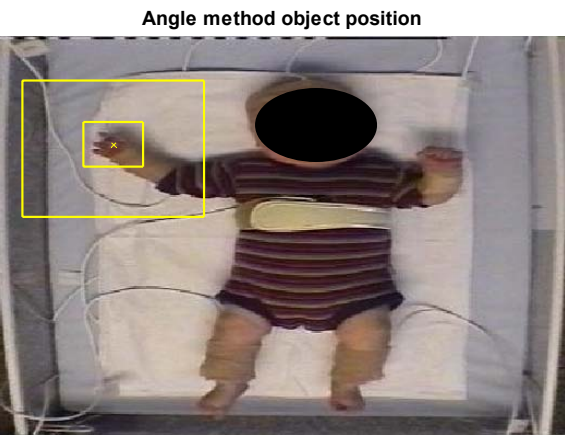
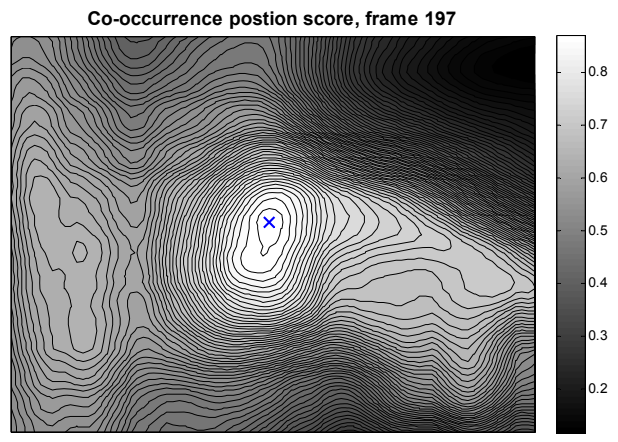
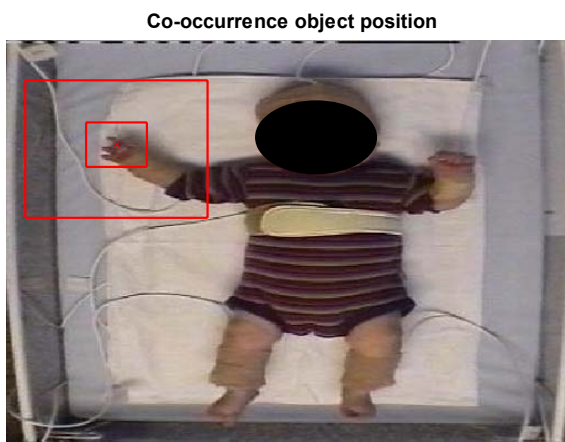
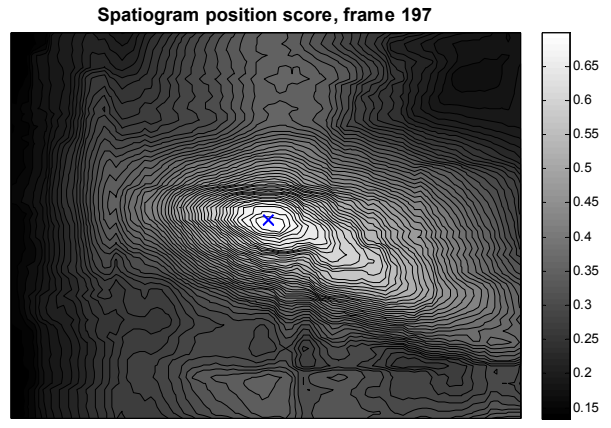
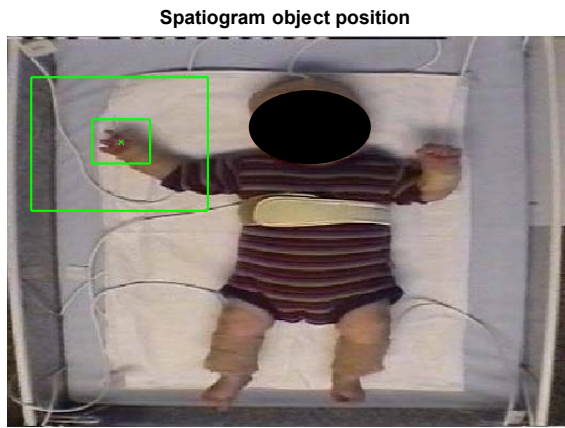
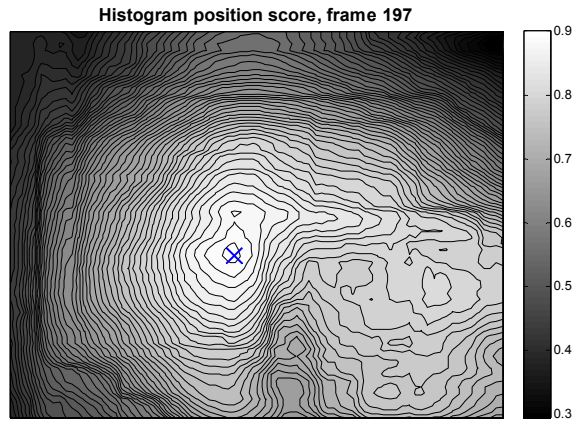
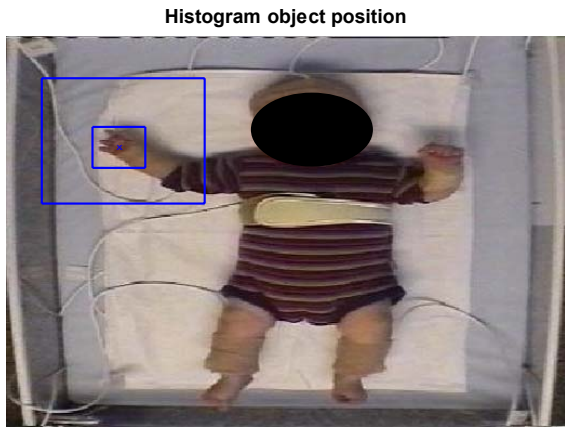
## 3.3 Følging av hender

Dette avsnittet beskriver resultatene fra følgingen av hendene. Det er foretatt tester der metodene er brukt hver for seg, og der de har samarbeidet om å gi et best mulig posisjonsestimert. Beskrivelsen av disse algoritmene er gitt i kapittel 2.6.

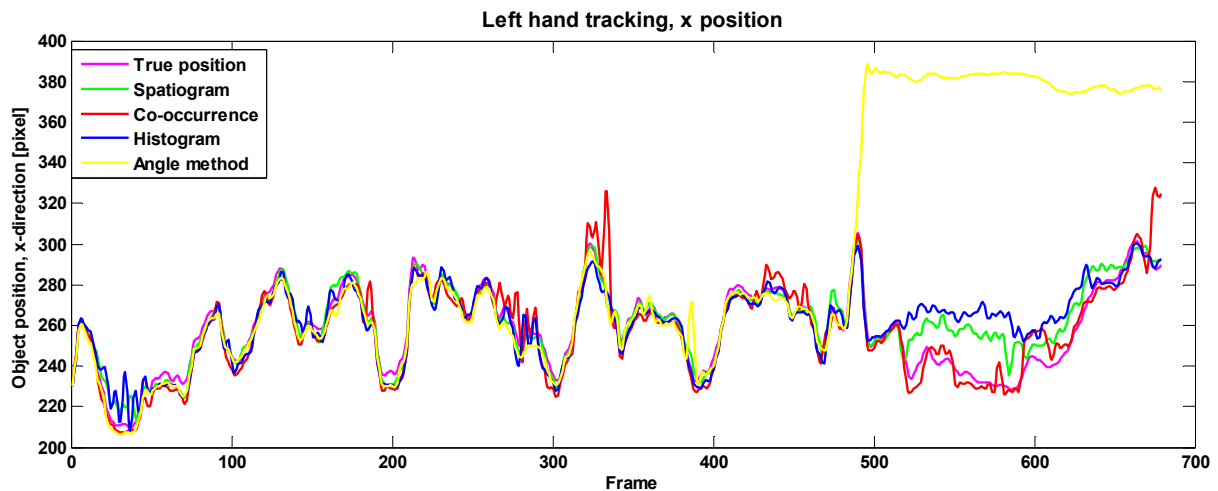
### 3.3.1 Metodene hver for seg

Figur 42 viser resultatene fra de forskjellige metodene kjørt på det samme datasettet med det samme referanseutsnittet. Figurene i venstre kolonne viser oversiktsbildet av barnet, søkeområdet og objektområdet. Til høyre ser man fordelingen av sammenligningsverdien innenfor søkeområdet.

Som man kan se fra figur 42 er alle metodene forholdsvis enige om hvor hånden befinner seg, og det er en tydelig topp i sammenligningsverdien for hver metode. Men man kan legge merke til sammenligningsverdien for spatiogram-metoden viser veldig tydelig hvor det er best likhet, mens de andre metodene ikke skiller så tydelig på hvor det er god og dårlig likhet.



**Figur 42: Metodenes posisjonsestimat ved bilde nr 197**



Figur 43: Tidsserie høyre hånd, metodene hver for seg

Figur 43 viser x-posisjonen i bildet for sekvensen der venstre hånd skal følges. Figuren viser at vinkelmetoden mister objektet etter 490 bilder, og kun co-occurrence metoden er god mellom bilde 500 og 600. For hvert bilde er det beregnet pikselavstanden mellom metodenes posisjonsestimat og den faktiske posisjonen. For hele sekvensen er det beregnet gjennomsnittlig avvik fra sann posisjon, samt standardavviket fra dette. Resultatene for dette er oppsummert i tabell 7.

	Left hand		Right hand	
	Mean	Std	Mean	Std
Spatiogram	5,97	5,63	3,46	1,94
Co-occurrence	6,42	5,28	6,73	6,82
Histogram	8,54	8,49	4,45	2,42
Angle method	43,38	62,72	61,12	70,30

Tabell 7: Oppsummering av resultatene for følging av hender

Se flere tidsgrafer for følging av hender i appendiks A.

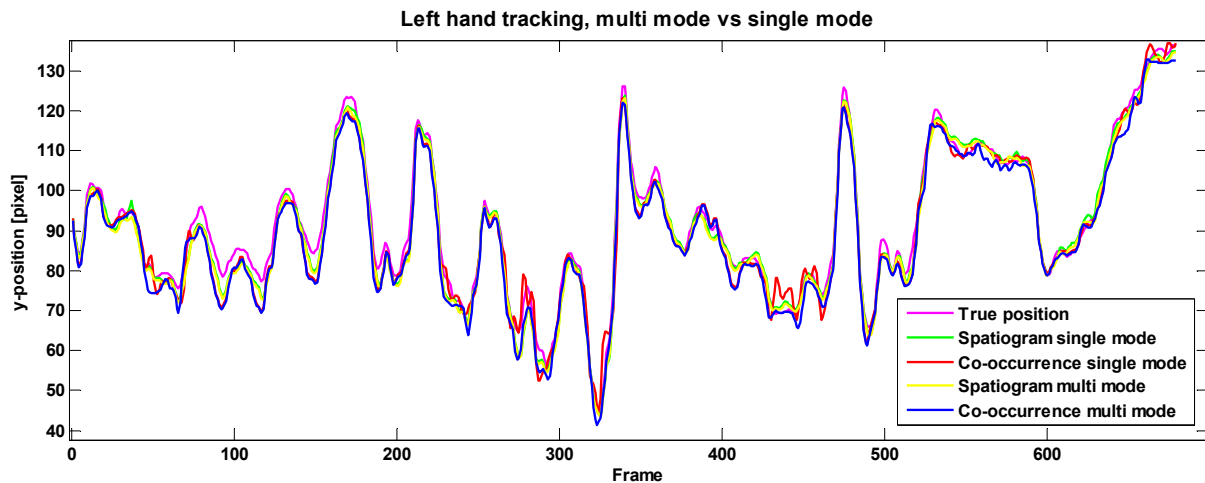
### 3.3.2 Metodene sammen

I dette avsnittet presenteres resultatene som er oppnådd når algoritmen velger posisjonsestimat som gir minst mulig omkrets. Denne algoritmen er beskrevet i kapittel 2.6.6.

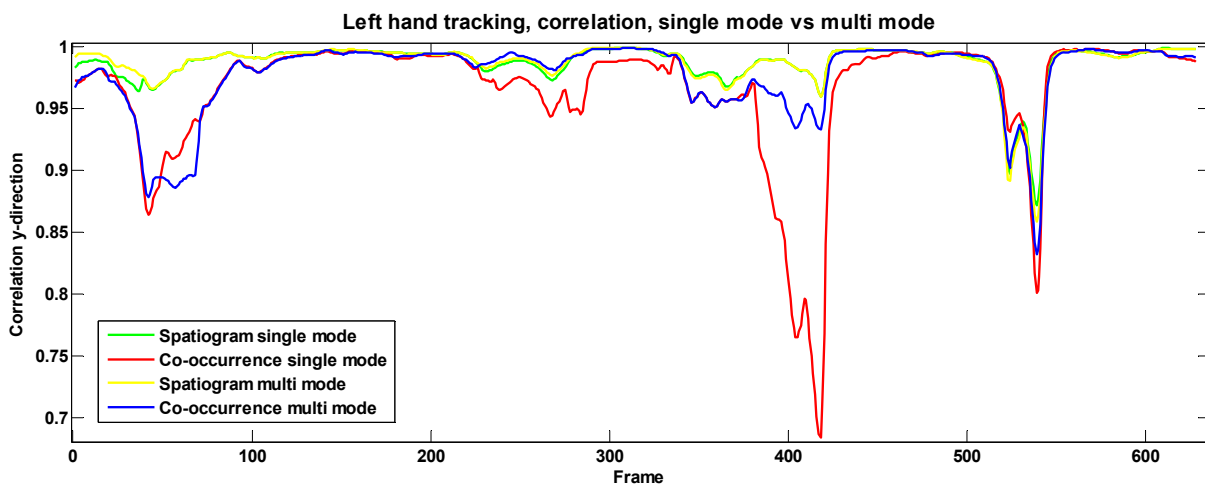
Figur 44 viser y-posisjonen, samt korrelasjonen mellom hver metode og den faktiske posisjonen. Det benyttes Pearsons korrelasjonskoeffisient, som beregnes mellom to vektorer på denne måten:

$$r_{xy} = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{(n-1) s_x s_y} = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2} \sqrt{n \sum_{i=1}^n y_i^2 - \left( \sum_{i=1}^n y_i \right)^2}} \quad (10)$$

Der  $\bar{x}$  og  $\bar{y}$  er gjennomsnittsverdien for henholdsvis  $X$  og  $Y$ , samt  $s_x$  og  $s_y$  er standardavviket til henholdsvis  $X$  og  $Y$ . Korrelasjonene er beregnet ved å ta et utsnitt på 50 elementer i posisjonsvektorene.



Figur 44: Posisjon i y-retning for flging av venstre hnd



Figur 45: Korrelasjon i y-posisjon for venstre hnd

Figur 44 og 45 viser en sammenligning av resultatene nr metodene kjrer hver for seg, og nr de blir samkjrt. Det kommer fram at spatioqram-metoden ikke endres i vesentlig grad, mens posisjonstrajektorien fra co-occurrence metoden blir forbedret. Dette kommer av at metoden for å finne beste felles posisjonsestimater utgangspunkt i de lokale maksimumspunktene for hver metode. Fordelingen av sammenligningsverdi for spatioqram-metoden er ofte slik at det kun eksisterer én topp, og dermed må denne bli valgt. Fordelingen

for co-occurrence og vinkelmetoden inneholder derimot ofte flere lokale maksima, slik at det optimale maksimumspunktet nødvendigvis ikke er det punktet med høyest verdi.

Tabell 8 oppsummerer forskjellen med og uten posisjonskorrigering.

	Left hand				Right hand			
	Single mode		Multi mode		Single mode		Multi mode	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Spatioqram	5,97	5,63	5,56	5,12	3,46	1,94	3,99	2,05
Co-occurrence	6,42	5,28	6,50	4,36	6,73	6,82	5,08	2,02
Histogram	8,54	8,49	6,54	5,78	4,45	2,42	4,82	2,08
Angle method	43,38	62,72	4,84	4,32	61,12	70,30	4,06	1,98

Tabell 8: Oppsummering målfølgning med og uten posisjonskorrigering

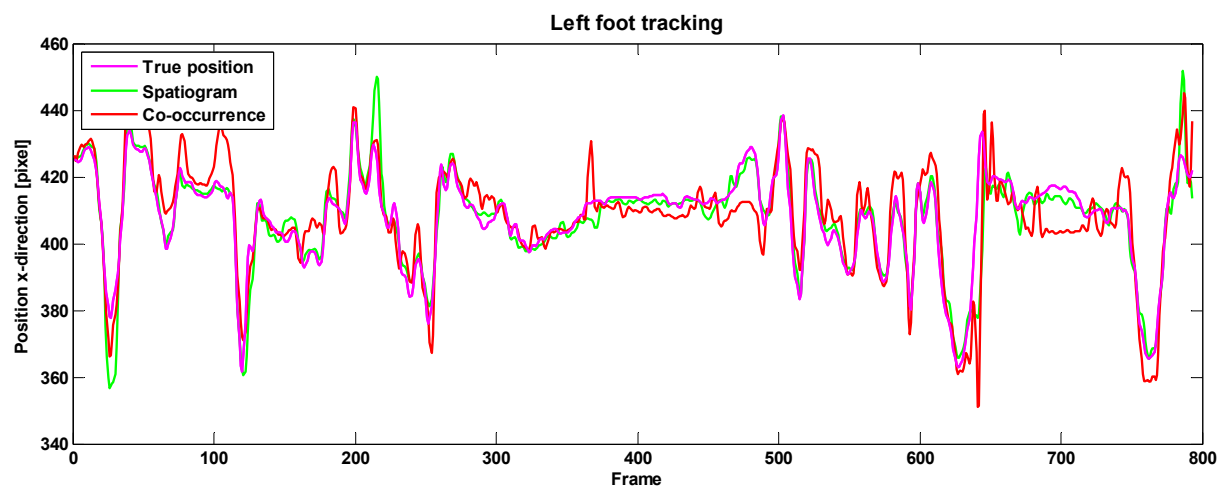
Se flere grafer i appendiks A.

### 3.4 Følging av føtter

I dette avsnittet presenteres resultatene fra følgningen av ungens føtter. Det er valgt en bildesekvens der føttene ofte går sammen og fra hverandre igjen. Dette er gjort for å vise problematikken dette skaper.

Det er benyttet den modifiserte algoritmen for å følge føttene, se kapittel 2.6.7.

Figur 46 viser tidsserien for venstre fot. Her velger metodene hver for seg hvilke referansemodeller de bruker. Tabell 9 oppsummerer serien ved å vise gjennomsnittlig avvik fra virkelig posisjon, samt standardavviket for hvor mye dette avviket varierer i tidsserien. Det fremkommer her at spatioqram-metoden fungerer best.

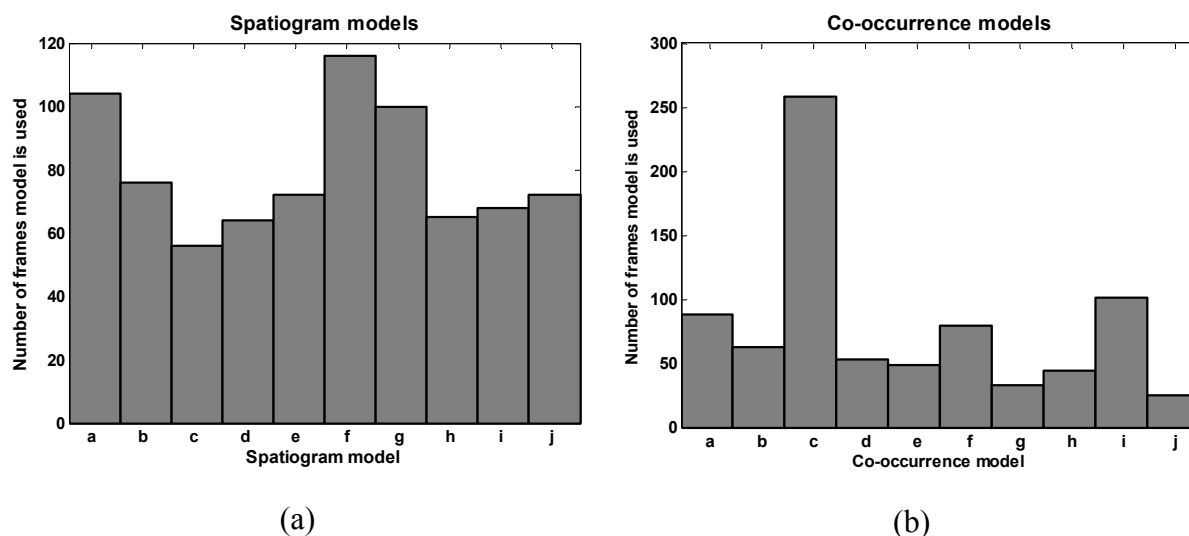


Figur 46: Tidsserie for følgning av venstre fot

	Left foot	
	Mean	Std
Spatioqram	4,20	5,17
Co-occurrence	8,19	6,41

Tabell 9: Gjennomsnittlig avvik fra virkelig posisjon for venstre fot

Figur 47 viser fordelingen av hvor ofte de ulike referansemodellene er benyttet i algoritmen. For hvert fjerde bilde i sekvensen blir det vurdert om det skal byttes referansemodell. Det er derfor bemerkelsesverdig at modell (c) (se figur 28) blir benyttet minst i spatioqram-metoden og mest i co-occurrence metoden.



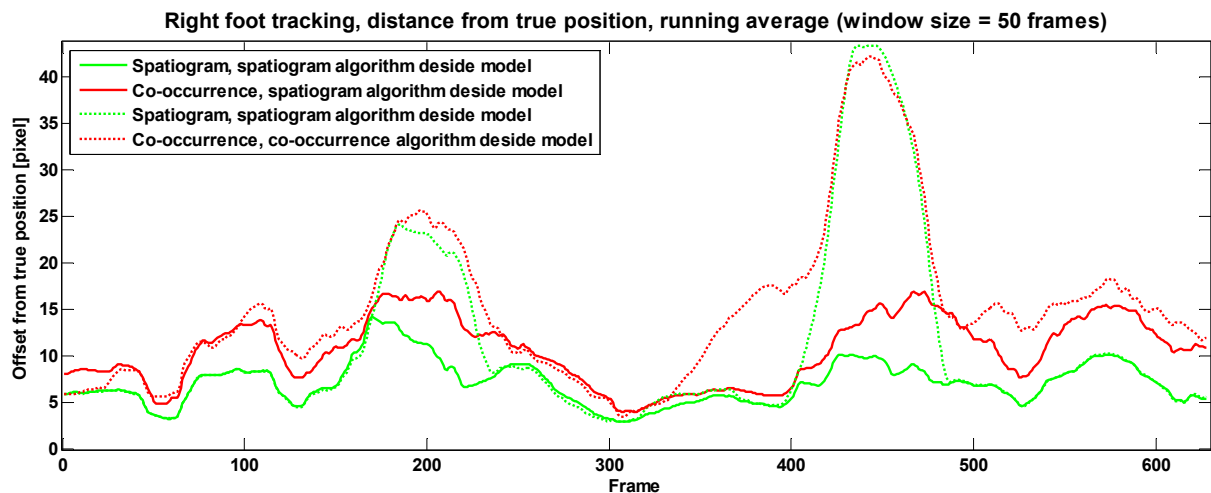
Figur 47: Oversikt over hvor ofte de forskjellige referansemodellene er brukt

Det kommer også fram at modellene er jevnere fordelt for spatioqram-metoden, mens for co-occurrence metoden er det mer ulik fordeling. Siden tabell 10 viser at spatioqram-metoden gir bedre resultat enn co-occurrence metoden, kan det tyde på at spatioqrammet er bedre egnet til å velge hvilken modellreferanse som skal benyttes. Det er derfor kjørt en test der co-occurrence metoden har brukt samme referansemodell som spatioqrammet til enhver tid. Dette har vist seg å gi et bedre resultat for begge metodene.

	Left foot				Right foot			
	Independent		Model from spat		Independent		Model from spat	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Spatioqram	4,20	5,17	3,76	3,98	10,17	14,51	6,77	6,05
Co-occurrence	8,19	6,41	6,33	4,99	14,24	13,57	10,23	7,64

Tabell 10: Oppsummering av resultater for følgende av føtter

Figur 48 viser løpende gjennomsnittsavvik fra den virkelige posisjonen.



Figur 48: Følging av venstre fot, avvik fra virkelig posisjon

Som figur 48 viser, vil begge metodene fungere best dersom spatioqramalgoritmen bestemmer hvilken referansemodell som skal benyttes. Når metodene velger referansemodell hver for seg, vil spatioqram-metoden velge samme modell som sist, mens co-occurrence metoden ofte velger en annen. Dette medfører at co-occurrence metodens estimat driver vekk, og genererer et posisjonskorrigeringskall når avstanden mellom de forskjellige metodene blir større enn 25 piksler. Denne grensen er satt som en indikasjon på at en av metodene ikke lenger følger rett objekt. Etter ca 400 bilder har co-occurrence metoden kommet så langt fra spatioqramestimatet at algoritmen aktiverer posisjonskorrigering. Dette innebærer optimalt valg av posisjonsestimater (beskrevet i kapittel 2.6.6). I denne situasjonen gjør det at spatioqramestimatet driver vekk fra objektet, og man ser at begge metodene får høyere avstand til den virkelige posisjonen. Det samme skjer også etter ca 100 bilder, men her blir resultatet at co-occurrence metoden blir dratt inn på rett objekt igjen.

Se appendiks B for flere grafer fra følging av føtter.

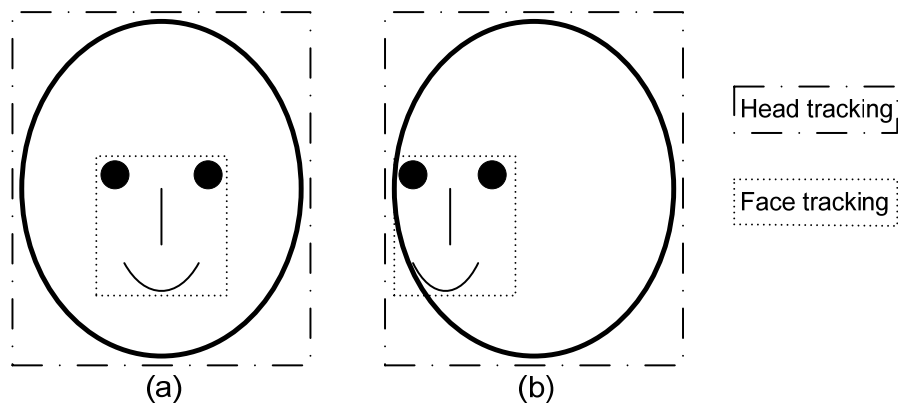
### 3.5 Følging av hode

I dette avsnittet vises resultatene fra følgingen av barnets hode.

Mye av ungens bevegelser av hodet er rene vridninger. Det betyr at hodets posisjon i bildet endres lite, mens ansiktets posisjon endrer seg mye fra side til side. Hvis man velger å følge hele hodet, vil ikke disse bevegelsene gi særlig stort utslag i forhold til å følge ansiktet. Denne situasjonen er vist i figur 49. I (a) illustreres en situasjon der ungen ser rett opp i kameraet, mens i (b) har ungen vridd hodet litt mot venstre. En algoritme som følger hele hodet vil si at objektet ikke har endret posisjon, mens en algoritme som følger ansiktet vil kunne detektere bevegelsen.

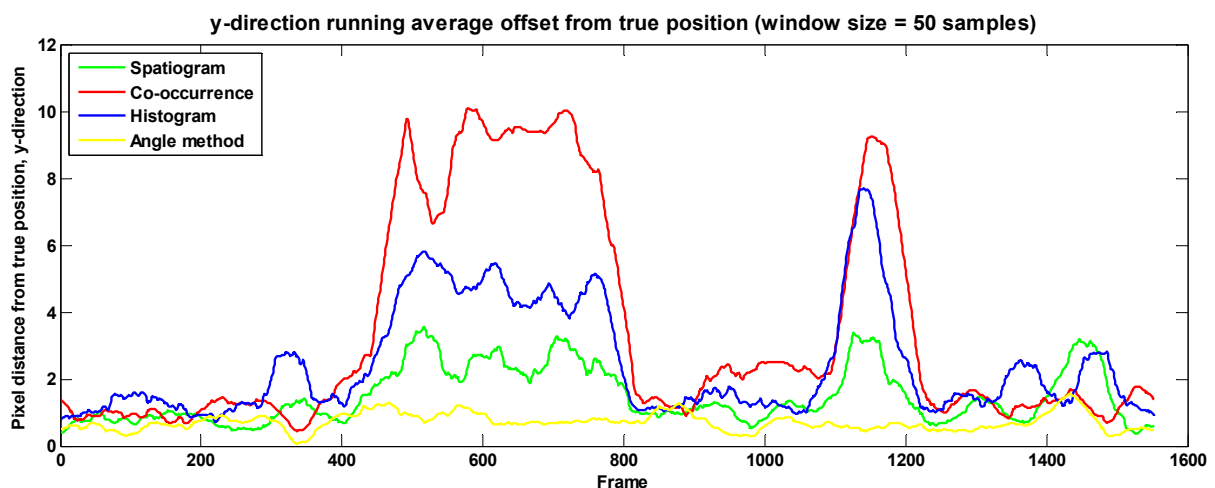


Som en følge av å velge et mindre referanseobjekt, vil også kjøretiden på algoritmene bli lavere.



Figur 49: Forskjellen på hode- og ansiktsfølging

På grunn av dette er det valgt å følge ansiktet til ungen. Ansiktet består av forholdsvis like farger, noe som gjør at metoder som er avhengig av å redusere antall farger i bildet ikke vil kunne klare å skille forskjellige objektkandidater godt fra hverandre. Siden fargefordelingen i ansiktet er ganske lik, vil metoden som baserer seg på co-occurrence matriser fungere dårlig i dette tilfelle. For ansiktsfølging har derfor metoden som baserer seg på cosinusvinkelen mellom bildeutsnittene vist seg å fungere best. Figur 50 viser at de andre metodene mister objektet to ganger i løpet av sekvensen, mens vinkelmetoden følger ansiktet hele veien.



Figur 50: Fortløpende avvik fra virkelig posisjon forfølging av ansikt

Siden ansiktet beveger seg sakte i forhold til bilderaten, kan man gjøre søkeområdet mindre enn beskrevet i kapittel 2.6. Man kan endre det fra å være tre ganger så langt i høyde og bredde som objektet, til å være 1.5 ganger høyde og bredde. Ved å halvere sidelengdene, blir antall mulige objektkandidater redusert med  $\frac{3}{4}$ , noe som fører til raskere kjøretid.

Tabell 11 oppsummerer resultatene fra ansiktsfølgingen. Vinkelmetoden hadde minst gjennomsnittlig avvik i posisjonen, og også minst varians. Dette betyr at det var den metoden som var mest stabil, og som traff best. Se flere grafer i Appendiks C.

	Face tracking	
	Mean	Std
Spatioqram	3,57	2,53
Co-occurrence	6,60	2,78
Histogram	3,74	2,70
Angle method	2,77	1,30

Tabell 11: Oppsummering av ansiktsfølging

### 3.6 Følging av bryst

Siden brystet på ungene ofte er ensfarget, kan det være problematisk å finne et område som godt lar seg følge. Et godt objektområde er det som klart kan skille seg ut i fra omgivelsene, enten ved farge, form eller mønster. Siden ungene som regel har på seg klær som er ensfarget, eller ikke har på seg klær i det hele tatt, vil det være vanskelig å skille objekt kandidatene fra hverandre. Lysforhold og skygger vil være avgjørende for drift underveis i sekvensen.

Barna har festet en magnetisk posisjonssensor rundt brystet. Men denne er også ensfarget, og danner kun en horisontal linje med klærne til ungen. Objekt kandidater på høykant vil kunne skille seg fra hverandre, mens kandidater langs denne linjen vil se helt like ut for følgingsalgoritmene.

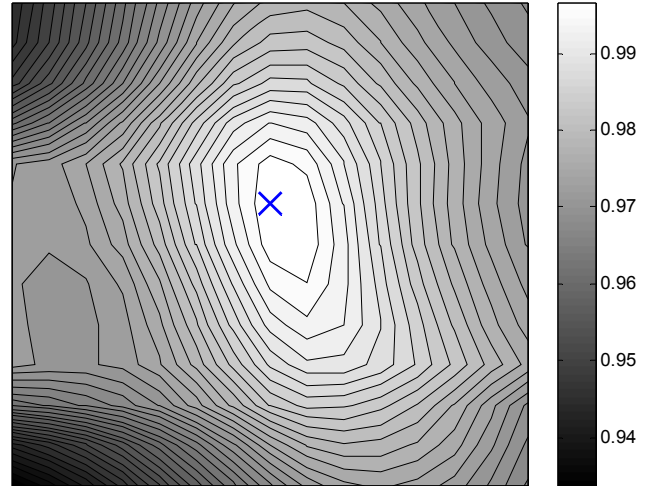
Testing har vist at festesonen til den magnetiske sensoren på ungens bryst er det beste stedet for målfølging. Se figur 51. Her skilles objekt kandidatene godt både horisontalt og vertikalt. Siden dette området ikke endrer form og farge i særlig stor grad, vil det her lønne seg å bruke vinkelmetoden. Fordelene ved denne metoden er at algoritmen bruker bildeutsnittet direkte, og gjør ingen forenklinger. I motsetning til dette, vil metodene som er basert på co-occurrence og histogrammer dele bildet opp i mindre farger enn originalt for å redusere dimensjonen på regneproblemet.

Siden co-occurrence metoden ser på fordelingen av farger blant nabopiksler, vil alle kandidatene langs kanten der hvor beltet festes få forholdsvis lik score, og posisjonsestimatet vil variere fra bilde til bilde. Dette resulterer i en trajektorie som er støyfull i forhold til den egentlige bevegelsen. Tilsvarende resultater får man også ved å bruke metodene som baserer seg på histogrammer. Se grafer i Appendiks D.

**Object area and search area**



**Position score**



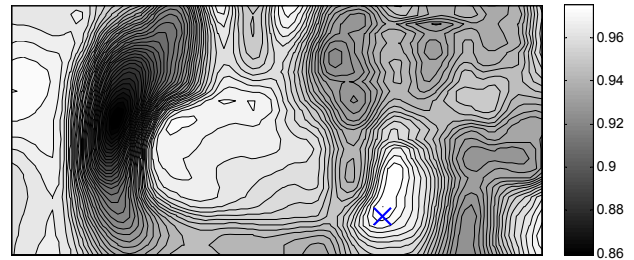
**Figur 51: Søkeområdet og posisjonsverdier for følging av bryst**

Et problem ved følging av brystet er at armene kan komme i veien og blokkere synsfeltet for algoritmen. Dette vil gi utslag i at den beste objekt kandidaten får vesentlig lavere sammenligningsverdi enn forventet. Som en løsning på dette problemet vil algoritmen utvide søkeområde, samt holde det konstant over alle bildene, helt til den igjen oppdager en objekt kandidat som gir høy nok sammenligningsverdi til at man kan forvente at objektet igjen er synlig. Figur 52 viser algoritmens søkeområde ved en situasjon der den har mistet objektet.

**Lost track, big search area**



**Position score**



**Figur 52: Okklusjon ved følging av bryst**

Tabell 12 viser resultatene fra de forskjellige algoritmene fram til det punktet der brystet blir dekket til for første gang. Man ser her at vinkelmetoden gir best og minst varierende resultat.

	To first occlusion		Whole sequence	
	Mean	Std	Mean	Std
Spatioqram	2,26	2,07		
Co-occurrence	5,50	2,21		
Histogram	6,06	7,39		
Angle method	1,52	0,70	1,63	0,84

**Tabell 12: Oppsummering av resultatene for følging av bryst**

## 4 Konklusjon, diskusjon og videre arbeid

### 4.1 Resultater

Det er utført vellykket målfølging av barns hender, føtter, hode og bryst. Dette er gjort uten noen form for optiske markører. Algoritmene som er benyttet er i en viss grad spesialtilpasset de forskjellige kroppsdelene, men metodene er de samme.

Siden hendene og føttene beveger seg mye, trengs robuste metoder for å kunne følge disse. For hendene er det benyttet en posisjonsalgoritme som baserer seg på et optimalt valg av posisjonsestimat fra de forskjellige metodene. For føttene benyttes flere referansemodeller slik at algoritmen kan kjenne igjen en fot selv om den har endret form og farge. Spatiogrammet har vist seg å være den beste måten å representere et objekt på. En videreutvikling av spatiogram-metoden ved bruk av flere referansemodeller vil sannsynligvis kunne gi bedre følging av hendene også.

For hodet og brystet er vinkelmetoden benyttet. Denne er god på å skille tilsynelatende like områder fra hverandre.

### 4.2 Diskusjon

Metodene som er brukt i dette prosjektet har vist seg å være gode når følgingsområdet opprettholder samme form og farge gjennom sekvensen. Ved å innføre flere referansemodeller er algoritmen forbedret med hensyn på å kunne takle endringer i form og farge. Likevel er det enkelte problemområder der det fortsatt er mye som kan forbedres.

#### **Okklusjon**

Okklusjon er når målobjektet forvinner bak andre områder. Dette forekommer når armene kommer i veien for brystet eller hode, eller når et bein dekker over den andre foten. I algoritmen som er presentert i denne rapporten, vil okklusjon merkes når sammenligningsverdien når under en kritisk grense. Da antas det at man ikke lenger følger objektet, og søkeområdet utvides og holdes i konstant posisjon. Algoritmen vil fortsette som vanlig når den finner en objekt kandidat som har tilfredsstillende sammenligningsverdi. På dette området bør det gjøres mer arbeid for at algoritmen skal bli bedre.

#### **Bytte av hender og føtter**

Føttene beveger seg ofte mot hverandre. Når de beveger seg fra hverandre igjen er situasjonen ofte slik at venstre og høyre fot ligner på hverandre. Algoritmen kan dermed ende opp med å

følge feil fot. I algoritmen som er utviklet vil dette detekteres ved at posisjonsestimatet for venstre fot befinner seg lengst til høyre, og motsatt. Ved å utvide søkeområdet og forkaste den høyre objektkandidaten, vil algoritmen som oftest hoppe tilbake til riktig fot. Men det er ikke alltid at dette er korrekt, for eksempel hvis ungen krysser føttene. Derfor bør det utvikles en mer robust måte å detektere om algoritmen følger feil objekt. Problematikken er tilsvarende for hendene.

## 4.3 Forslag til forbedringer

### 4.3.1 Datagrunnlag

Ved å endre litt på opptaksmetodikken, vil sannsynligvis algoritmene for målfølgning kunne gi et bedre resultat. Lyssetting, fokusering og bakgrunn er enkle grep som vil gjøre problemstillingen enklere. Ved å belyse området rett ovenfra vil man unngå å skape skygger som gjør at målet endrer farge underveis i sekvensen. Ved å ha en homogen bakgrunn vil det være enklere å skille ut barnet fra bakgrunnen, og på denne måten redusere antall mulige posisjonskandidater. Ved å bedre fokus og oppløsning vil man kunne få et klarere skille mellom barnet og bakgrunnen, og kantdeteksjon vil kunne fungere bedre.

### 4.3.2 Nye metoder

Her presenteres noen metoder som kan være med på å forbedre målfølgingsalgoritmen.

#### **Bevegelsesdeteksjon**

For lettere å kunne avgjøre hendenes og føttenes posisjon kan man benytte seg av metoder som oppfatter bevegelser i videostrømmen. Den helt enkleste måten å gjøre dette på er å se på differansen mellom to påfølgende bilder. Da vil de områdene som har vært i ro bli svarte, mens der det er bevegelse vil det bli en lysere farge. Hvis ungen beveger seg sakte i forhold til bilderaten, vil ungens konturer komme til syne. Hvis ungen derimot beveger seg for fort, vil store deler av bevegelsesområde endre farge, og metoden vil ikke være like nyttig.

Optical flow kan også benyttes som bevegelsesdeteksjon. Da vil man registrere forflytningen til alle pikslene i bildet.

**Motion detection, slow movements**



(a)

**Motion detection, fast movements**



(b)

**Figur 53: Resultat for bevegelsesdeteksjon ved sakte bevegelser (a) og rasker bevegelser (b)**

### **Kantdeteksjon**

Ideen med kantdeteksjon er mye den samme som for bevegelsesdeteksjon, nemlig å kunne finne omrisset til hendene og føttene. Fordelen med kantdeteksjon fremfor bevegelsesdeteksjon er at man ikke trenger å ha bevegelse i bildet for å finne konturen. Denne konturen kan fungere som en grense for søkeområdet, og sammen med andre metoder vil den kunne gi et bidrag til å gjøre følgingsalgoritmen mer robust. Figur 54 viser et utsnitt av venstre hånd, og resultatet etter at kantdeteksjonen er utført. Canny kantdeteksjon (Canny, 1986) er en ferdig implementert funksjon i MATLAB.

**Canny edge detector**



**Figur 54: Kantdeteksjon brukes for å finne venstre hånd**

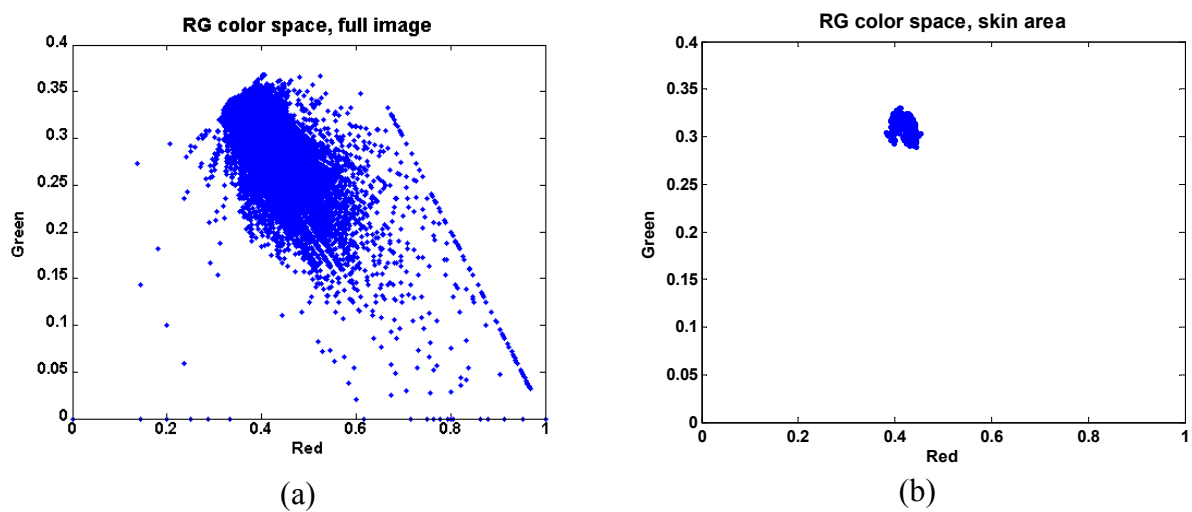
### **Huddeteksjon**

Hudfargen kan brukes for å avgjøre hvor hendene og føttene befinner seg i bildet. Man kan da benytte seg av en metode som filtrerer bildet på bakgrunn av fargen. Yang et al. (1996) viser at man kan oppnå gode resultater kun ved å se på rød- og grønnfargen i bildet. De viser også at forskjellen i menneskers hudfarge henger mer sammen med intensiteten i fargene, og ikke

fargene i seg selv. Derfor er det brukt et fargerom som normaliserer fargene ved å dele på den totale intensiteten.

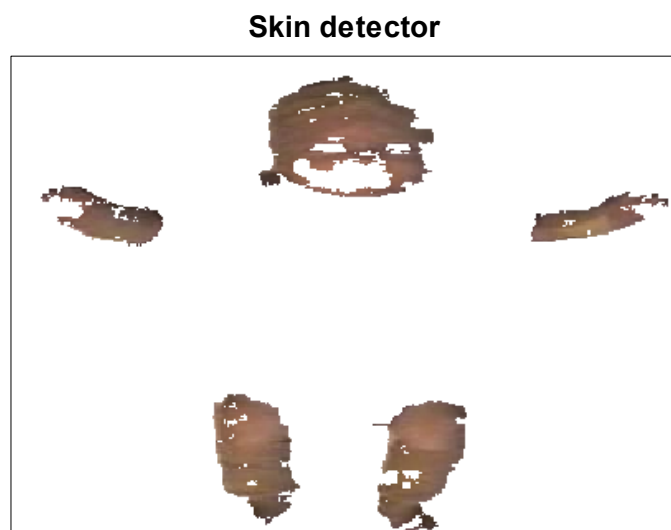
$$\begin{aligned}
 I_{i,j} &= R_{i,j} + G_{i,j} + B_{i,j} \\
 r_{i,j} &= \frac{R_{i,j}}{I_{i,j}} \\
 g_{i,j} &= \frac{G_{i,j}}{I_{i,j}}
 \end{aligned}
 \tag{11}$$

I figur 55 er det vist fordeling av rød-grønne fargeverdier i et bilde av en unge. Figuren til venstre viser fordelingen av hele bildet, mens figuren til høyre viser fordelingen av de områdene som er hudfarget. Figuren viser at hudfarge er i piksler som har grønnverdi i området  $[0.29 \ 0.33]$  og rødverdi i området  $[0.38 \ 0.44]$



Figur 55: Rød-grønn fargefordeling i hele bildet (a) og i områder med hud (b)

Figur 56 viser resultatet av filtreringen når det kun vises piksler som har rød- og grønnfarge i området  $[0.29 \ 0.33]$  og  $[0.38 \ 0.44]$ .



Figur 56: Resultat av huddeteksjon



## Bibliografi

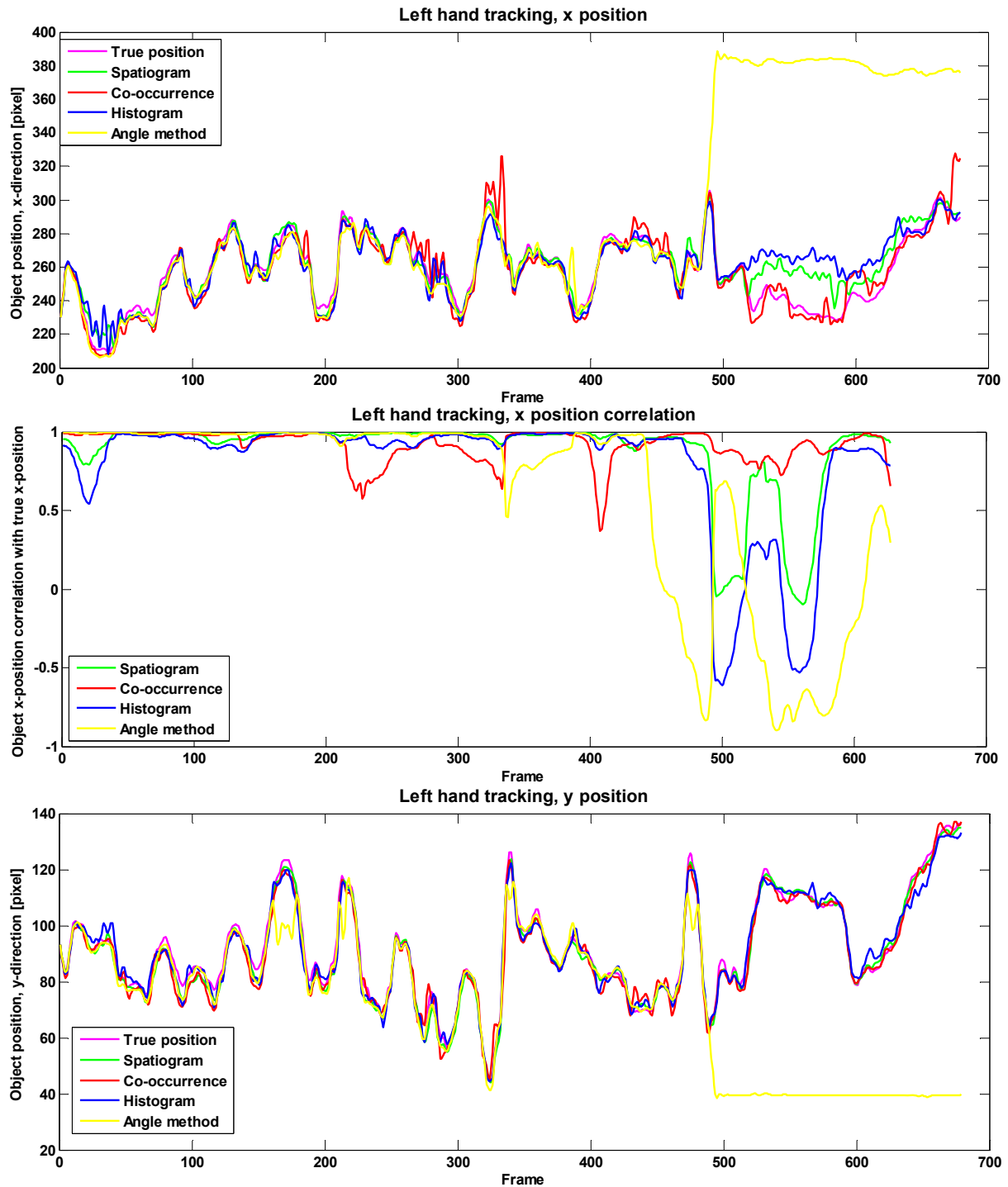
- [1] **A. Bhattacharyya.** *On a measure of divergence between two statistical populations defined by probability distributions.* Bull. Calcutta Math. Soc., vol. 35, pp. 99–109, 1943.
- [2] **S.T. Birchfield.** *Elliptical Head Tracking Using Intensity Gradients and Color Histograms.* IEEE Conference on Computer Vision and Pattern Recognition, 1998.
- [3] **S.T. Birchfield, S. Rangarajan.** *Spatiograms versus Histograms for Region-Based Tracking.* Computer Vision and Pattern Recognition. Vol. 2, 2005.
- [4] **S.T. Birchfield, S. Rangarajan.** *Spatial Histograms for Region-Based Tracking.* ETRI Journal. Vol. 29, No. 5, 2007.
- [5] **M. Borchani og G. Stamon.** *Texture features for image classification and retrieval.* SPIE Multimedia Storage and Archiving Systems, Vol. 3229, p. 401-406, 1997.
- [6] **J. Canny.** *A Computational Approach To Edge Detection,* IEEE Trans. Pattern Analysis and Machine Intelligence, s 679-714, 1986.
- [7] **S. Chin, M. Choo.** *Moving Vehicle Tracking for the Web Camera.* International Conference on Web Engineering, 2003.
- [8] **S. Choi, Y. Seo, H. Kim, K.S. Hong.** *Where are the ball and players? Soccer game analysis with color-based tracking and image mosaic.* International Conference on Image Processing, 1996.
- [9] **D. Comaniciu, R. Visvanathan.** *Kernel-Based Object Tracking.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003.
- [10] **C.O. Conaire, N.E. O'Connor, A.F. Smeaton,** *An Improved Spatiogram Similarity Measure for Robust Object Localisation.* Acoustics, Speech and Signal Processing. 2007.
- [11] **C.O. Conaire, N.E. O'Connor, A.F. Smeaton.** *Thermo-Visual Feature Fusion for Object Tracking Using Multiple Spatiogram Trackers.* Journal of Machine Vision and Applications. May, 2007.
- [12] **Y. Du, J. Chrisman.** *A Color Projection for Fast Generic Target Tracking.* IEEE/RSJ International Conference on Intelligent Robots and Systems, 1995.
- [13] **K.T. Fung, N.F. Law, W.C. Siu.** *Region-based Object Tracking for Multipoint Video Conferencing using Wavelet Transform.* Department of Electronic and information Engineering, The Hong Kong Polytechnic University, Hong Kong, Kina, 2001.
- [14] **S. Haker, A. Tannenbaum, G. Sapiro, D. Washburn.** *Missile Tracking Using Knowledge-Based Adaptive Thresholding.* International Conference on Image Processing, 2001.
- [15] **H.W. Hseu, A. Bhalerao, R. Wilson.** *Image Matching Based On The Co-occurrence Matrix.* Department of Computer Science, University of Warwick, 1999.

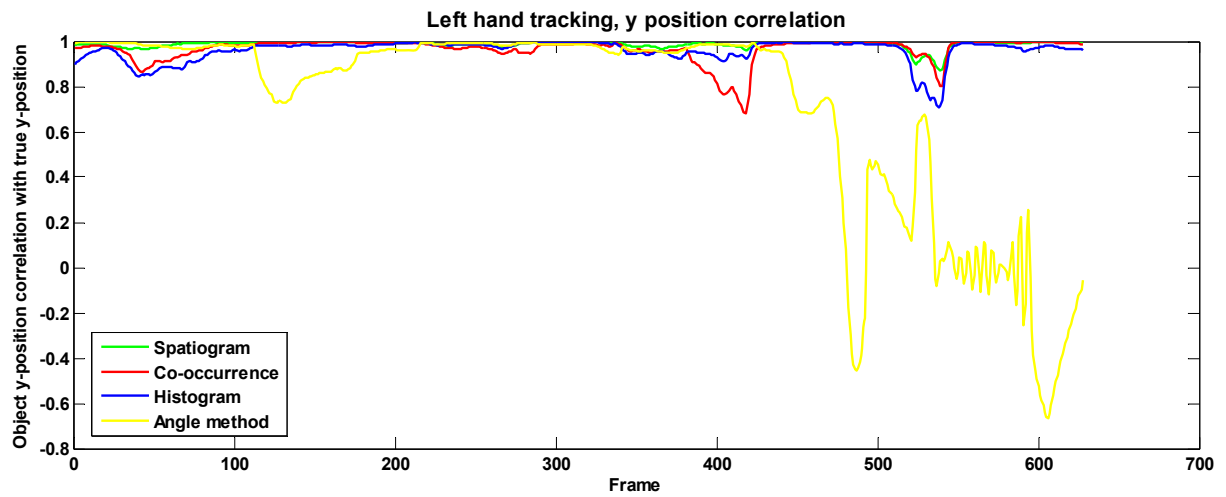
- [16] **R. K. Hammon.** *Improvements to the Gray-Level Co-occurrence Matrix (GLCM) Based Trackability Metric*, Missile Research, Development and Engineering Center, US Army, Alabama, 2002
- [17] **S. Iwase, H. Saito.** *Tracking soccer players based on homography among multiple views*. Department of Information and Computer Science, Keio University, Yokohama, Japan, 2003.
- [18] **R.E. Kahn, M.J. Swain, P.N. Prokopowicz, R.J. Firby.** *Gesture recognition using the Persus architecture*. Computer Vision and Pattern Recognition, 1996.
- [19] **K. Kim, S. Jeong, B.T Chun, J.Y Lee, Y. Bae.** *Efficient video images retrieval by using local co-occurrence matrix texture features and normalised correlation*. TENCON 99. Proceedings of the IEEE Region 10 Conference. Vol. 2, 1999.
- [20] **F. Lange, G. Hirzinger.** *Predictive Visual Tracking of Lines by Industrial Robots*. The International Journal of Robotics Research, 2003.
- [21] **M. Lucena, J.M. Fuertes, N. Perez de la Blanca.** *Using Optical Flow for Tracking*. Progress in Pattern Recognition, Speech and Image Analysis, 2003
- [22] **Norsk Barnelegeforening.** *Cerebral Parese*. <http://www.legeforeningen.no/index.gan?id=93683>, mai 2008.
- [23] **K.A. Mann, F.W. Werner, A.K. Palmer.** *Frequency spectrum analysis of wrist motion for activities of daily living*. Journal of Orthopaedic Research, 1989.
- [24] **A.G. Med-World.** *Infantile Zerebralparese*. 2002
- [25] **J. Ohya, A. Utsumi, J. Yamato.** *Analyzing Video Sequences of Multiple Humans*. Kluwer Academic Publishers, Boston, USA, 2002.
- [26] **D. Prewer, L. Kitchen.** *A Simple Fast Edge-Based Visual Tracker*. Department of Computer Science, The University of Melbourne, Australia, 1997.
- [27] **H.F. Prechtl.** *General Movement Assessment as a method of developmental neurology: new paradigms and their consequences*. Developmental Medicine & Child Neurology, 2001
- [28] **H.F. Prechtl, C. Einspieler, G. Cioni, A.F. Bos, F. Ferrari, D. Sontheimer.** *An early marker for neurological deficits after perinatal brain lesions*. The Lancet, 349(9062):1361–1363, 1997
- [29] **C. Rasmussen, K. Toyama, G.D. Hager.** *Tracking Objects By Color Alone*. Department of Computer Science, Yale University, 1996
- [30] **N. T. Siebel.** *Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications*. Department of Computer Science, The University of Reading, UK, 2003.
- [31] **N. T Siebel, S. Maybank.** *Real-Time Tracking of Pedestrians and Vehiles*. IEEE Workshop on Performance Evaluation of Tracking and Surveillance, 2001.

- [32] **O.M. Staal.** *Robust Video Based Motion Tracking in Young Infants*. Master thesis, Norwegian University of Science and Technology, 2006.
- [33] **J.D. Still.** *Skin detection*. [http://www.urwelcome.us/Jstill/Matlab/Skin\\_Detection.htm](http://www.urwelcome.us/Jstill/Matlab/Skin_Detection.htm), mai 2008.
- [34] **P. Stoica, R.L. Moses.** *Introduction to Spectral Analysis*. Prentice-Hall, 1997.
- [35] **H. Wang, D. Suter, K. Schindler, C. Shen.** *Adaptive Object Tracking Based on an Effective Appearance Filter*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 29, No. 9, 2007.
- [36] **H. Wang, D. Suter.** *Efficient Visual Tracking by Probabilistic Fusion of Multiple Cues*. IEEE Computer Society. Vol. 4, 2006.
- [37] **H. Wang, D. Suter, K. Schindler.** *Effective Appearance Model and Similarity Measure for Particle Filtering and Visual Tracking*. European Conference on Computer Vision. 2006.
- [38] **P. Welch.** *The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms*. IEEE Transactions on Audio Electroacoustics, Vol AU-15, June 1967
- [39] **M. Xu, R. Niu, P.K Varshney.** *Detection and Tracking of Moving Objects in Image Sequences with varying Illumination*, IEEE Conference on Image Processing, 2004.
- [40] **J. Yang, A. Waibel.** *A real-time face tracker*. Proceedings of the 3th IEEE Workshop on Applications of Computer Vision, Sarasota, Florida, s 142-147, 1996
- [41] **A. Zhao.** *Robust Histogram-based Object Tracking in Image Sequences*. Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, 2007.
- [42] **G. Zhu, Q. Zeng, C. Wang.** *Efficient Edge-based Object Tracking*. Pattern Recognition, Vol 39, 2006.
- [43] **Z. Zivkovic, B. Kröse.** *An EM-like algorithm for color-histogram-based object tracking*. IEEE Conference Computer Vision Pattern Recognition, 2004.
- [44] *What is Deinterlacing? Facts, solutions, examples*. <http://www.100fps.com>, mai 2008.

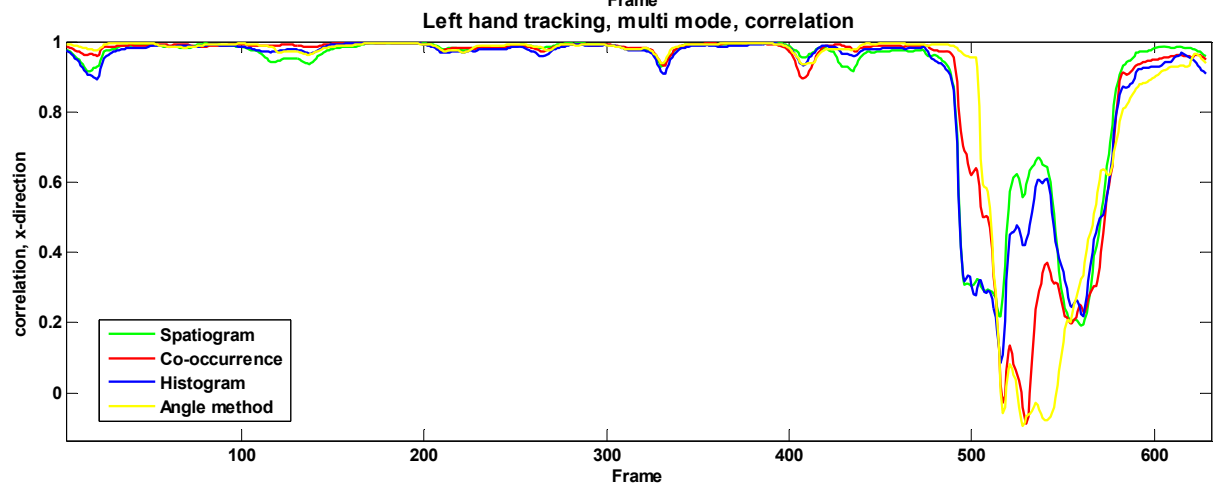
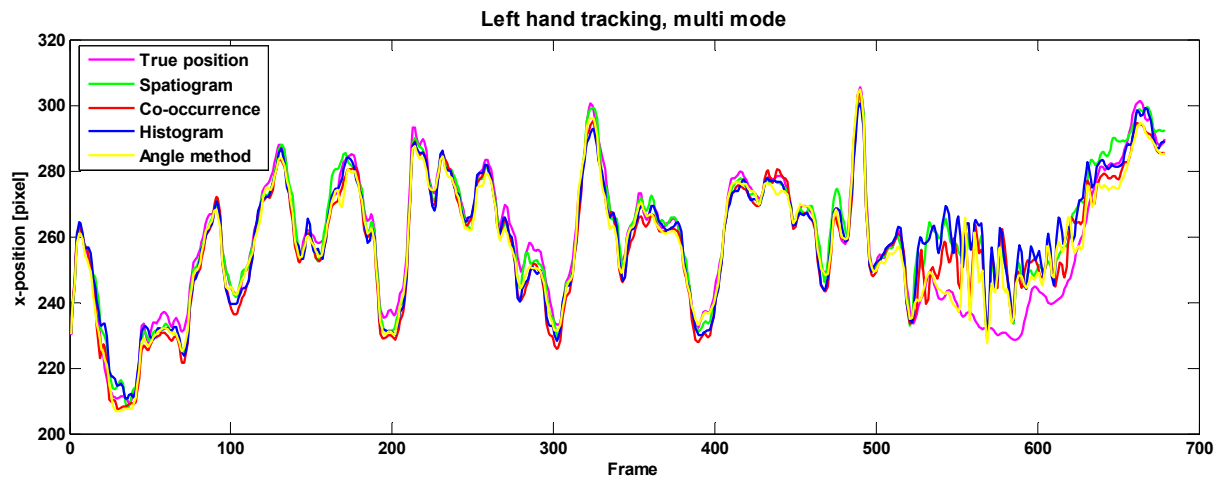
# Appendiks A Grafer fra følging av hender

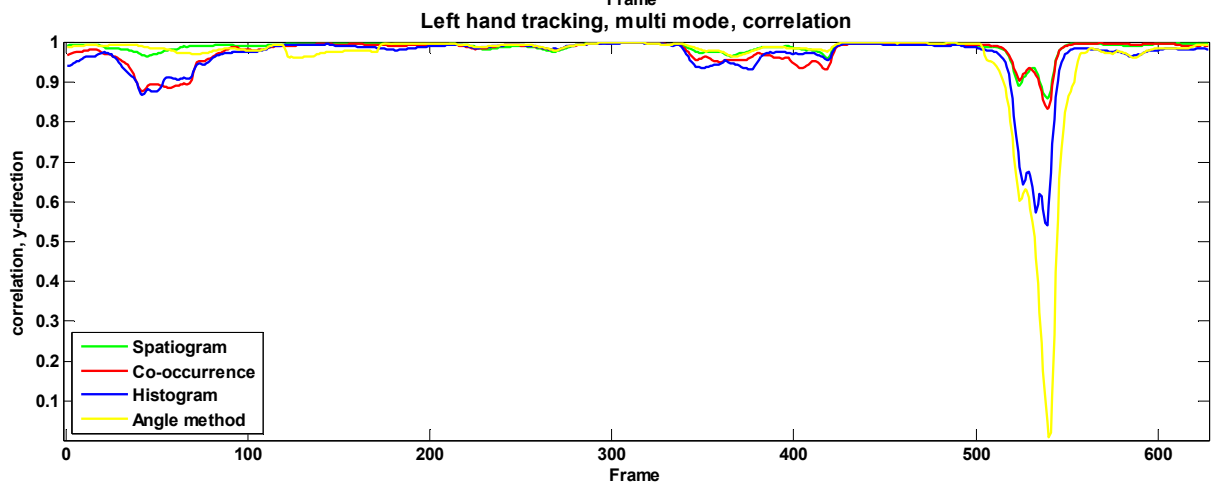
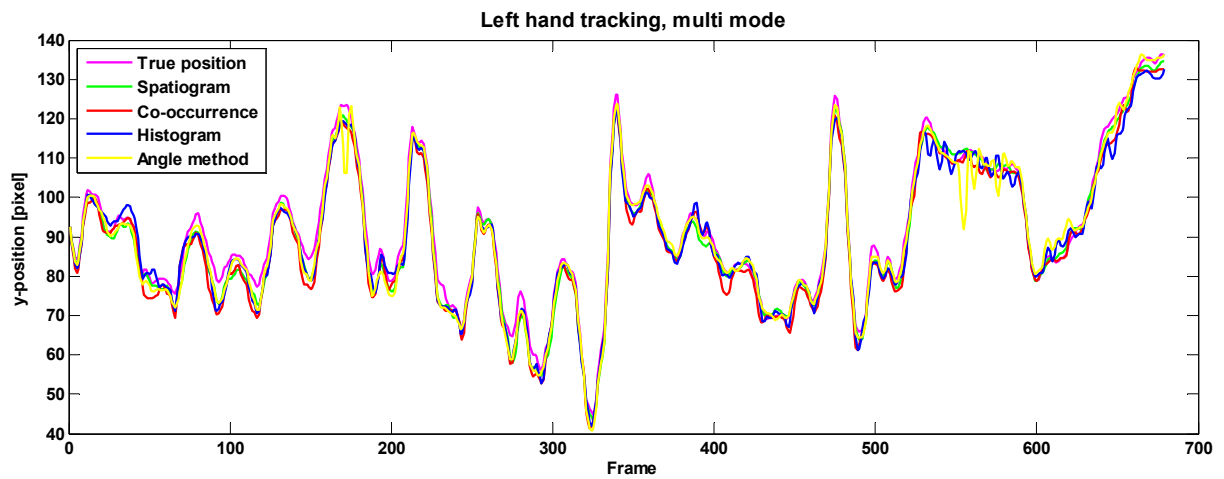
Venstre hånd, metodene hver for seg:



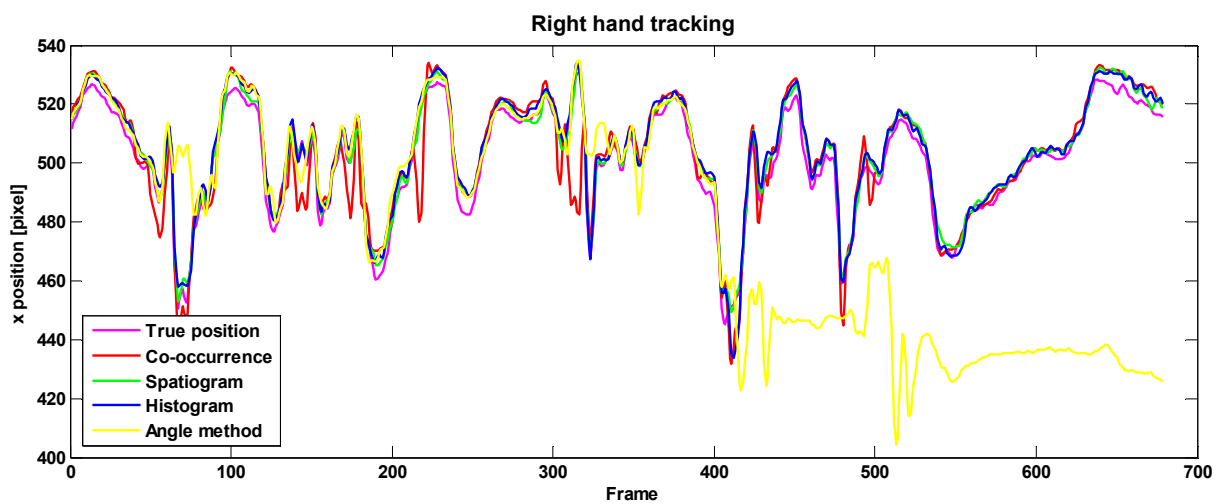


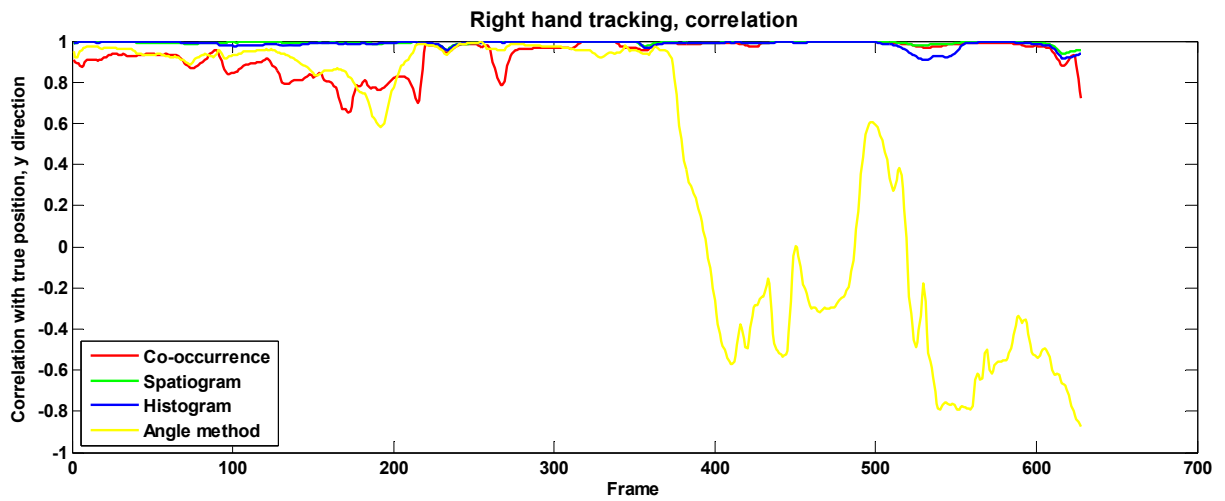
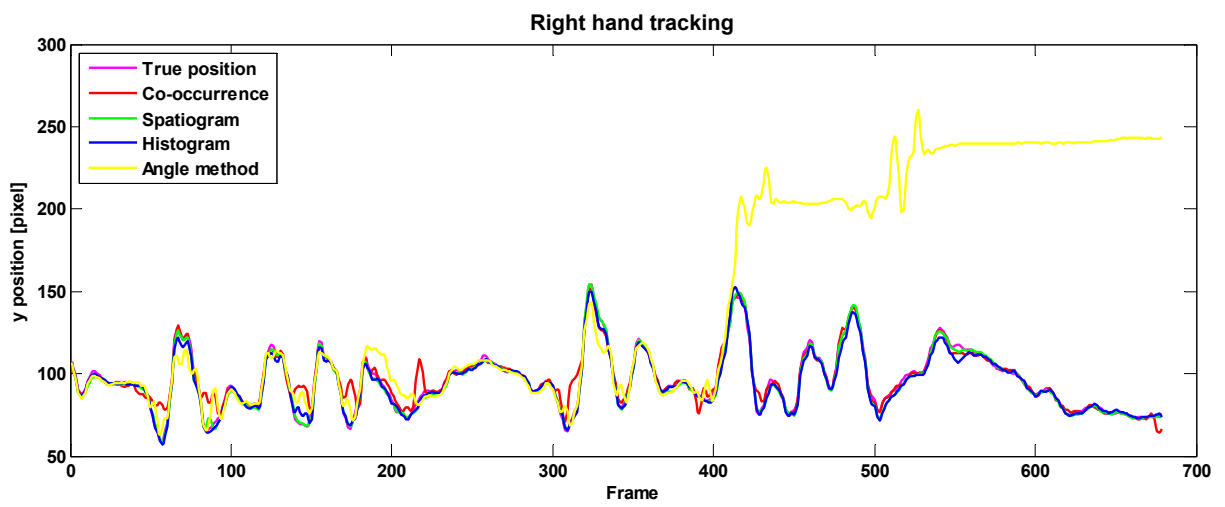
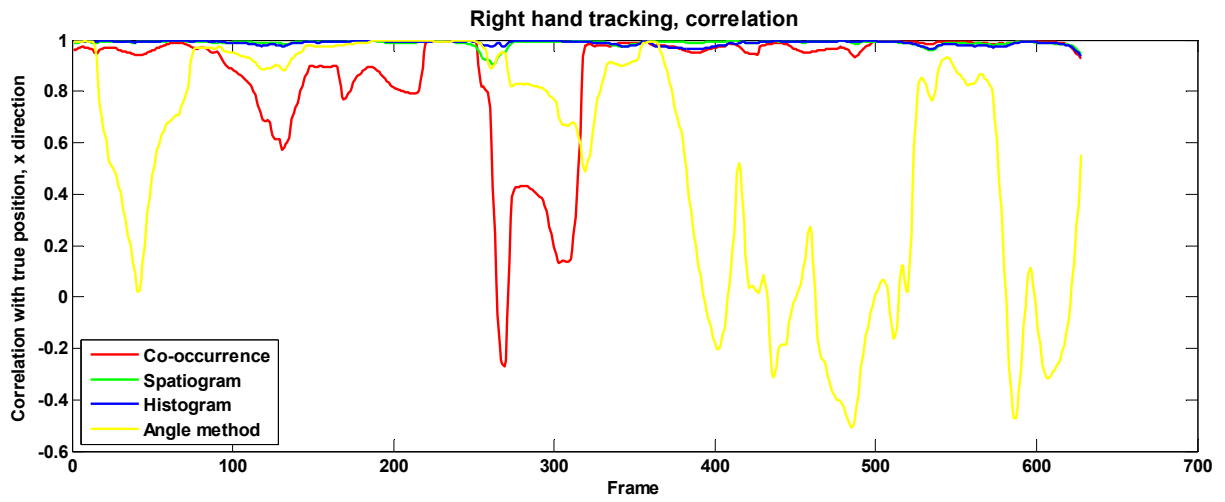
Venstre hånd, posisjonskorrigering:



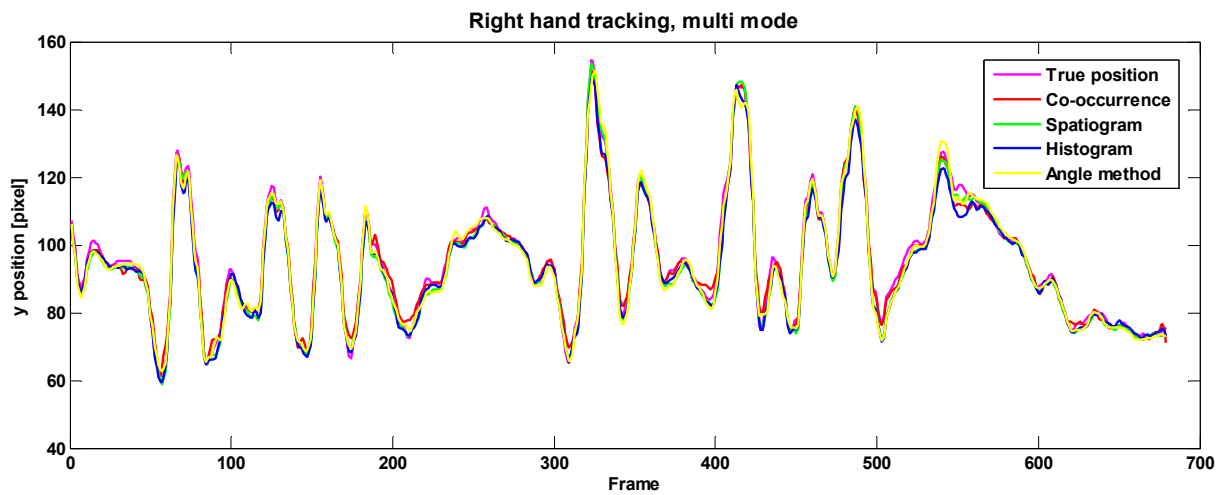
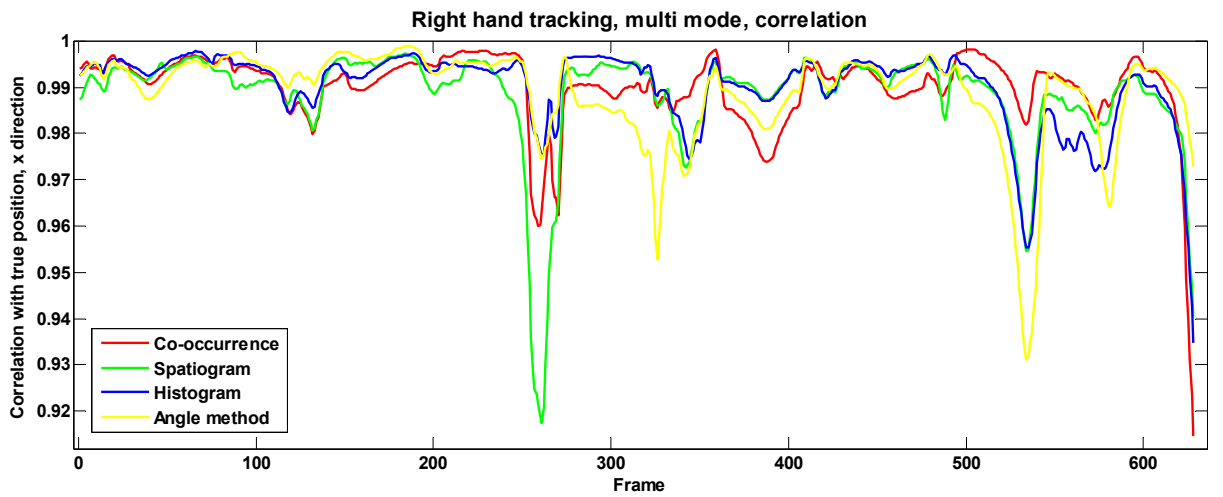
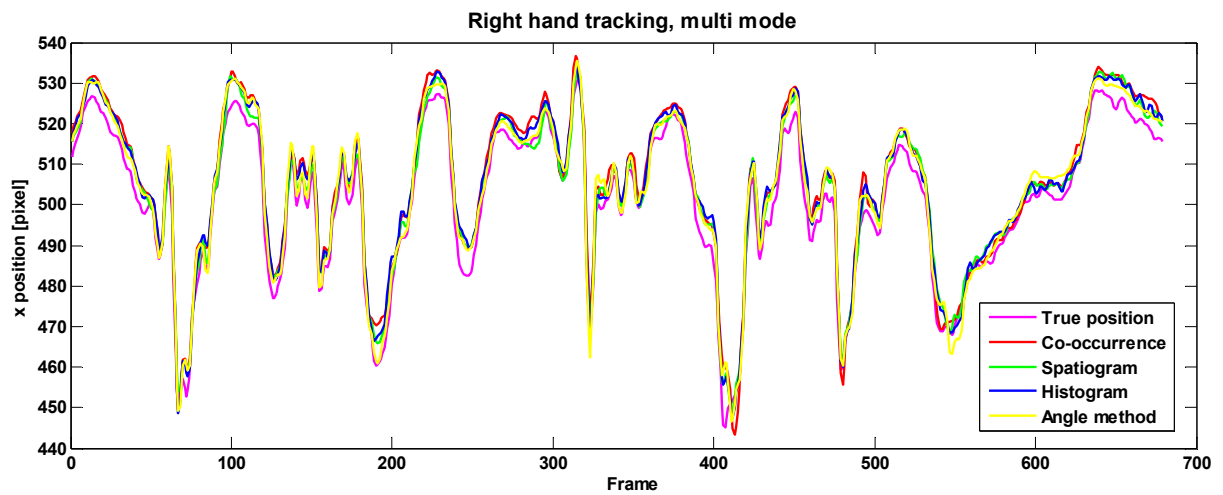


Høyre hånd, metodene hver for seg:

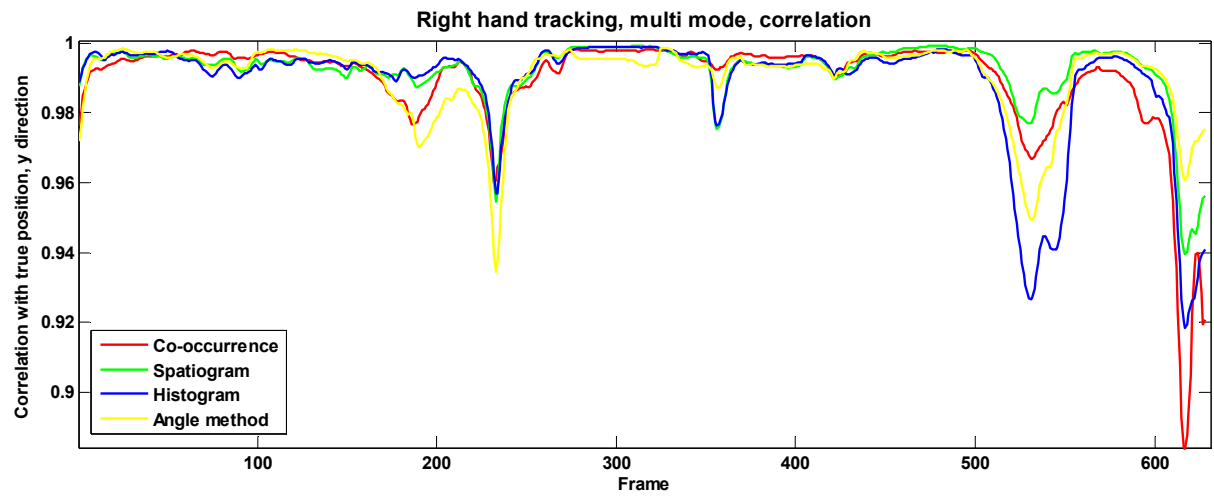




Høyre hånd, posisjonskorrigering:

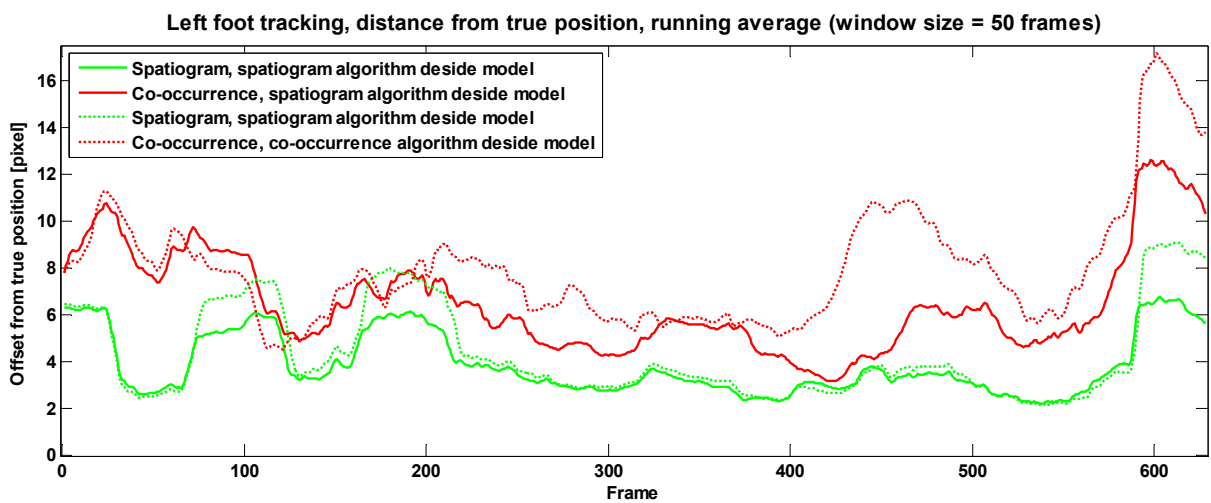
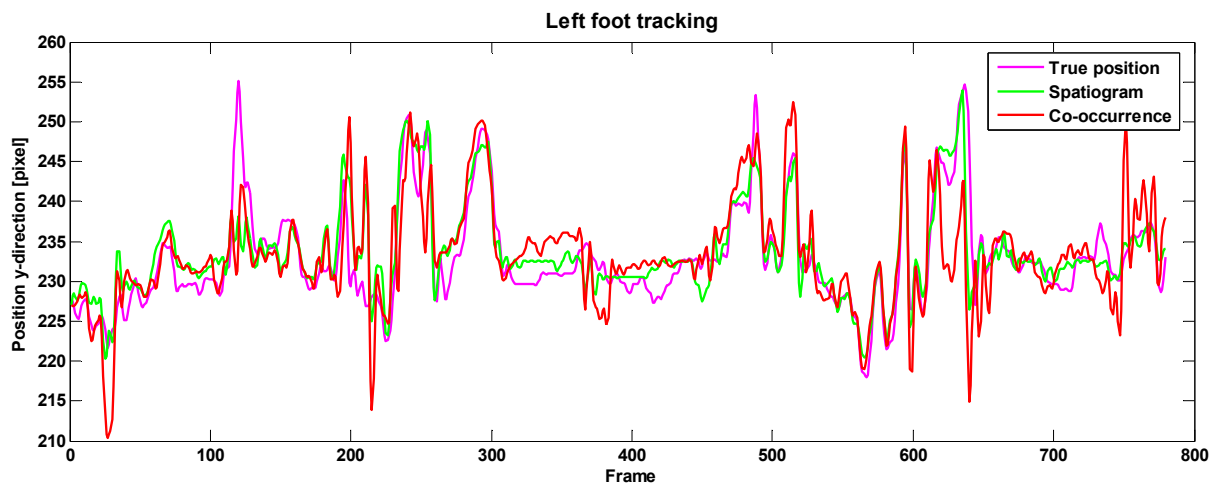
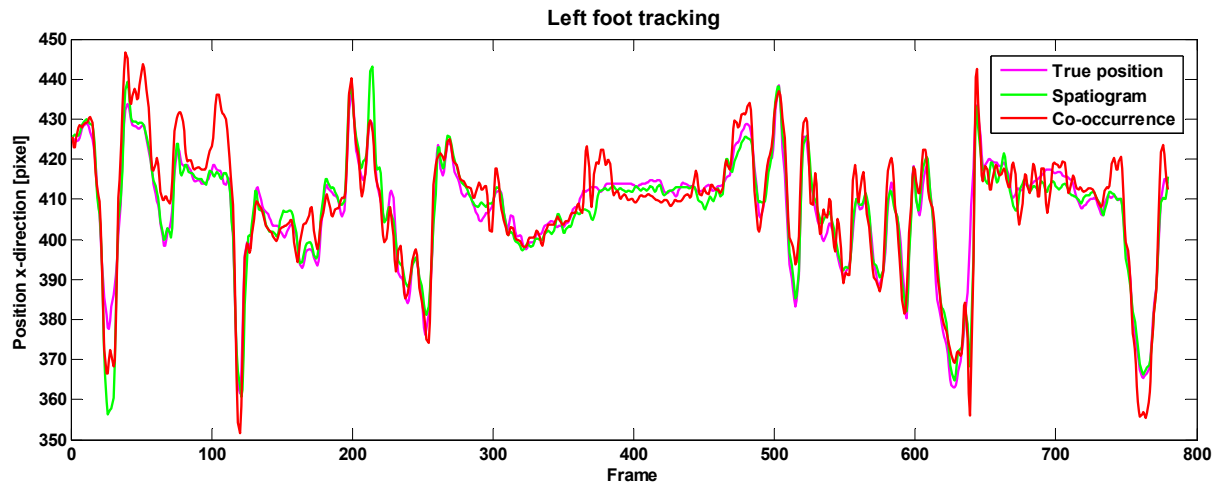




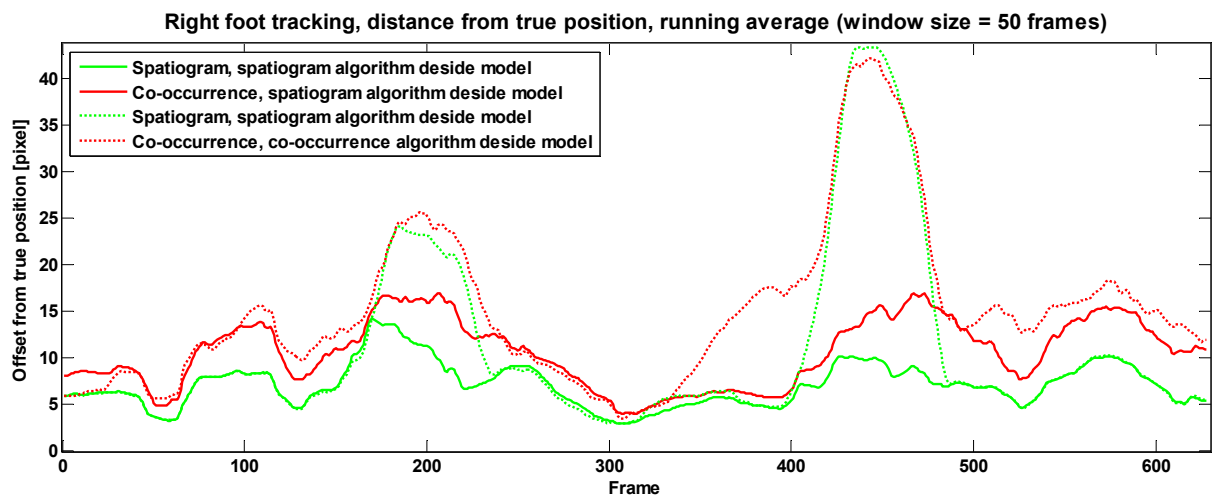
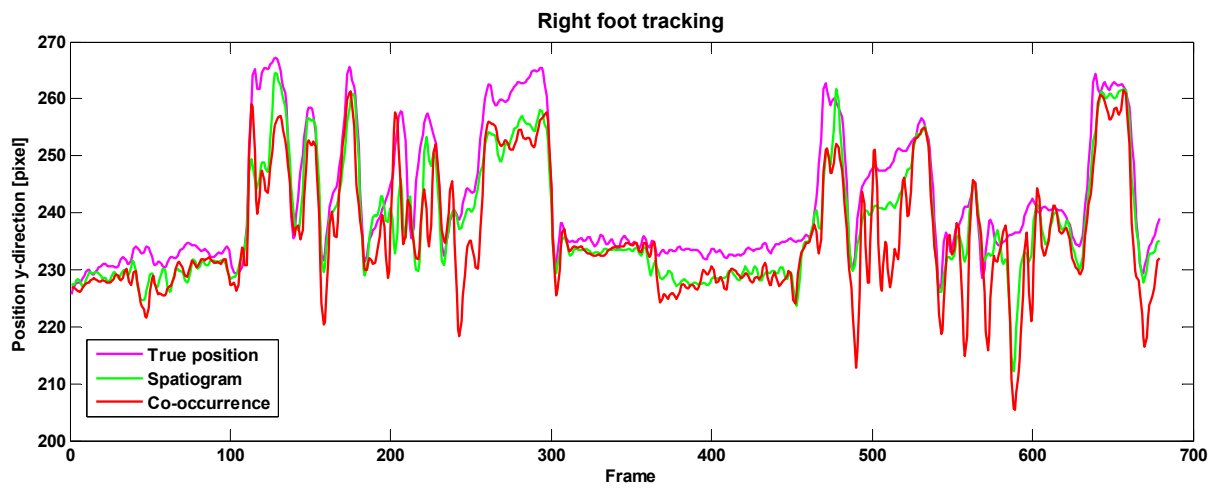
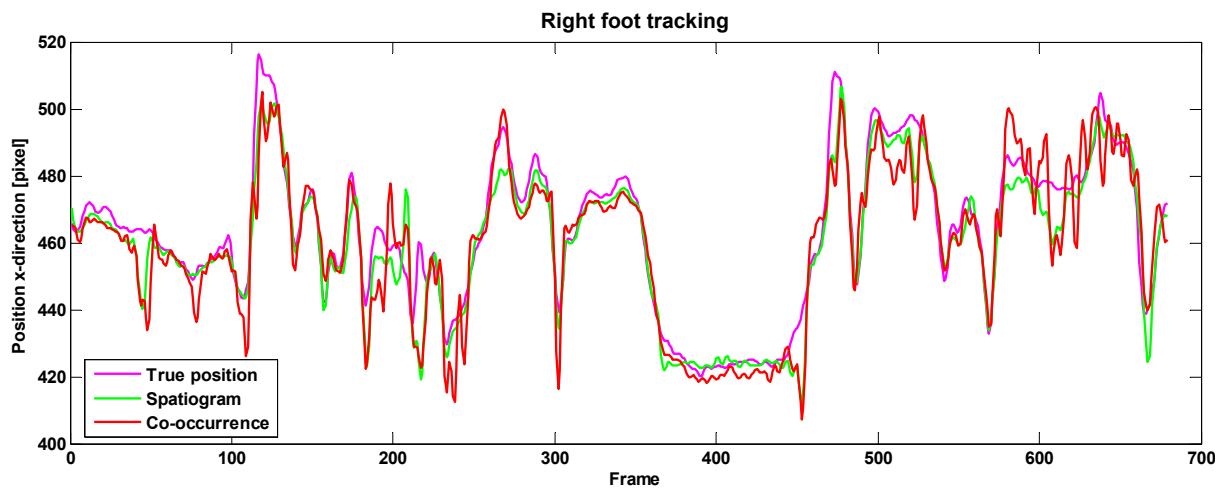


# Appendiks B Grafer fra følging av føtter

Vestre fot, felles modellvalg:

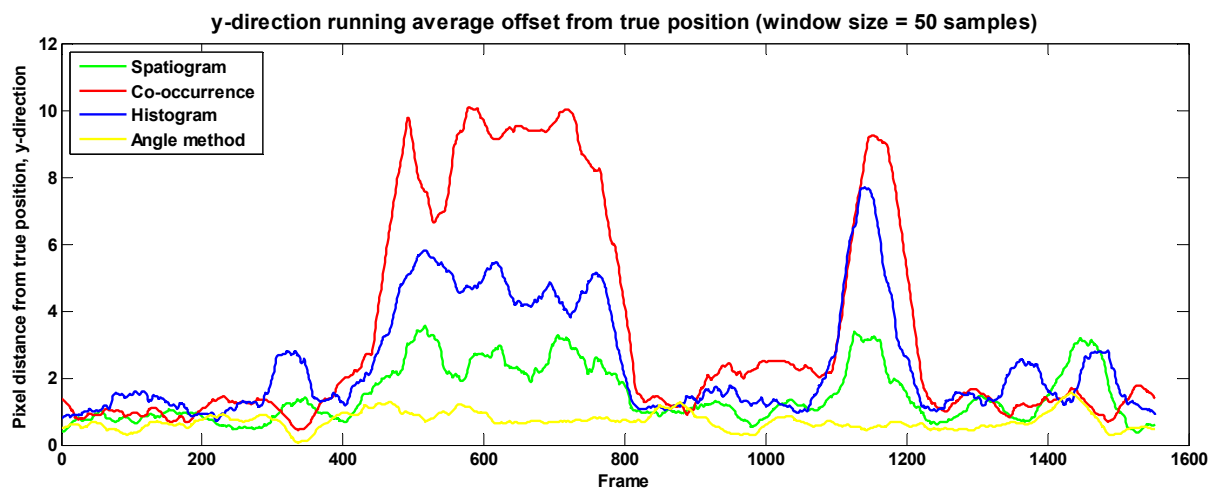
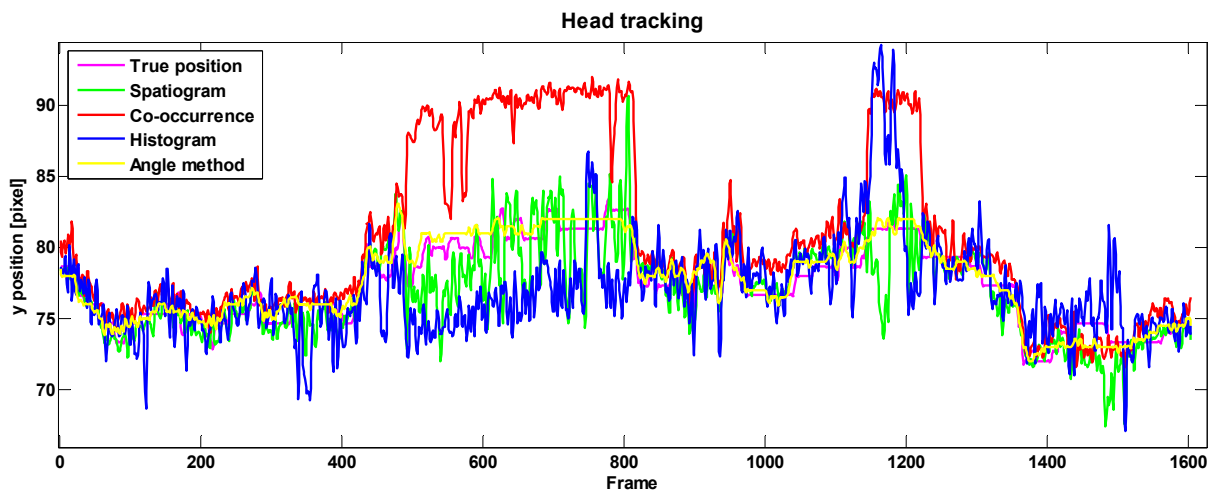
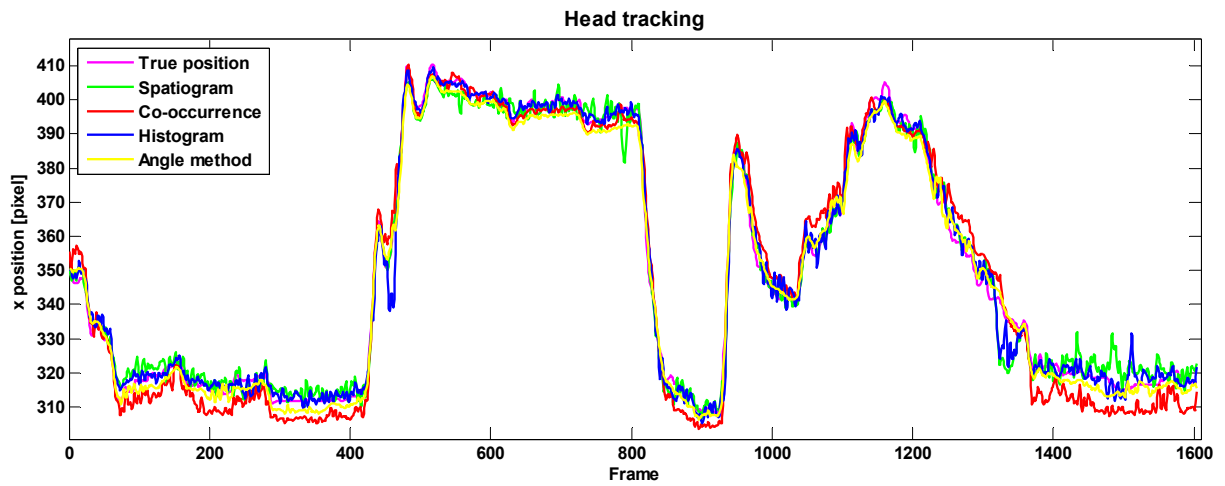


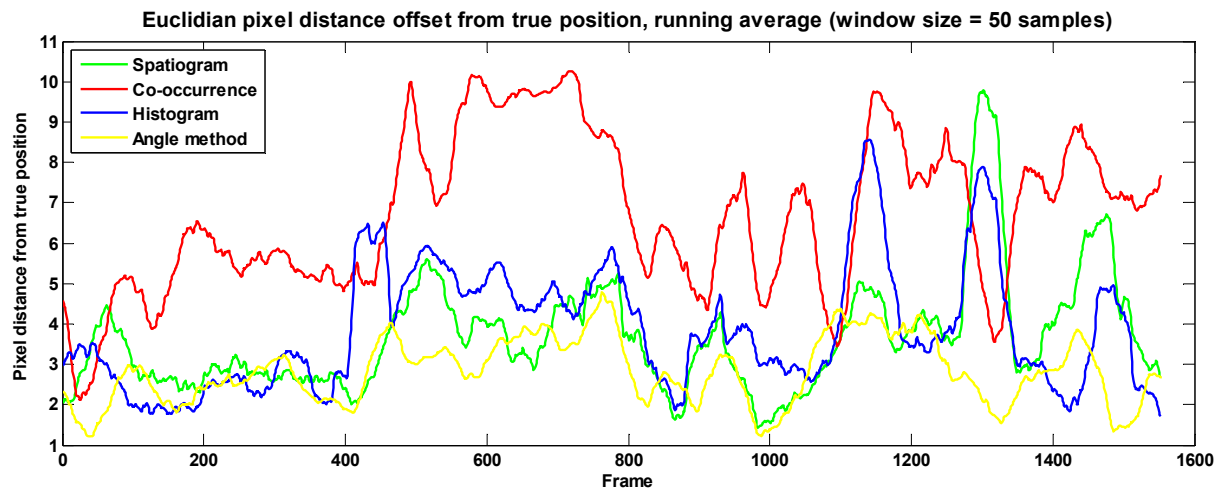
Høyre fot, felles modellvalg:



# Appendiks C Grafer fra følging av hode

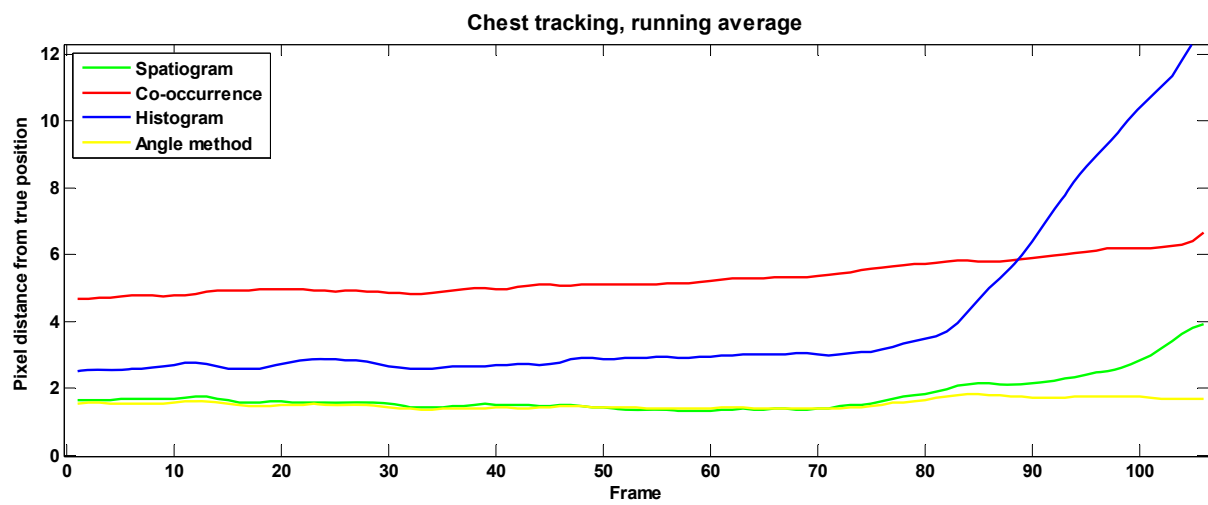
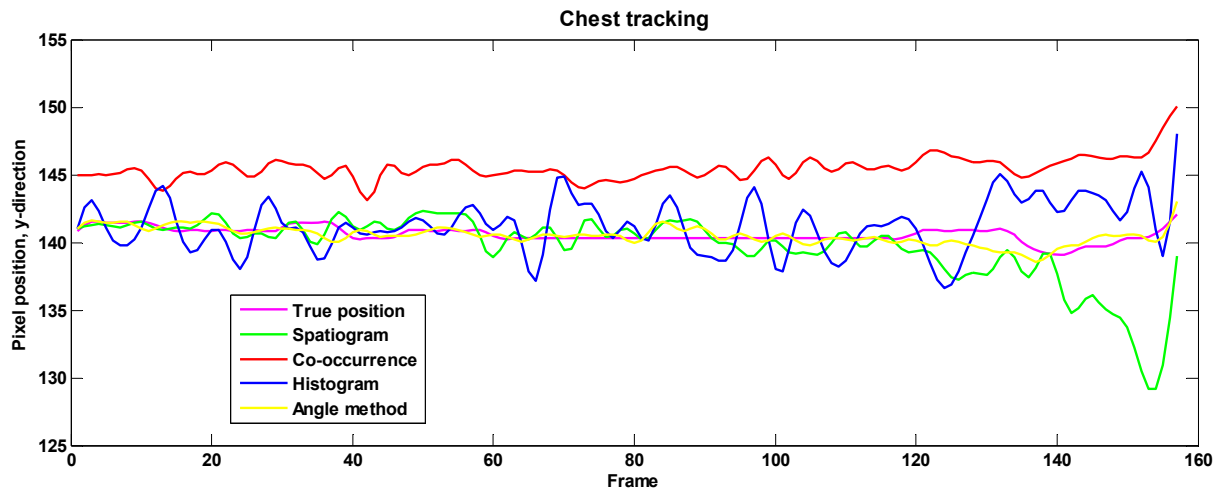
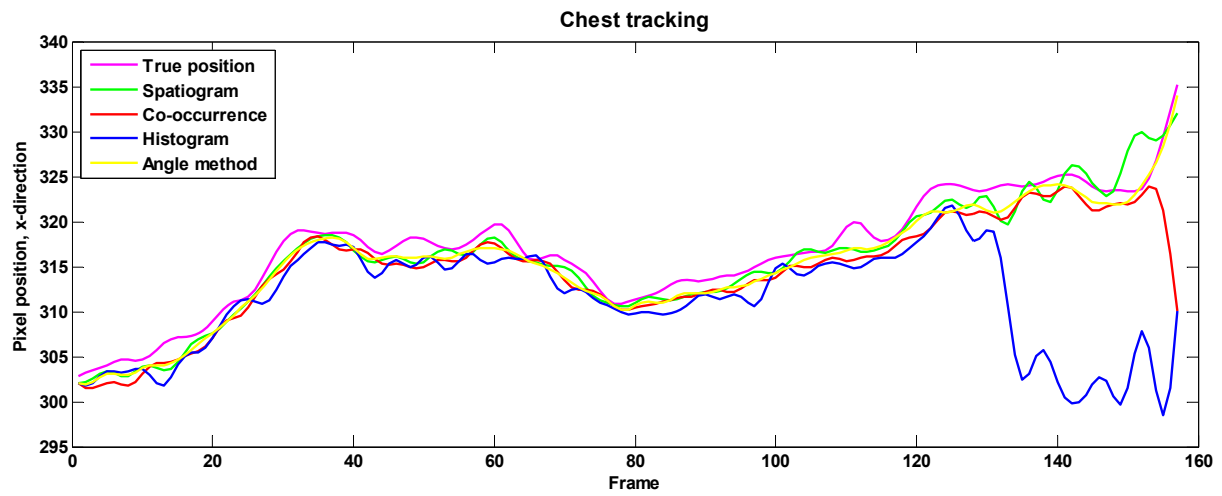
Posisjon for nese:



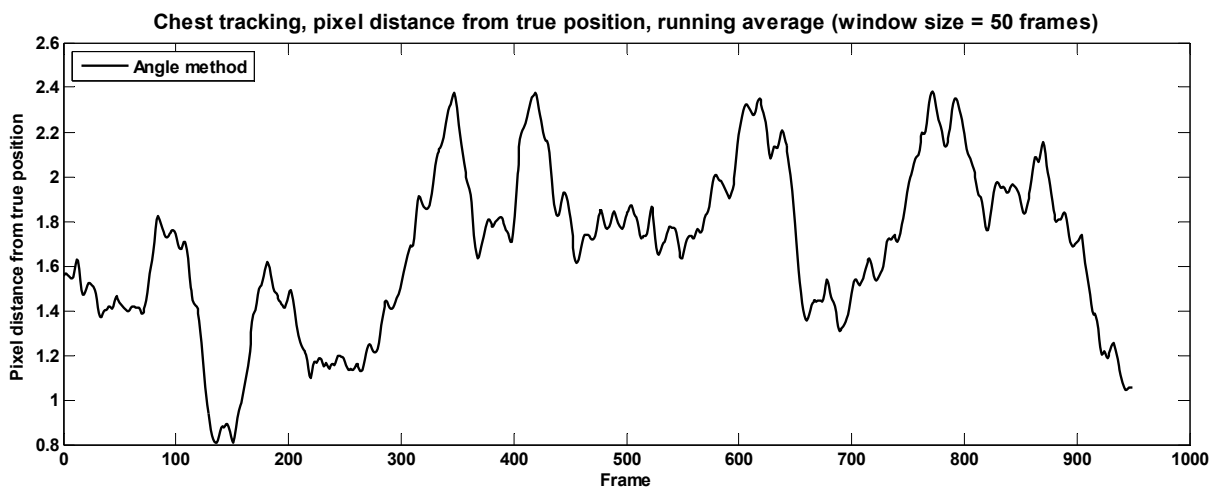
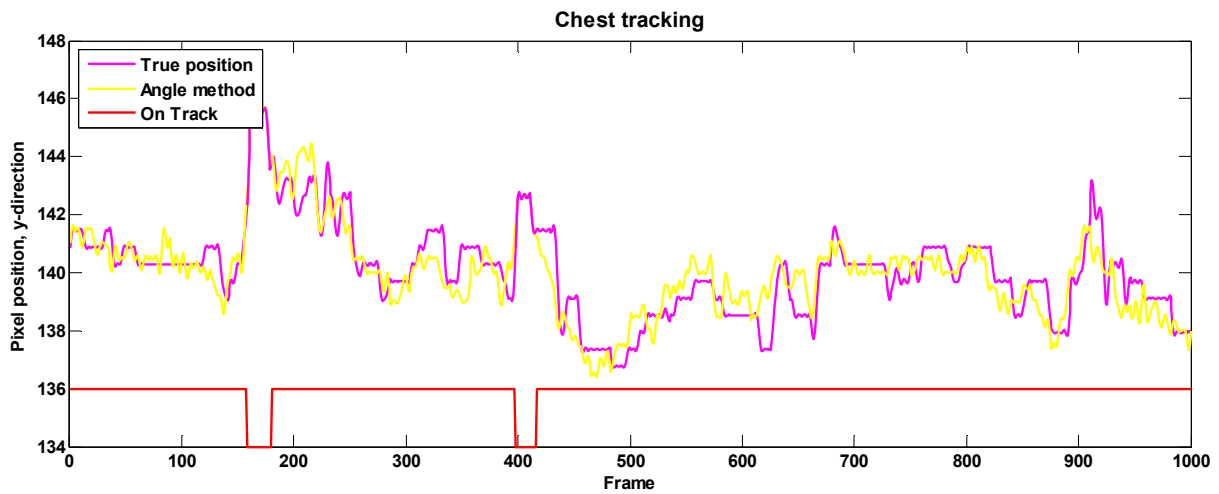
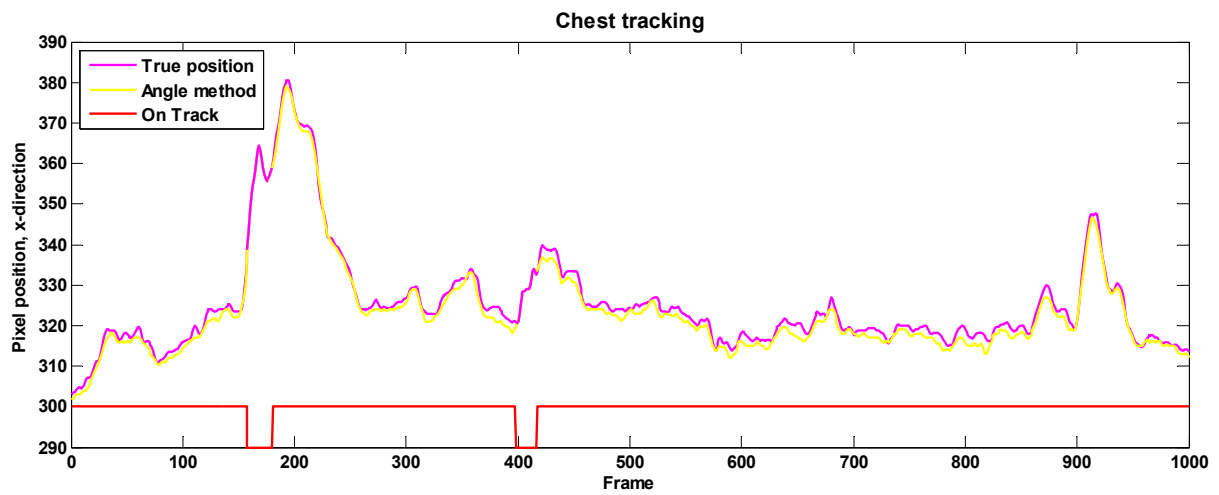


# Appendiks D Grafer fra følging av bryst

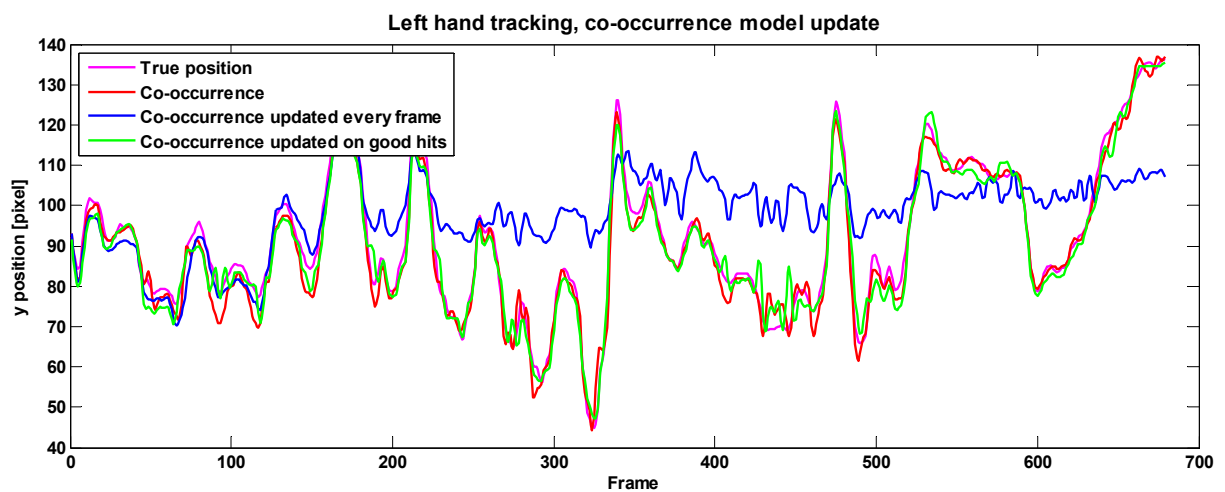
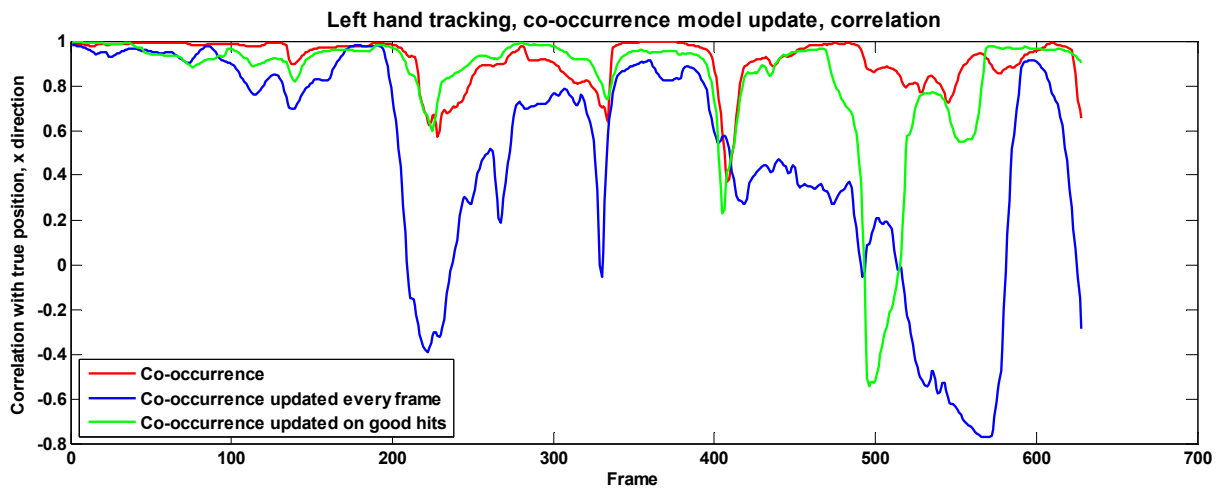
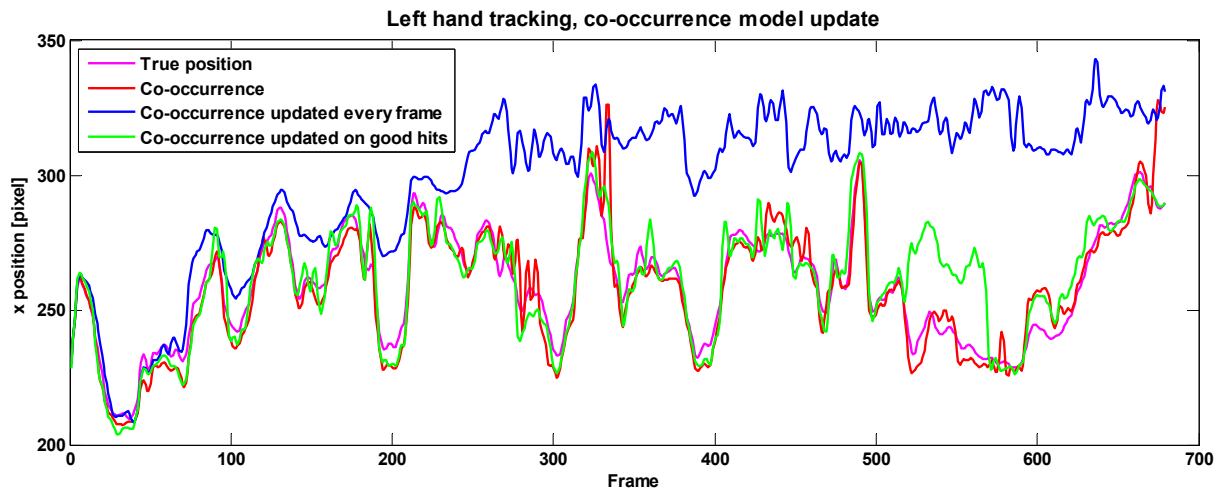
Fram til første okklusjon:



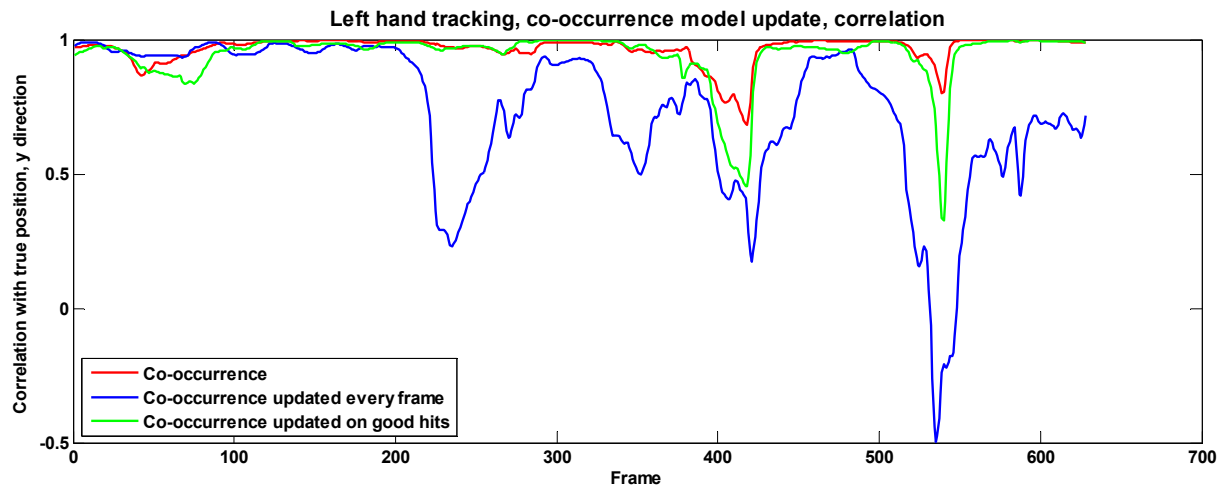
Hele serien:



# Appendiks E Plott av co-occurrence modeloppdatering







## **Appendiks F Programkode og videoer**

Vedlagt ligger en DVD med programkode, noen videofiler og denne rapporten i elektronisk form.