

VISUAALISEN ATTENTION ANALYSOINTI
PARIOHJELMOINNISSA ASL 504- JA TOBII 1750 –
SILMÄNLIIKEKAMEROILLA NAUHOITETUN
DATAN AVULLA

Timo Ikonen

8.7.2008

Joensuun yliopisto
Tietojenkäsittelytiede
Pro gradu -tutkielma

TIIVISTELMÄ

Silmänliiketutkimuksessa käytettävää katseenseurantaa voidaan soveltaa käytettävyytutkimuksessa tietotekniikka-alalla. Tällöin halutaan yleensä kerätä tietoa järjestelmän käyttäjän tarkkaavaisuuden suuntautumisesta ja kiinnittymisestä näytölle. Saatua dataa voidaan sen jälkeen käyttää esimerkiksi käyttöliittymän suunnittelun apuna. Tutkimukseen löytyy avuksi erilaisia silmänliikekameroita ja niiden mukana tulevia analysointi- ja nauhoitusohjelmistoja. Tässä tutkielmassa on pyritty analysoimaan kollaboratiivista silmänliikedataa ASL 504 ja Tobii 1750 silmänliikekameroiden välillä. Analysointi pohjautuu testitilanteessa nauhoitettuun raakadataan. Nauhoitukset sisältävät dataa, jota on nauhoitettu niin yksin- kuin pariohjelmointia demonstroimalla. Tavoitteena on ollut selvittää miten raakadatat eroavat toisistaan ja miten vertailukelpoisia ne ovat keskenään.

Tutkielma jakaantuu siten, että ensin perehdytään pariohjelmointiin eräänä työskentelytapana ohjelmistotuotannossa. Sen jälkeen tutustutaan visuaaliseen attention sisältäen tietoa ihmisen näköjärjestelmästä, näköaistista ja koko havaitsemisprosessista. Sen jälkeen mennään tarkemmin katseenseurantaan ja mittaamiseen. Lopussa käydään läpi nauhoituksissa saadun raakadatan analysointi.

Tutkielmassa käydään läpi ensin molemmista raakadatoista fiksaatioiden määrät, kestot, keskimääräiset kestot ja fiksaatioiden suhteellinen kesto nauhoitusajan kokonaisajasta. Sen jälkeen tutkitaan vielä päällekkäisiä attention kohteita sekä ASL:n, että Tobii raakadatoista. Tutkimus osoittaa, miten ASL 504:n raakadatasta fiksaatioita suodatettaessa suodatettujen fiksaatioiden keskimääräinen kesto on lyhyempi kuin Tobii 1750:stä saadussa raakadatassa.

ACM-luokat (ACM Computing Classification System, 1998 version): D.1.m, D.2.1

Avainsanat: Pariohjelmointi, Visuaalinen attention, silmänliiketutkimus, katseenseuranta

SISÄLLYSLUETTELO

1	JOHDANTO	1
2	KOLLABORATIIVINEN TYÖSKENTELY	2
2.1	Pariohjelmointi.....	2
2.1.1	Pariohjelmoinnin historia.....	2
2.1.2	Pariohjelmointi ja extreme programming	4
2.1.3	Roolit pariohjelmoinnissa	4
2.1.4	Työskentely pariohjelmointia hyväksikäyttäen	5
2.1.5	Ohjelmistoprosessiin liittyvät edut	7
2.1.6	Työskentelyyn liittyvät edut	7
2.2	PARIOHJELMOINNIN VAATIMAT RESURSSIT	9
2.2.1	Taloudelliset resurssit	10
2.2.2	Henkilöstöön liittyvät resurssit	10
2.2.3	Aikatauluun liittyvät resurssit	11
2.2.4	Käyttöönottoon liittyvät ongelmat	12
2.2.5	Työn tehokkuus.....	13
2.2.6	Työskentelytavat	14
2.2.7	Koulutukseen liittyvät yksityiskohdat.....	15
3	VISUAALINEN ATTENTIO.....	16
3.1	Taustaa	16
3.2	Näköaisti	17
3.3	Näköjärjestelmä	18
3.3.1	Silmän rakenne ja toiminta	18
3.3.2	Silmänliikkeet	20
3.3.3	Ennakoivat silmänliikkeet.....	22
3.4	Kognitio	24
3.5	Visuaalinen havaitseminen	24
3.5.1	Skeemat.....	25
3.5.2	Valikoiva katsominen	25
4	KATSEENSEURANTA	27
4.1	Silmänliiketutkimuksen historiaa.....	27
4.2	Katseenseurannan teoriaa.....	28
4.2.1	Katseenseuraaja.....	29
4.2.2	Katseenseurannalla saatavat hyödyt	30
4.2.3	Katseenseurannan haasteet.....	31
4.3	Videopohjainen tekniikka katseenseurannassa	32
4.4	Silmänliikkeet lukemisen aikana	32
4.5	Huomioitavia asioita katseenseurannassa	33
4.5.1	Katseenseurannassa käytettävät laitteet	35
4.5.2	Ohjelmistot.....	37
4.5.3	Tutkimusympäristö	39
4.6	Katseenseurannan vaiheet.....	39
4.6.1	Testihenkilön asettaminen paikoilleen.....	40
4.6.2	Katseenseurantaohjelman säätäminen	40
4.6.3	Kalibrointi	41
4.6.4	Monitorointi	42

4.7	Tutkimuksessa käytetyt silmänliikekamerat	42
4.7.1	ASL 504	43
4.7.2	Tobii 1750.....	44
5	KATSEENSEURANNAN MITTAAMINEN	46
5.1	Johdatus katseenseurannan mittaukseen	46
5.2	Mitä mitattiin	47
5.3	Miten mitattiin	48
5.4	Datan analysointi	52
5.4.1	Alkuasetukset yksinohjelmointia simuloitaessa	53
5.4.2	Alkuasetukset pariohjelmointia simuloitaessa.....	59
5.4.3	ClearView ja Eyenal - fiksaatioalgoritmit	60
5.5	Datan vertailu Tobii:n ja ASL:n välillä	63
5.5.1	Fiksaatioalgoritmi	65
5.6	Attention päällekkäisyys	67
5.7	Testituloksiin vaikuttavat tekijät.....	70
6	YHTEENVETO	73
	VIITELUETTELO.....	74

1 JOHDANTO

Silmänliiketutkimusta on käytetty hyväksi ohjelmistotuotannossa pitkälti käytettävyytutkimuksessa, jossa pääpaino on ollut katseenseurannassa. Katseenseurannan avulla on saatu tietoa siitä, miten käyttäjä reagoi katseellaan tiettyyn testattavaan käyttöliittymään. Samalla on saatu tärkeää tietoa siitä, miten nopeasti käyttäjä pystyy omaksumaan käyttöliittymän sisällön, muistamaan tärkeimmät toiminnot ja yleensä käyttämään käyttöliittymää. Näiden tietojen perusteella käyttöliittymiä on voitu muokata yhä enemmän käyttäjäystävällisempään suuntaan.

Nykyään silmänliiketutkimuksessa käytetään hyväksi saatavilla olevia silmänliikekameroita ja niiden mukana tulevia ohjelmistoja. Laitteita löytyy eri valmistajien ja eri mallien mukaisesti useita erilaisiin käyttötarkoituksiin. Perusajatus laitteilla on kuitenkin aina sama. Niiden avulla pyritään keräämään tietoa siitä, miten ihmisen silmät ja katse toimivat kun henkilö katsoo johonkin tai tekee jotain tehtävää, jossa joutuu kiinnittämään katseensa tiettyihin kohtiin näytöllä.

Tässä tutkielmassa suoritetaan vertailu kahden silmänliikekameratyyppin välillä, perustuen niiden antamaan dataan tietyssä testitapauksessa. Vertailu suoritetaan analysoimalla laitteista saatua raakadataa, joka on tallennettu tietyssä koetilanteessa nauhoittamalla testihenkilön silmänliikkeitä. Samalla tutkitaan laitteistoista saadun raakadatan eroja niin yksinohjelmoinnissa, kuin myös pariohjelmoinnissa. Tutkielmassa käytetyt silmänliikekamerat ovat ASL 504 ja Tobii 1750.

Ensin perehdytään kollaboratiiviseen työskentelyyn eräänä työskentelytapana ohjelmistotuotannossa, koska myöhemmin tehdään mittauksia pariohjelmointiasetuksilla. Sen jälkeen käydään läpi mitä visuaalinen attentio tarkoittaa ja samalla tutustutaan ihmisen näköjärjestelmään, silmän toimintaan ja havaitsemiseen. Sen jälkeen käsitellään katseenseurannan teoriaa jonka jälkeen esitellään suoritettujen mittaustulokset. Lopuksi tehdään vielä yhteenveto saaduista tuloksista.

2 KOLLABORATIIVINEN TYÖSKENTELY

Tässä tutkielmassa tarkastellaan eräänä työtapana kollaboratiivista työskentelyä. Kollaboratiivinen työskentely on tapa, jossa työtä tehdään läheisessä yhteistyössä. Esimerkkinä kollaboratiivisesta työskentelystä käytetään pariohjelmointia.

Tutustutaan lähemmin pariohjelmointiin eräänä työskentelytapana ohjelmistotuotannossa. Pariohjelmointia käytetään myöhemmin katseenseurannan yhteydessä, kun tehdään nauhoituksia yksin- ja pariohjelmointiasetuksilla. Tätä varten on syytä ensin perehtyä tarkemmin pariohjelmoinnin yksityiskohtiin.

2.1 Pariohjelmointi

Aluksi tutustutaan pariohjelmoinnin historiaan, taustoihin ja siihen miten pariohjelmointi on kytköksissä extreme programming - menetelmään. Luvussa esitellään myös pariohjelmoinnin sisältämät roolit, sekä käydään läpi miten pariohjelmoinnissa työskennellään.

2.1.1 Pariohjelmoinnin historia

Pariohjelmoinnin historia alkaa vuodesta 1995, jolloin Larry Constantine havaitsi ohjelmoijaparien tuottavan koodia nopeammin ja virheettömämmin kuin koskaan aikaisemmin. Samana vuonna Jim Coplien julkaisi organisaatiomallin jota hän kutsui kehittämisen-pareittain (*developing-in-pairs*) – malliksi. Nämä edellä mainitut työskentelytavat nojasivat parityöskentelyyn ja niiden perusteella pareittain tehtyä työtä

voitaisiin hyvin kokeilla myös laajemmassa määrin yksinohjelmointia suosivalla ohjelmistoalalla. (Kessler & Cunningham, 2000)

Vuonna 1996 alkoi Smalltalk ohjelmoija ja konsultti Kent Beck, yhdessä Ward Cunninghamin ja Ron Jeffriesin kanssa kehittämään extreme programming eli XP – ohjelmistonkehitystapaa. Se perustui ketteriin menetelmiin, joissa ohjelmistoa kehitetään niin että toimiva ohjelmisto on etusijalla ja jossa ohjelmistoprosessissa käytetään hyväksi suoraa viestintää ja muutoksiin osataan reagoida nopeasti. Kyseinen tapa pyrkii keskittymään pääasiassa tuotettavaan tuotteeseen, eikä niinkään ylenmääräiseen dokumentointiin ja ohjelmistotuotannon muuhun sivutuotantoon.

Ketterien menetelmien tavoitteena on jakaa ohjelmistokehitys lyhyisiin iteraatioihin eli jaksoihin, joihin jokaiseen kuuluu ohjelmistoprojektin vaiheet suunnittelusta dokumentointiin. Raskaan dokumentoinnin sijaan ketterät menetelmät painottavat tiedon välityksessä henkilöiden keskinäistä keskustelua. Ketterissä menetelmissä dokumentteja ja niiden määrää ei käytetä tuotteen mittaamiseen samassa määrin kuin yleensä muissa menetelmissä. Suunnittelu on kuitenkin tärkeää, mutta nyt suunnitelmat ovat joustavia muutoksille.

Extreme programming toi mukanaan joukon uusia tapoja kehittää ohjelmistoja. Ohjelmistotuotannon tärkeimmät vaiheet eli määrittely, suunnittelu, toteutus, testaus ja käyttöönotto otettiin kaikki omalla tavalla huomioon XP:ssä. Pääpaino oli prosessin joustavuudella ja toisaalta ennen kaikkea tavalla kommunikoida asiakkaan kanssa paljon entistä tehokkaammin. XP:n tarkoituksena on saavuttaa laadukkaampaa tulosta sovitussa aikataulussa. Vieläpä niin, että asiakas olisi koko ajan tietoinen projektin etenemisestä ja koko ajan informoituna valmistettavan tuotteen sisällöstä. (Kessler & Cunningham, 2000)

2.1.2 Pariohjelmointi ja extreme programming

Kuten edellä jo mainittiin, pariohjelmointi on eräs XP:n työskentelytapa, toisin sanoen käytäntö. Extreme programming nojaa pariohjelmointiin niin vankasti, että se vaatii kaiken tuotantoon liittyvän koodinkirjoituksen tapahtuvan pareittain. Tämä ei kuitenkaan aina käytännössä toteudu täydellisesti, mistä myöhemmin lisää. Toinen XP:n tavoite on, että ohjelmakoodi on tuotannosta vastaavien henkilöiden osalta kollektiivisessa omistuksessa, eli kaikilla tekijöillä on mahdollisuus päästä käsiksi ohjelmakoodiin ja mahdollisuus pystyä tulkitsemaan ja lukemaan sitä helposti. Tuotetun ohjelmakoodin on tätä varten oltava sijoitettuna fyysisesti siten, että siihen päästään helposti käsiksi (esimerkiksi oltava sijoitettuna hakemistoon, jonne kaikilla kyseisen sovelluksen tuottajilla on luku- ja kirjoitusoikeudet).

Koodin uudelleenkäytön on oltava mahdollisimman vaivatonta ja yrityksen on toimittava tarkasti yhteisten, ennalta sovittujen työskentelytapojen mukaan. Muuten XP:ssä määritellyt tavoitteet eivät täyty. XP antaa tiettyjä suosituksia myös itse työn tekemiselle. Eräs niistä on kriittinen suhtautuminen ylitöiden tekemisen suhteen ja sen sijaan aikataulua pyritään jopa muuttamaan siihen suuntaan että ylitöitä ei tarvitsisi tehdä. (Succi & Marchese, 2001)

2.1.3 Roolit pariohjelmoinnissa

Työskentely pariohjelmointia hyväksikäyttäen asettaa muutamia ennalta määriteltyjä rajoitteita. Yksi tärkein niistä on parien roolit. Pariohjelmoinnissa on yksinkertaisesti ilmaistuna kaksi henkilöä yhden koneen äärellä. Kummallakin parin jäsenellä on oma roolinsa. Rooleja ovat toteuttaja (*driver*) ja varmentaja (*navigator*). Molemmat henkilöt, sekä toteuttaja että varmentaja, istuvat yhden koneen äärellä ja työskentelevät saman ohjelman parissa. Toteuttaja käyttää näppäimistöä ja hiirtä ohjelmoidessaan ja

varmentaja seuraa vieressä ja kommentoi ohjelmakoodia, mikäli näkee siinä jotain huomauttamisen aihetta. Pari keskustelee muutenkin keskenään ja varmentaja huomauttaa aina, jos hänen mielestään jokin asia on virheellisesti tai epävalidisti ilmaistuna ohjelmakoodissa. Toteuttajan tulee pyrkiä toteuttamaan ohjelmoimansa tuote parhaalla mahdollisella tavalla. Tässä häntä auttaa varmentaja. Toteuttajan on myös aina pystyttävä huolella perustelevaan ratkaisunsa liittyen ohjelmakoodiin kirjoitukseen varmentajalle. Varmentaja keskittyy työssään pohtimaan sopiiko toteuttajan lähestymistapa ongelman ratkaisuun, onko algoritmi hänen mielestään mahdollisimman hyvä, ja miettimään onko testejä jotka eivät menisi läpi. Varmentaja pohtii myös onko olemassa parempia (yksinkertaisempia tai tehokkaampia) lähestymistapoja kyseisen ongelman ratkaisemiseen. Lisäksi varmentaja tarkastaa onko koodissa virheitä joko syntaksissa tai ohjelman logiikassa. Tällainen työnjako tarjoaa uuden lähestymistavan ohjelmistotuotantoon verrattuna yksinohjelmointiin. Varmentaja joutuu katsomaan tuotetta siitä näkökulmasta, että tuote olisi mahdollisimman tarkoin vaatimusten ja määritysten mukainen. Toteuttaja pyrkii osaltaan työssä sellaiseen tulokseen, että se tyydyttäisi varmentajan ja tietysti ohjelmistolle asetetut vaatimukset. (Succi & Marchese, 2001)

2.1.4 Työskentely pariohjelmointia hyväksikäyttäen

Pari työskentelee edellä esitettyjen roolien mukaisesti. Rooleilla on jo olemassa valmiina tietyt vastualueet, joista kunkin parin jäsenen tulee pitää kiinni. Pariohjelmoinnissa parin rooleja voidaan vaihtaa useasti päivässä, joten kiertoa työtehtävissä saadaan näin ollen helposti aikaan. Tyylinä pariohjelmointi on varsin dynaaminen ja se perustuu henkilöiden vuorovaikutukseen. Kommunikointi on kaikin tavoin välitöntä. Yhteistyön ansiosta ideaaliympäristössä ohjelmointipari työskentelisi koko ajan yhdessä, mutta tosiasiaassa parin työskentelyyn vaikuttavat esimerkiksi sairastumiset ja aikatauluista johtuvat poikkeamat. Tämä johtaa lopulta siihen, että pari ei voi työskennellä koko aikaa täysin yhdessä. Sen takia parin on yleensä priorisoitava ne asiat ohjelmiston kehityskaaressa, jotka ovat erityisen tärkeitä tehdä yhdessä.

Samalla on myös mietittävä mitkä asiat pari voi tehdä erikseen. Erikseen tehtävien töiden on oltava helposti kokoon kasattavissa, kun pari palaa jälleen yhteen. Kun suurikokoinen, ohjelmistoihin keskittynyt työskentelyryhmä omaksuu tietyllä ajanjaksolla pariohjelmoinnin työskentelytavakseen, käy tietyn parin pitkäaikainen yhteistyö vähitellen tarpeettomaksi, koska työskentelytapa on tässä vaiheessa muodostunut henkilöille niin tutuksi ja pariin koostumusta voidaan vaihtaa ilman ongelmia. (Kessler & Cunningham, 2000)

Pariohjelmoinnissa parityötä voidaan käyttää hyödyksi myös ohjelmiston määrittely- ja suunnitteluvaiheissa. Tämä onkin varsin loogista ja järkevää, sillä tällöin saadaan ohjelmistonkehityksessä parityölle vankka perusta jo heti alusta alkaen. Määrittelyssä ja suunnittelussa pätevät samat hyvät puolet kuin varsinaisessa ohjelmointivaiheessakin. Pari ajattelee yhdessä ongelmia yksittäistä henkilöä syvällisemmin ja kyseenalaistaa paremmin erilaisten ongelmien erilaiset ratkaisuvaihtoehdot. Edellä mainitut vaiheet ovatkin tärkeitä pariohjelmoinnissa. Näin ohjelmisto on paremmin suunniteltu ja se on helpompi toteuttaa. Määrittelyä ja suunnittelua seuraa toteutus eli varsinainen pariohjelmointi sanan varsinaisessa merkityksessä. Viimeinen vaihe parityöskentelyssä on testaus, jossa parityötä ei ole asetettu ehdottomaksi työskentelytavaksi. Pääasia on, että pari tekee ainakin testitapaukset yhdessä, mutta varsinainen testaus voidaan suorittaa yksin eri koneilla. Jos ja kun virheitä ilmenee, palaa pari takaisin yhteen ja korjaa havaitut virheet yhdessä. (Kessler & Cunningham, 2000)

Pareja muodostettaessa voidaan tietysti toimia varsin vapaasti, mutta eräiden tutkimusten mukaan voidaan myös käyttää tiettyjä ennalta määrättyjä asetelmia joiden mukaan voidaan saada haluttuja tuloksia aikaan. Esimerkiksi jos aloittelijan halutaan oppivan uusia asioita, kuten yrityksen tai yhteisön toimintatapoihin liittyviä käytäntöjä, muodostetaan pari siten, että siinä on toisena osapuolena ammattilainen. Ammattilainen Näin kokeneempi parin jäsen opastaa työssään aloittelijaa, joka taas oppii yhteistyön seurauksena helposti miten tietyt käytännöt toimivat yrityksen sisällä. (Lui & Chan, 2006)

2.1.5 Ohjelmistoprosessiin liittyvät edut

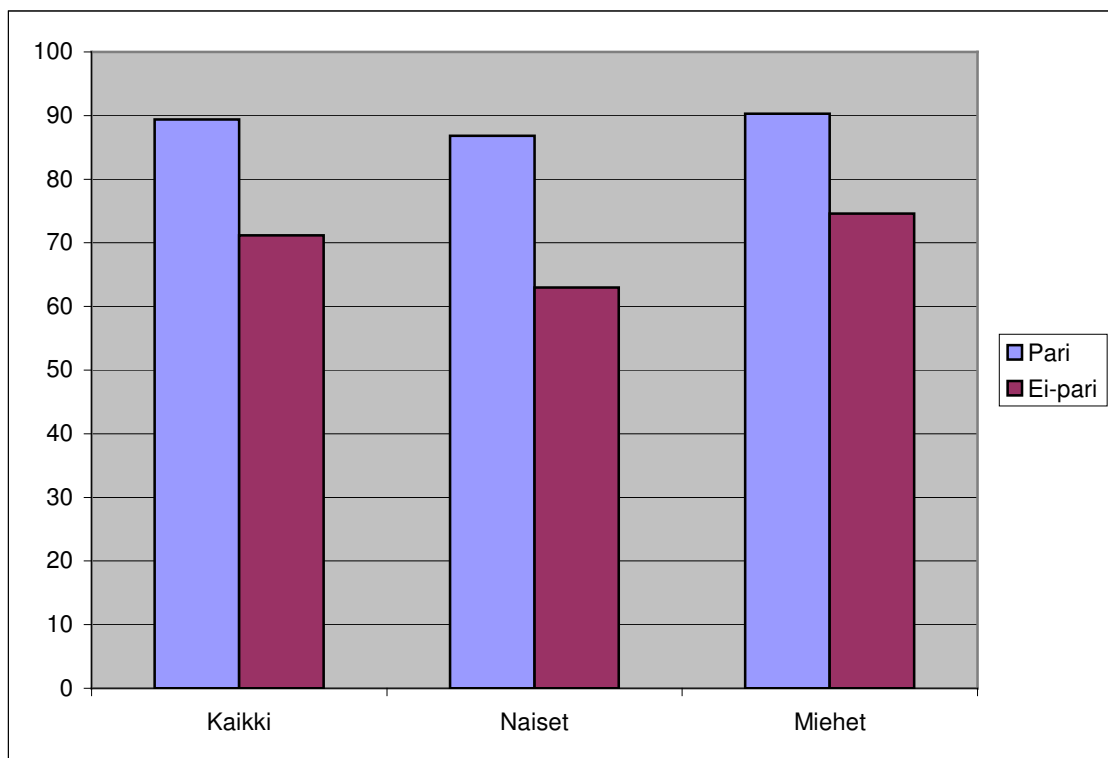
Ohjelmistoprosessin kannalta pariohjelmoinnilla saavutetaan todistetusti korkeampi tuottavuus laadullisesti kuin mitä yksinohjelmoinnilla. Silti ohjelmistonkehitysaikaa ei pariohjelmoinnilla voida lyhentää, vaan päinvastoin, kehitysaika jopa hieman pitenee. Pariohjelmointi tuottaa paremman tuloksen silloin kun ratkaistaan monimutkaisia ja laajoja ongelmia. Prosessin kannalta olisikin ideaali tilanne, jos kehitettäisi laajaa sovellusta yrityksessä jossa olisi jo käytössä ketteriin menetelmiin pohjautuva työskentelytapa. Parhaassa tapauksessa tietysti XP, joka jo sinänsä velvoittaa käyttämään pariohjelmointia. Pariohjelmointia ei pidä kuitenkaan käyttää kaikissa tapauksissa, etenkin triviaalisissa ja nopeasti työstettävissä ratkaisuisissa, joissa pärjätään hyvin yksinohjelmoinnilla. (Hulkko & Abrahamsson, 2005)

2.1.6 Työskentelyyn liittyvät edut

Pariohjelmointiin siirtymiseen saattaa liittyä muutosvastarintaa, kun se otetaan käyttöön sellaisessa yrityksessä tai yhteisössä jossa sitä ei ole aiemmin käytetty. Tästä huolimatta pariohjelmointi tarjoaa myös etuja yrityksen sisäiseen työskentelyyn. Pariohjelmoinnissa parit ja kokonaiset tiimit oppivat kommunikoimaan toistensa kanssa paremmin ja suoremmin kuin mitä yksin ohjelmoimissa. Taas täyttyy eräs XP:n tavoitteista (kts. kohta 2.1.2 Pariohjelmointi ja Extreme Programming) eli välitön kommunikointi. Pareittain työskentely voi mahdollisesti myös parantaa työmoraalia, etenkin jos henkilöt pitävät työskentelystä jossa toimitaan yhdessä toisen henkilön kanssa. Samoin työntekijät eivät varmastikaan tuhlaa niin paljon aikaansa esimerkiksi henkilökohtaisen sähköpostin lukemiseen tai muuhun työhön liittymättömään toimintaan, koska joutuvat nyt keskittymään työhönsä jo pelkästään parinsa takia.

Pariohjelmoinnin avulla voidaan pareille antaa vaativampia tehtäviä, kuin mitä henkilölle joka ohjelmoi yksin. Parityöskentely helpottaa vaikeampien ja laajojen ongelmien ratkaisemista. (Succi & Marchese, 2001)

California-Santa Cruzin yliopistossa tehdyssä tutkimuksessa, (McDowell, C. Werner, L. Bullock, H.E. Fernald, J.) joka tehtiin vuoden aikana tietojenkäsittelytieteen yliopisto-opiskelijoille, saatiin selvää näyttöä siitä että luottamus omaan työhön kasvaa käytettäessä pariohjelmointia. Kuvassa 1 nähdään miten luottamus omaan tuotettuun työhön lisääntyy, kun vertaillaan pariohjelmointia yksinohjelmointiin. Tutkimuksessa haastateltiin opiskelijoita jotka käyttivät työssään molempia työskentelytapoja. Tummat pylväät edustavat yksinohjelmointia ja vaaleat edustavat pariohjelmointia. Ensimmäinen pylväspari osoittaa kaikkien tutkittujen henkilöiden luottamuksen omaan työhönsä nousseen keskimäärin 18,2 % pariohjelmoinnin avulla. Naisten kohdalla luottamus omaan työhönsä on noussut peräti 23,8 % ja miesten 15,7 %. (McDowell & al., 2006)



Kuva 1. Luottamus omaan työhön pari- ja yksinohjelmoinnissa.

(McDowell & al., 2006)

Pariohjelmointi tuo hyötyä myös uusien asioiden oppimiseen ja käytännön kokemuksen lisäämiseen. Tämä on todettu myös yrityselämässä, jossa ihmiset ovat antaneet kiitosta pariohjelmoinnille juuri eräänä tapana opettaa ja oppia uusia asioita. (McDowell & al., 2006)

Tutkimusten mukaan työntekijän suunnittelutaidot paranevat pariohjelmointia hyväksikäytettäessä. Samoin koodinkirjoitustaito paranee ja syntaksiin liittyvät virheet vähenevät. Ohjelmoija tekee koodistaan luettavampaa pariohjelmoinnin ansiosta. (Succi & Marchese, 2001)

Utahin yliopistossa haastateltujen tietojenkäsittelytieteen opiskelijoiden mielestä (Succi, G. & Marchese, M) pariohjelmointi on mielekkäämpi työskentelytapa verrattuna yksinohjelmointiin. Alussa tosin pariin mielestä työmäärä epärealistisesti kaksinkertaistui (tutkimuksessa annettiin pareille kaksinkertainen työmäärä verrattuna yksinohjelmoiviin), mutta lopulta pareissa työtään tehneet olivat tyytyväisempiä työtapaansa. (Succi & Marchese, 2001)

Yrityksen johdon näkökulmasta pariohjelmointi helpottaa henkilöstöpolitiikkaa siten, että jos yksi henkilö joutuu syystä tai toisesta lähtemään ohjelmointitiimistä, ei hänen poislähtönsä vaikeuta jäljelle jäävien osuutta, sillä jokainen on tietoinen ohjelmitavasta tuotteesta riittävän hyvin. (Succi & Marchese, 2001)

2.2 PARIOHJELMOINNIN VAATIMAT RESURSSIT

Tässä luvussa on tarkoituksena käsitellä pariohjelmoinnin vaatimia resursseja. Ensin kiinnitetään huomiota taloudellisiin resursseihin, joiden jälkeen katsotaan mitä

vaatimuksia pariohjelmointi asettaa henkilöstölle ohjelmistoalalla toimivassa yrityksessä tai yhteisössä. Toisin sanoen ihmisille jotka aikovat käyttää pariohjelmointia hyväkseen. Kolmas kappale on omistettu ajankäyttöön liittyville seikoille ja siihen, miten ohjelmistotuotannon eri vaiheissa aikaa hävitään ja toisissa vaiheissa taas voitetaan verrattuna yksinohjelmointiin. Luvun neljännessä kappaleessa tutkitaan mitä ongelmia liittyy pariohjelmoinnin käyttöönottoon.

2.2.1 Taloudelliset resurssit

Pariohjelmointi vaatii tiettyjä taloudellisia resursseja eli satsauksia. Eräs suurimmista vaadituista resursseista on kohonnut palkkakustannukset. Pariohjelmointi vaatii kaksi ohjelmoijaa yhdelle koneelle, joten voidaankin sanoa että ohjelmointipuolella palkkakustannukset nousevat jonkin verran (eivät kuitenkaan suoranaisesti kaksinkertaistu). Kustannukset nousevat yleensä juuri ohjelmointivaiheessa, jolloin pariohjelmointina suoritettu työ hiukan pitkittyy verrattuna kahteen henkilöön, jotka ohjelmoisivat eri koneilla. Yritysten johto näkee usein ohjelmoijat kustannuseränä ja taakkana ja pitävät kaksinkertaistettua ohjelmoijien määrää resurssien haaskauksena. Monesti heiltä jää huomaamatta pariohjelmoinnin kokonaisvaikutukset. Palkkakuluja kun voidaan säästää jonkin verran testauksen ja virheiden korjaamisen osalta. Pariohjelmoinnissa tuotettu ohjelmakoodi on, kuten edellä esitetyistä tutkimuksista kävikin ilmi, yleensä laadukkaampaa. (Kessler ja Cunningham, 2000)

2.2.2 Henkilöstöön liittyvät resurssit

Jos yrityksessä tai yhteisössä on totuttu tekemään työtä perinteisimmillä tavoilla, saattaa muutosvastarinta pariohjelmointiin siirryttäessä olla suurta. Moni ohjelmoija ei välttämättä pidä tavasta jossa työskennellään pareittain, vaikka toimisikin muuten

työssään osana ryhmää. Eräs ongelma voi myös syntyä tilanteessa, jossa parina on kokematon ja toisaalta erittäin kokenut ammattilainen. Vaikka tämän pitäisi toisaalta olla hyvä asia siinä mielessä, että nyt kokeneempi voisi opettaa vähemmän kokeneelle uusia asioita, niin toisaalta tässä saattaa piillä myös katastrofin ainekset. Tällöin nimittäin pari on myös samaan aikaan tiedoiltaan ja taidoiltaan epätasa-arvoisessa asemassa ja kokeneempi työntekijä voi tuntea olonsa turhautuneeksi. Tämä ei tietenkään ole hyväksi pariohjelmoinnin eräänä etuna pidetylle opettavalle toiminnalle, vaan nyt se onkin kääntynyt itseään vastaan. Tämä mainittu huono puoli on ilmennyt osassa tutkimuksista, joissa haastatellut ammattilaiset työntekijät ovat valittaneet joissakin tapauksissa parin olevan eräänlainen taakka. Näin usein juuri silloin kun pari on tiedoiltaan ja taidoiltaan suuresti eriarvoinen. Toisaalta kaikki henkilöt eivät välttämättä edes sovellu syystä tai toisesta pariohjelmointiin. Osa haluaa varta vasten työskennellä yksin ja suosii muutenkin yksinohjelmointia. Jotkut näkevät että ohjelmakoodi on niin henkilökohtaista, että sen käsittely kuuluisi vain tekijänsä vastuulle. Toiset taas ovat huolissaan pariohjelmoinnissa koodin versioinnin hallinnasta, koska he pelkäävät parityöskentelyssä versioinnin vaikeutuvan parityön takia. (Lui & Chan, 2006)

2.2.3 Aikatauluun liittyvät resurssit

Pariohjelmoinnissa pelkkä ohjelmakoodin kehitysaika pitkittyy suhteellisesti eräiden tutkimusten mukaan (Succi, G. Marchese, M) noin 15 prosenttia. Tämä saattaa johtaa siihen, että vaikka pariohjelmoinnilla saavutetaankin parempilaatuista jälkeä, voidaan esimerkiksi kilpailutilanteesta johtuen menettää kriittistä aikaa joka tarvittaisi markkinoiden valtaamiseen. Tämä ylimääräinen kulunut aika tosin varmistaa sen, että ohjelma on laadukkaampi ja virheettömämpi kuin yksinohjelmointina tehty. Tästä voidaankin päätellä, että ohjelman ylläpito myöhemmin on helpompaa ja virheitä löydetään käytössä vähemmän pariohjelmointia hyväksikäyttäen tehdyistä ohjelmista. Pariohjelmoinnissa hyvin laaditun aikataulun merkitys korostuu, koska XP ei katso hyvällä ylitöitä. Niinpä parityössä aikataulu on laadittava siten, että kiiretilanteessa sitä

pystytään juostavasti muokkaamaan. Tämä ei ole aina kilpailutilanteessa helppoa. (Succi & Marchese, 2001)

2.2.4 Käyttöönottoon liittyvät ongelmat

Pariohjelmoinnin käyttöönottoaminen ei sinälläänkään ole aina välttämättä helppoa. Monet yritykset pitävät jo yleisellä tasolla työskentelytavasta, jossa työtä tehdään yksin (eli yksi henkilö työskentelee yhden koneen äärellä). Yritykset ja yhteisöt ovat tottuneet tähän vuosien aikana. Usein ainakin niin, että jokaisella työntekijällä on oletusarvoisesti oma tietokone, oma työhuone ja niin edelleen. Tällaiseen ympäristöön pariohjelmoinnin tuominen saattaa olla vaativa urakka. Ensinnäkin yhteisistä työtavoista on sovittava koko yrityksen kesken ja työntekijöiden on oltava valmiit työskentelemään määrättyllä tavalla (esimerkiksi XP:n määrittelemien työtapojen mukaan). Yrityksen tai yhteisön johdon on myös sitouduttava tekemään työtä sen menetelmän mukaan, joka on kulloinkin käytössä. Tämä tarkoittaa sitä, että johdon on alusta asti tuettava käytössä olevaa menetelmää, tehtävä laite- ja ohjelmistohankinnat sen mukaan ja niin edelleen. (Succi & Marchese, 2001)

Pariohjelmoinnin käyttöönottohetkellä osa yrityksen työntekijöistä on mahdollisesti hiukan erilaisessa asemassa ja suoraan pariohjelmointiin siirtyminen ohjelmoijien keskuudessa saattaa herättää keskustelua. Eräs uusi asia on työskentelytapa, pariohjelmointi, mutta myös siihen liittyvä korkeamman tason menetelmä, kuten XP. Tämä menetelmä saattaa muuttaa montakin rutiininomaista työtapaa prosessissa erilaiseksi kuin mihin yrityksen sisällä on aiemmin totuttu. Ja muutosvastarinta on lisäksi yleensä tutkimusten mukaan suurta. (Succi & Marchese, 2001)

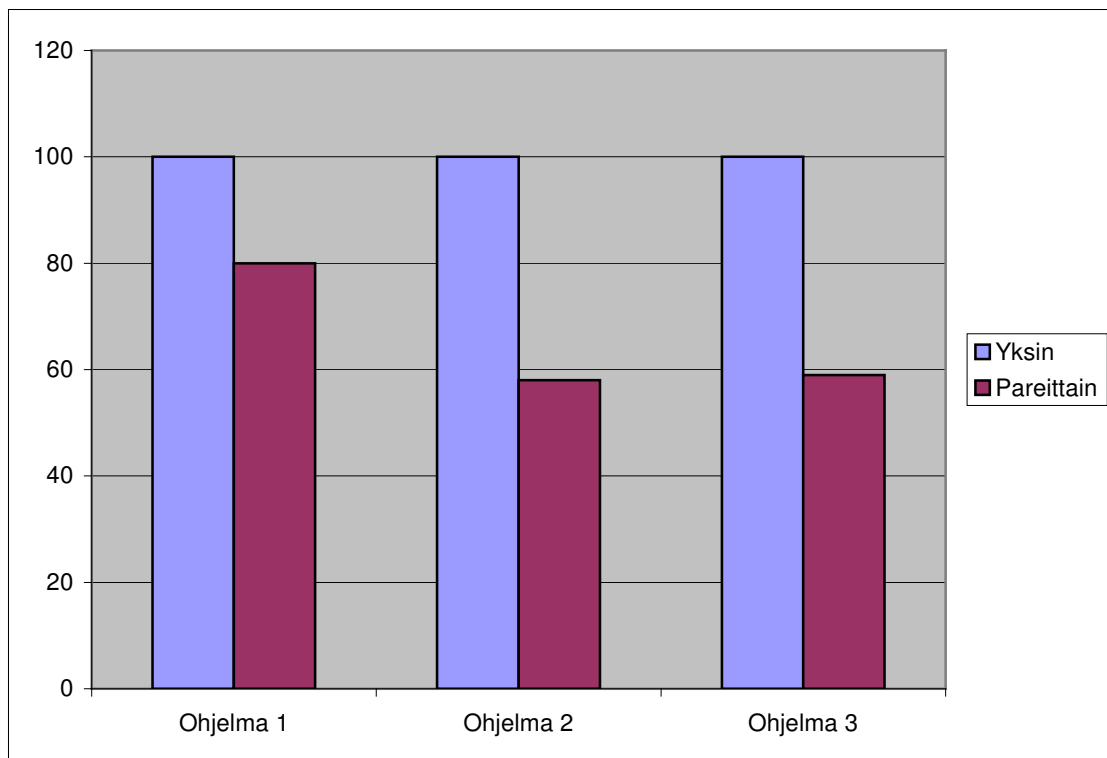
Toinen seikka mikä aiheuttaa todennäköisesti ongelmia on se, että useimmat työntekijät ovat koulutettu siten, että heillä ei usein ole minkäänlaista käsitystä pariohjelmoinnista tai he ovat jo unohtaneet sen mistä pariohjelmoinnissa varsinaisesti on kyse. Pariohjelmointia ei nimittäin kovinkaan useasti opeteta korkeakouluissa tai kouluissa yleensä sellaisella syvällisyydellä kuin yksinohjelmointia. Näin pariohjelmoinnin

peruskäsitteet ja toimintatavat ovat usein varsin heikoissa kantimissa yrityksen työntekijöiden keskuudessa. Tämän takia yksinohjelmointi tuntuukin luonnollisemmalta vaihtoehdolta tehdä töitä. (Muller, 2006)

2.2.5 Työn tehokkuus

Seuraavalla sivulla olevassa kuvassa 2 nähdään kuinka Laurie Williamsin Utahin yliopistossa järjestämässä kokeessa (Succi, G. Marchese, M) saatiin pariohjelmointia ja yksinohjelmointia vertailtaessa konkreettisia tuloksia työn tehokkuuteen liittyen. Tutkimuksessa tutkittiin kokeneita ohjelmistopuolen opiskelijoita, joille annettiin tehtäväksi ohjelmoida eräs ohjelma. Kolmannes luokasta teki tehtävän yksin ja kaksi kolmasosaa pareittain. Tutkimuksessa jokainen opiskelija kirjasi käyttämänsä työtunnit tietokantaan aina säännöllisin väliajoin. Tutkimuksen mukaan miestyötunnit lisääntyivät parityöskentelyssä vain 15 prosenttia. Eli jos yksittäisellä henkilöllä kului aikaa ohjelman tekemiseen 10 tuntia, meni parilla aikaa sen suorittamiseen 5 tuntia ja 45 minuuttia. Näin ollen tutkimuksen mukaan parityöskentelyssä muodostuneet kulut eivät olisi nousseet 50 % vaan 15 %. Toisaalta työn tekemiseen olisi samalla kulunut suhteellisesti pitempi aika ja tämä olisi aiheuttanut aikatauluun lisävaatimuksia. Tämä kaikki kuitenkin vain ohjelmoinnin osalla. Tässä yhteydessä tulee nimittäin vielä huomioida, että parin työntulos paranee koska virheiden määrä todistetusti laskee. Näin ollen aikaa säästetään myöhemmin testauksen ja etenkin virheiden korjauksen osalla. (Succi & Marchese, 2001)

Työn tehokkuuteen vaikuttavat monet asiat kuten parantunut työmotivaatio, suunnittelun tehokkuus ja virheiden määrän väheneminen. Tulos tosin osoittaa, että parityöstä saatu etu vaihtelee tapauskohtaisesti ja se ei ole missään nimessä täysin vakio. (Succi & Marchese, 2001)



Kuva 2. Työtuntivertailu yksinohjelmointia ja pariohjelmointi käytettäessä. (Succi & Marchese, 2001)

Yksinkertaisissa ja suoraviivaisissa töissä pariohjelmoinnin tuoma etu ei ole näin selvä, itse asiassa etua ei muodostu välttämättä lainkaan, mutta laaja-alaisissa ongelmissa ja paljon suunnittelua vaativissa tehtävissä pariohjelmoinnin hyödyt korostuvat entisestään. (Succi & Marchese, 2001)

2.2.6 Työskentelytavat

Yksinohjelmointi on kiistatta suosituin työskentelytapa ohjelmistoalalla tällä hetkellä. Syynä tähän on usein se, että pariohjelmointi työskentelytapana on vielä kuitenkin varsin tuntematon monelle alan ammattilaiselle. Vaikka työntekijät olisivatkin kuulleet pariohjelmoinnista koulutuksensa jossain vaiheessa, ei heidän käytännön kokemus ole välttämättä vielä sillä tasolla, että he haluaisivat työskennellä pariohjelmointia

hyväksikäyttäen. Myös tottuneisuus vanhaan tapaan ja muutosvastarinta ovat syitä sille miksi usein on päädytty yksinohjelmointiin. Pariohjelmointi tuntuu monelle vielä uudelta ja vieraalta. Myös yrityksen johdon on pariohjelmointiin siirryttäessä oltava valmis tukemaan kyseistä työtapaa. Tähän sisältyy moniakin seikkoja kuten henkilöstön koulutus. Edellä olleiden tutkimusten perusteella monet jotka ovat kokeilleet pariohjelmointia, ovat todenneet sen olevan motivoiva tapa tehdä töitä. Se on myös toisaalta opettavainen tapa, jonka avulla voidaan ratkoa suuriakin ongelmia. (Succi & Marchese, 2001)

2.2.7 Koulutukseen liittyvät yksityiskohdat

Henkilöstön koulutuksessa on vuosikausia keskitytty yksinohjelmointiin ja se onkin toki usein luonnollisin tapa tehdä ohjelmointitehtäviä. Sen sijaan esimerkiksi suunnittelu ja testaus on totuttu tekemään suuremmissa ryhmissä. Pariohjelmointia ei opeteta kovinkaan useasti kouluissa ja silloinkin kun sitä opetetaan, tapahtuu se yleensä pintapuolisesti ja siihen satsataan suhteessa vähemmän aikaa kuin mitä yksinohjelmointiin. Näin ollen onkin selvää, että valmistuneet alan ammattilaiset pitävät yksinohjelmointia kaikesta ”luonnollisempana” vaihtoehtona tehdä työtä. Tämä taas lisää muutosvastarintaa pariohjelmointia kohtaan työpaikoilla, koska vanhat tavat ovat tuttuja ja niihin on totuttu vuosien varrella. Pariohjelmointi vaatisikin tutustumista extreme programming – ohjelmistonkehitystapaan ja samalla koulutusta liittyen parisuunnittelun ja pariohjelmointiin. (Kessler & Cunningham, 2000)

3 VISUAALINEN ATTENTIO

Tässä luvussa tutustutaan visuaaliseen attention eli huomiokyvyn kohdistamiseen näkökentässä. Sitä tarvitaan myöhemmin kun perehdytään silmänliiketutkimukseen. Luvun alussa käydään läpi hieman visuaalisen attention taustaa ja sen jälkeen tarkastellaan ihmisen näkökykyä, silmän rakennetta ja tiedon välittymistä silmien kautta aivoihin. Silmän rakenne ja toiminnot esitellään siinä määrin kuin on tarpeellista katseenseurannan osalta. Tulemme myös huomaamaan että silmänliiketutkimus on läheisessä yhteydessä psykologiaan. Luvun lopussa pureudutaan visuaalisen attention toimintaan ja siihen miten se liittyy ihmisen kognitiivisiin prosesseihin. Lopussa vielä selvitetään miten esimerkiksi erilaiset ennalta annetut tehtävämääritykset vaikuttavat katseen liikkeeseen saman kuvan kohdalla eri tehtävänannon ollessa kyseessä.

3.1 Taustaa

Visuaalinen attentio tarkoittaa sitä visuaalista huomiokykyä joka ihmisellä on kun tämä havainnoi tiettyjä, jollakin tavalla häntä kiinnostavia objekteja ympäriltään ja keskittää huomionsa johonkin tiettyyn objektiin tai objektiryppääseen. Havaitseminen taas tarkoittaa aistien (näkö-, kuulo-, haju-, maku- ja tuntoaisti) havaintomaailman tuottaman tiedon vastaanottamista, sekä järjestelyä ja tulkintaa. Tulemme myöhemmin huomaamaan miten havaitsemista tutkitaan kognitiotieteissä, joissa tutkimus nojautuu osin psykologiaan ja osin filosofiaan. (Wolfe Jeremy M. 2000)

Ihminen aistii silmillään visuaaliset ärsykkeet, joista eräs on valo. Valoa tarvitaan myös objektien näkemiseen. Näkökentässä olevat objektit heijastavat ympäristöstä saamaansa

valoa katsojan silmiin, jolloin aivoihimme välittyy kuva objektien rakenteesta. Visuaalinen attentio ei siis tarkoita pelkästään näkökykyä, vaan sitä kognitiivista prosessia (näistä puhutaan tarkemmin kohdassa 3.4) jossa huomiokyky keskitetään tiettyyn objektiin koko näkökentässä ja samalla muut näkökentässä olevat objektit jätetään vähemmälle huomiolle. Aivoissa visuaaliseen attentionon vaaditaan tiedon prosessointia eli kognitiivisia prosesseja. Visuaalisen attention avulla ihminen esimerkiksi erottaa tai valitsee tietystä joukosta haluamansa objektin. Käytännössä tämä tapahtuu vaikkapa silloin kun etsimme isosta ihmisjoukosta tuttuja kasvoja. Tai käydessämme kaupassa, jolloin joudumme etsimään hyllystä juuri haluamme tuotteen muiden vastaavien tuotteiden joukosta. Käytämme siis visuaalista huomiokykyä päivittäin. (Wolfe Jeremy M. 2000)

3.2 Näköaisti

Näköaisti on yksi ihmisen viidestä perusaistista, jonka avulla ihminen aistii visuaalisesti ympärillään olevia objekteja. Näköaistiin vaikuttaa merkittävästi saatavilla olevan valon määrä. Valon määrän noustessa ihmisen kyky nähdä objekteja ympärillä kasvaa, koska esineet heijastavat valoa. Oikeastaan voimme sanoa, että näemme objektin sijasta valosäteet jotka heijastuvat objektista. Täysin pimeässä tilassa ihminen ei näe ympärilleen. (Austin Roorda, 2002)

Muita näköaistiin vaikuttavia tekijöitä ovat esimerkiksi henkilön ikä, silmäsairaudet, sekä silmän taittovika jota pyritään korjaamaan silmälaseilla. Näkökykyyn eli kykyyn nähdä ympärillä olevia visuaalisia ärsykeitä liittyy myös muita muuttujia. Objektien koko ja etäisyys katsojasta vaikuttavat niiden näkyvyyteen. Näin ollen näköaisti liittyy hyvin läheisesti visuaaliseen attentionon. Meidän on ensin nähtävä ympärillemme, että voisimme sen jälkeen huomioida näkökentästämme asioita ja prosessoida niitä aivoissamme. (Austin Roorda, 2002)

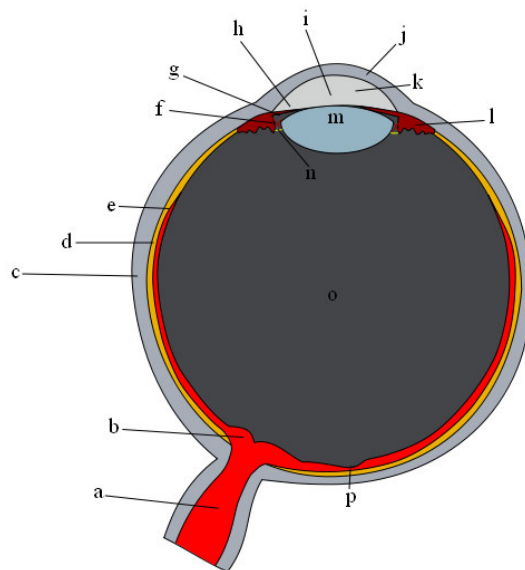
3.3 Näköjärjestelmä

Näköjärjestelmäksi kutsutaan sitä osaa ihmisen hermojärjestelmästä joka mahdollistaa elimistöltä näkemisen. Näköjärjestelmä tulkitsee tietoa näkyvästä valosta ja muodostaa kuvaa ihmisen näkökentästä. Näkökenttänä toimii yhden tai molempien silmien näkemä alue, joka on normaalisti molemmilla silmillä vaakasuunnassa noin 180 astetta ja pystysuunnassa noin 130 astetta (pystysuunnassa näkökenttää rajoittavat posket, nenä ja otsa) (Ditchburn, 1973). Tutustutaan seuraavaksi lähemmin silmän rakenteeseen ja toimintaan osana näköjärjestelmää.

3.3.1 Silmän rakenne ja toiminta

Ihminen aistii valoa näköelimillä joita kutsutaan silmiksi. Silmiin pääsevää valoa säädellään himmentimenä toimivassa värikalvossa, (kts. kuva 3) jonka aukko on mustuainen, toiselta nimeltään pupilli. Silmässä on useita valoa taittavia osia. Niitä ovat sarveiskalvo, lasiainen ja mykiö toiselta nimeltään linssi. Ihmisen silmän rakenne on kolmesta kalvosta muodostuva silmämuna, joka on kiinni silmäkuopassa näköhermon ja silmälihaksien avulla. (Roorda, Austin 2002)

Tarkastellaan seuraavaksi tarkemmin silmän rakennetta. Kuvasta 3 käyvät ilmi silmän eri osat. Näitä ovat: a) näköhermo, b) näköhermon nysty (näköhermon silmänsisäinen osa, jonka keskeltä haarautuvat verkkokalvon keskusvaltimo ja -laskimo), c) kovakalvo d) suonikalvo, e) retina, f) mykiön ripustinsäikeet, g) takakammio, h) iiris eli värikalvo, i) pupilli eli mustuainen, j) sarveiskalvo, k) etukammio (kammionesteen täyttämä), l) sädelihäs (supistuminen sallii mykiön ripustinsyiden löyhtymisen ja mykiön taittovoiman lisääntymisen), m) linssi eli mykiö, n) kannatin; ripustin, o) lasiainen, p) fovea eli tarkan näön alue



Kuva 3. Silmän rakenne (Wikipedia, the free encyclopedia, 2007b)

Ihmisen näköjärjestelmä toimii yksinkertaisesti ilmaistuna siten, että silmät aistivat valoa ja kuljettavat tietoa nähdystä ympäristöstä eteenpäin aivoihin. Tarkalleen ottaen tapahtuu seuraavaa. Ensin valo läpäisee sarveiskalvon (*cornea*), jonka jälkeen valo läpäisee mustuaisaukon eli pupillin. Pupilli valvoo verkkokalvolle pääsevän valon määrää niin, että vähäisessä valossa pupilli laajenee ja kirkkaassa valossa pienenee. Silmän linssi tarkentaa katsottavan objektin silmän takaosassa sijaitsevalle verkkokalvolle. Verkkokalvolla sijaitsee 127 miljoonaa valoherkkää solua, joista 120 miljoonaa on sauvasoluja ja 7 miljoonaa tappisolua. Sauvasolut havaitsevat muutokset valon määrässä, mutta eivät välitä väri-informaatiota. Tappisolut sen sijaan välittävät väri-informaatiota, mutta eivät ole niin herkkiä havaitsemaan valoerojen vaihteluita. Tappisoluja sijaitsee enimmäkseen verkkokalvon keskellä olevalla pyöreällä alueella (*macula lutea*). Siellä sijaitsee myös hyvin pieni painauma, jota sanotaan tarkan näkemisen alueeksi (*fovea*). Näköaistimus tapahtuu varsinaisesti vasta aivoissa, kun sinne saapuu tietoa hermoimpulsseina näkötietoa pitkin aistinsolujen välityksellä. (Roorda, Austin 2002)

3.3.2 Silmänliikkeet

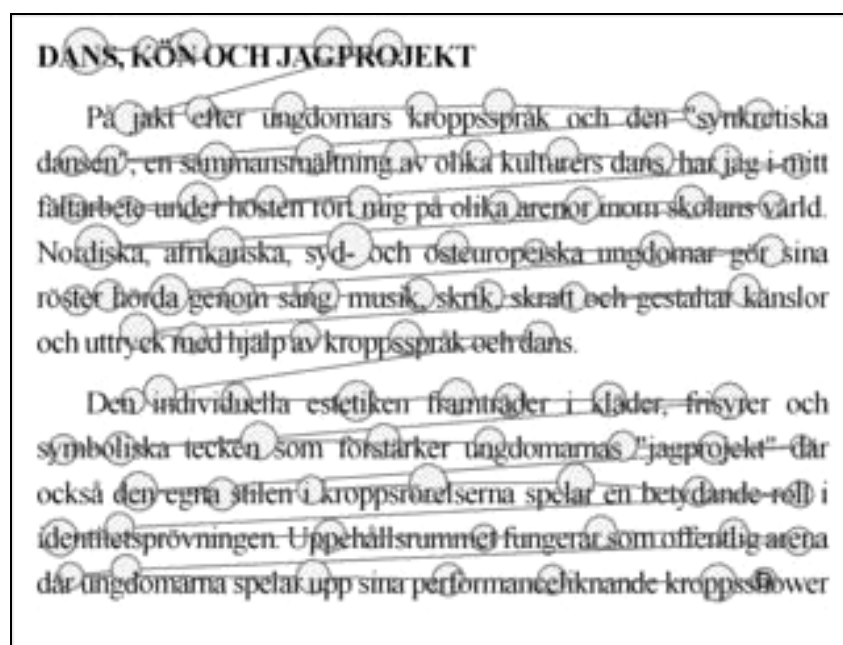
Silmänliiketutkimukseen liittyvässä kirjallisuudessa silmänliikkeet on usein jaoteltu eri tavalla osiin, riippuen silmänliikkeen tyyppistä. Katsotaan seuraavaksi joitakin jaotteluita, joista on hyötyä katseenseurannassa. Ensimmäisessä perusjaottelussa (Jack Kanski, 2003) on silmän liikkeet jaettu kolmeen eri luokkaan seuraavasti:

- **Duktiot** (ductions) Yhden silmän liike.
- **Versiot** (versions) Kummankin silmän kääntyminen yhtäaikaisesti samaan suuntaan.
- **Vergenssit** (vergences) Silmien kääntyminen siten, että molemmat silmät kääntyvät eri suuntiin.

Edellä esitelty jaottelu kertoo karkeasti miten eri tavoin silmät voivat liikkua. Tarkempi jako on tarpeen kun puhutaan silmänliikkeistä katseenseurannassa. Tällöin tarkastellaan silmänliikkeistä jotka koostuvat pitkälti kiinnittymisistä kohteeseen ja hypähdyksistä. Kiinnittymisestä kohteeseen käytetään termiä *fiksaatio*. Fiksaation aikana nähdään tarkasti noin yhden asteen kulmassa näkyvä objekti. Se vastaa silmästä 57 senttimetrin etäisyydessä olevaa senttimetrin kokoista aluetta (tätä asteen kulmaa käytetään hyväksi myöhemmin kun erotellaan fiksaatioita toisistaan). Fiksaatio on kestoltaan tyypillisesti n. 200 – 600 millisekuntia. (Lehtinen, 2005)

Fiksaatioiden aikana katseen kohteet pysyvät näkökentän kiinnostuksen kohteena (Martinez-Conde & al., 2004) ja samalla tapahtuu myös tiedon prosessointi aivoissa. Fiksaatioiden välisiä silmänliikkeitä kutsutaan *sakkadeiksi* tai hypähdyksiksi. Niiden (Lehtinen, 2005) aikana molemmat silmät liikkuvat samansuuntaisesti yhtä aikaa. Sakkadit kestävät yleensä n. 25-100 millisekuntia.

Alla olevassa kuvassa 4 nähdään miten fiksaatiot ja sakkadit ilmenevät kun on mitattu silmänliikkeitä luettaessa tekstiä. Kuvassa ympyrät ovat fiksaatioita, ympyröiden säteet kuvaavat fiksaatioiden kestoja ja ympyröiden välissä olevat viivat ovat sakkadeja. Fiksaatioiden aikana katse kiinnittyy tekstin tiettyyn kohtaan lyhyeksi ajaksi ja sakkadien aikana katse liikkuu tekstinsuuntaisesti. Myöhemmin seuraavassa luvussa käymme tarkemmin läpi miten katse käyttäytyy lukemisen aikana. Lisäksi silmänliikkeistä löytyy käsite *sakkadinen suppressio*, joka tarkoittaa tilannetta jossa aivot tukkivat valikoiden visuaalisen prosessoinnin silmänliikkeiden aikana tavalla, jossa sekä silmän liike että väli näkökyvyn aistimuksessa ei ole henkilön kannalta huomattavissa.



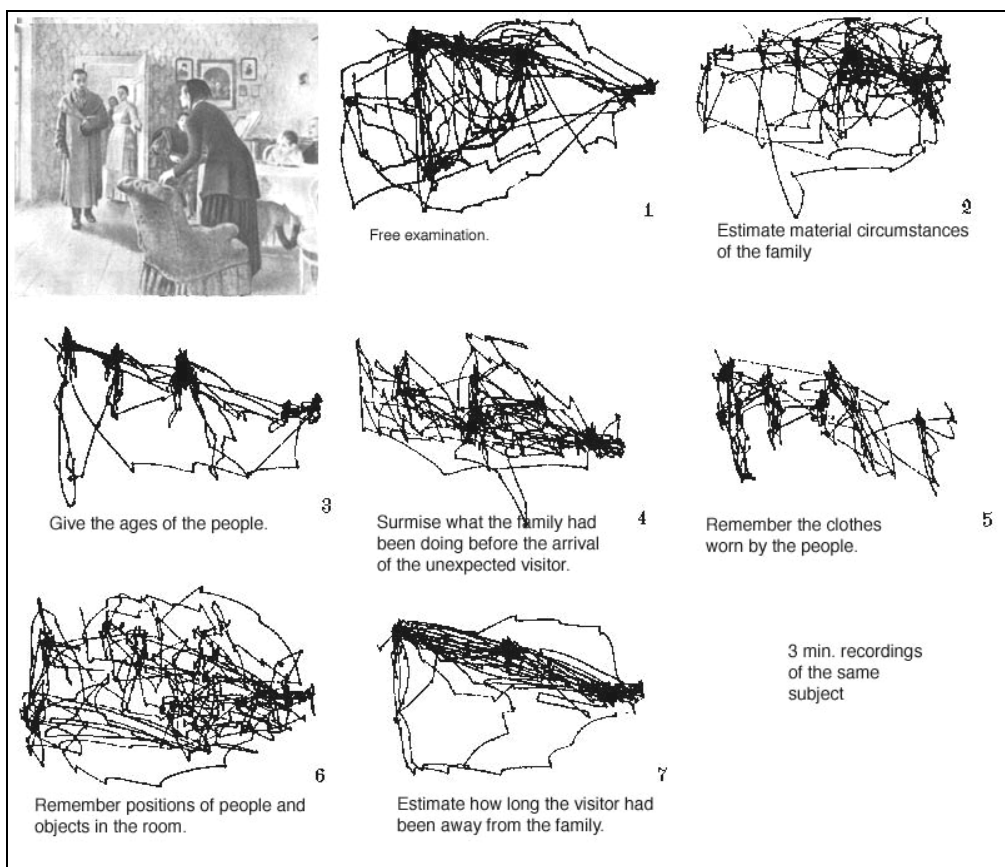
Kuva 4. Fiksaatioiden ja sakkadien ilmeneminen tekstiä luettaessa
(Wikipedia, the free encyclopedia, 2007a)

Visuaalisen tiedon välittyminen näköjärjestelmään tapahtuu fiksaatioiden aikana. Sen sijaan sakkadien aikana informaation välittyminen on estynyt. Mielenkiintoista on huomata että fiksaatioiden aikana silmät keräävät tietoa myös näkökentän ääreisalueilta ja seuraava sakkadi suunnataan seuraavaan kiinnostavaan informaatioon jotta havainto tarkastelun alla olevasta kohteesta syvenisi. Liikkuvaa kohdetta seuratessaan silmät liikkuvat sakkadien lisäksi tasaisesti. Näitä tasaisia liikkeitä kutsutaan *tavoitteleviksi*

silmänliikkeiksi ja ne ovat täysin tahdosta riippumattomia. Niiden avulla katseen kohde voidaan pitää vakaana verkkokalvolla. On myös olemassa muita silmänliikkeitä kuin edellä esitelty, mutta niiden merkitys katseenseurannassa ja käytettävyytutkimuksessa on vähäistä. (Lehtinen Merja, 2005)

3.3.3 Ennakoivat silmänliikkeet

Silmänliikkeisiin vaikuttaa osaltaan myös annettu tehtävä. Testihenkilölle, jolle on annettu katsottavaksi tietty kuva, voidaan samalla antaa myös tehtäväksi huomioida tiettyjä asioita kuvasta. Tällöin katseen kulussakin huomataan muutoksia. Kuvassa 5 (kirjasta Yarbus, A.L., *Eye Movements and vision*) nähdään esimerkin avulla miten tämä toimii käytännössä. Kuvassa ylhäällä vasemmalla on piirros jota testissä on katsottu. Kohdassa 1 on testihenkilön annettu katsoa kuvaa vapaasti. Muissa kohdissa on määritelty ennalta jokin tehtävä. Kohdassa 2 on testihenkilöä pyydetty arvioimaan piirroksessa olevan perheen taloudellinen tilanne. Kohdassa 3 on testihenkilöä pyydetty arvioimaan piirroksessa olevien ihmisten iät. Tällöin katse näyttää kohdistuneen pitkälti piirroksessa oleviin henkilöihin ja heidän kasvoihin. Kohdassa 4 testihenkilöä on pyydetty veikkaamaan, mitä perhe on ollut tekemässä ennen odottamattoman vieraan saapumista. Kohdassa 5 testihenkilöä on pyydetty muistamaan millaiset vaatteet piirroksen henkilöillä on yllä, jolloin katse on käynyt läpi piirroksen henkilöiden vaatteet. Kohdassa 6 muistamaan piirroksessa olevien ihmisten ja esineiden sijainti ja kohdassa 7 arvioimaan miten pitkään vieras on ollut erossa piirroksen perheestä.



Kuva 5. Silmien liike eri tehtävänantojen mukaan (Yarbus, 1967)

Kuvan 5 perusteella voidaan todeta, että kun katseenseurantaan osallistuvalla testihenkilöllä annetaan ennalta määritelty tehtävä ennen kuvan katsomista, osaa hän hakea katseellaan olennaisen tiedon kuvan sisältä suuntaamalla katseen olennaisina pitämiinsä asioihin kokonaistehtävän kannalta. Testihenkilö siis ennakoii silmänliikkeitään. Tästä syystä esimerkiksi käytettävyyystutkimuksessa voi testitulokset riippua siitä, annammeko testihenkilölle jonkin ennalta määritellyn tehtävänannon vai suorittaako testihenkilö vapaata katsomista.

3.4 Kognitio

Kognitio on psykologiaan ja kognitiotieteisiin liittyvä käsite (Neisser, 1982) jonka avulla käsitellään erityyppisiä tietotoimintoja. Kognitiivisiin toimintoihin kuuluvat esimerkiksi erilaiset tiedon vastaanottamiseen, tallentamiseen, käsittelyyn ja käyttöön liittyvät prosessit. Tällaisia toimintoja ovat ihmisellä esimerkiksi kyky värien ja muotojen tunnistamiseen. Ihmisen kognitiiviseen suoriutumiseen vaikuttavat mm. koulutus, älykkyys ja yleinen vireystila. Ihminen voi harjoitella ja kehittää kognitiivisia prosessejaan. Esimerkiksi kirjoittaminen ja lukeminen ovat monimutkaisia, vaativia ja useista eri osatekijöistä koostuvia kognitiivisia toimintoja. Tässä yhteydessä keskitytään lähinnä näköaistiin ja silmänliikkeisiin liittyviin kognitiivisiin prosesseihin.

Silmänliikkeitä tutkittaessa (Neisser, 1982) on kiinnitetty huomiota siihen, miten esimerkiksi fiksaatioiden kesto vaikuttaa tiedon omaksumiseen. Tutkimuksissa on huomattu, että tehokkaan oppimisen ja fiksaatioiden kestojen välillä on tietty yhteys. Tehokkaiden oppijien fiksaatiot olivat kestoiltaan pitempiä kuin heidän, jotka eivät olleet niin tehokkaita oppimaan. Huomattavaa on kuitenkin, että kaikkien henkilöiden osalta fiksaatiot pitenevät kun kyseessä on vaikea oppimistehtävä. Samoin käy myös sakkadien kohdalla, joiden määrä kasvaa vaativissa harjoituksissa. Myös muilla silmänliikkeillä on yhteyttä kognitiivisiin prosesseihin. Tällaisia silmänliikkeitä ovat pupillin koon muuttuminen ja silmän räpäytykset.

3.5 Visuaalinen havaitseminen

Visuaalinen havaitseminen on - kuten jo edellä olevassa kappaleessa todettiin - kognitiivinen prosessi. Psykologiassa on esitelty (Neisser, 1982) erilaisia

havaitsemisteorioita, joilla pyritään ymmärtämään sitä tapahtumaketjua, joka käydään läpi ihmisen mielessä kun tämä havaitsee asioita ympärillään. Tässä yhteydessä ei lähdetä käymään läpi psykologian eri havaitsemisteorioita sen tarkemmin, vaan todetaan joitain niitä yhdistävistä tekijöistä jotka omalta osaltaan vaikuttavat visuaaliseen attentionoon.

3.5.1 Skeemat

Tutustutaan ensin käsitteeseen *skeema*, joka on psykologiassa mielen sisäinen malli reaalimaailmasta (Neisser, 1982). Ihmiset järjestävät tietoa mielessään skeemoihin ja sisäistettyä skeemaa käytetään uuden tiedon ymmärtämiseen. Skeema on itsessään sisäisesti järjestynyt muistirakenne, joka kiteyttää ja pelkistää valtaisan määrää havaintoinformaatiota ja nopeuttaa informaatiosta tehtäviä johtopäätöksiä. Skeeman aktivoituessa kaikissa alajärjestelmissä aktivoituvat myös vastaavat reaktiot, kuten merkitykseen liittyvät fysiologiset reaktiot, tunteet, käyttäytyminen ja motivaatio. Käytännössä tämä voi tarkoittaa esimerkiksi sitä, että ihmisellä on käsitys siitä millainen on auto. Kyseisen skeeman avulla ihminen voi tunnistaa hyvin erinäköisiä autoja samaksi kulkuneuvotyyppiä. Skeeman avulla tiedostetaan myös auton toiminta eli että autolla voi ajaa. Henkilö voi myös yhdistellä skeemoja toisiinsa, jolloin tapahtuu niin sanottuja assimilaatioita (uuden tiedon nivomista vanhaan skeemaan).

3.5.2 Valikoiva katsominen

Visuaaliseen havaitsemiseen liittyy myös valikoiva katsominen, joka tarkoittaa sitä että ihminen pystyy valikoimaan näkemästään kahdesta päällekkäisestä informaatiolähteestä haluamansa. Neisser (1982) osoittaa kokeissaan, että ihminen voi katsoessaan kahta päällekkäin näkyvää erillistä videokuvaa, seuraamaan niistä toista ilman suurempia

ongelmia. Kyseessä on siis hyvin samantapainen ilmiö kuin mikä tapahtuu valikoivassa kuulemisessa. Ihminen voi kuunnella kahta puhujaa tai kahta eri radiokanavaa (olettaen että ne kuuluvat suurin piirtein samalla äänenvoimakkuudella, muuten toisen lähteen signaali peittäisi toisen) samanaikaisesti niin että keskittyy niistä vain toiseen ja saa siitä selvää. Mikäli ihminen joutuisi seuraamaan molempia videokuvia tai radiokanavia samaan aikaan, vaikeutuisi suoritus merkittävästi. Ihminen keskittyy hänen mielestään primääriin lähteeseen ja pyrkii sivuuttamaan sekundäärisen. Ihminen ei kuitenkaan sulje sekundääristä lähdettä pois mielestään, vaan sen sijaan omistautuu primääriselle lähteelle aktiivisesti. Tämä perustuu siihen että vain se tapahtuma jota henkilö seuraa, on mukana ennakoitien, tiedon vastaanottamisen ja tutkimisen kierrossa. Siksi tietty tapahtuma nähdään tai kuullaan. Sen takia Neisserin (1982) mukaan tarkkaavaisuus onkin itse asiassa havaitsemista.

Myös kahtaalle kohdistuva tarkkaavaisuus on Neisserin mukaan mahdollista, mutta tällöin ihmisen kapasiteetin rajat tulevat jossain kohtaan vastaan. Neisserin mukaan ihmisen kapasiteetille ei voida kuitenkaan asettaa mitään yksiselitteistä rajaa, koska ei ole olemassa mitään fysiologista tai matemaattisesti vahvistettua rajaa tiedolle, jonka voimme vastaanottaa kerralla. Kahtaalle kohdistuvan tarkkaavaisuuden ongelmaksi nousee yleensä se, että ihmisellä on vaikeuksia yhdistää toisiinsa useampaa tehtävää, joilla ei ole luonnostaan mitään yhteyttä toisiinsa.

4 KATSEENSEURANTA

Tässä luvussa käydään läpi katseenseurantaan liittyviä käsitteitä ja yksityiskohtia. Ensin esitellään silmänliiketutkimuksen ja katseenseurannan teoriaa tarvittavissa määrin siten että lukijalle selviää niihin liittyvät taustatiedot. Sen jälkeen kerrotaan millaisilla laite- ja ohjelmistokokoonpanoilla tässä tutkielmassa suoritettut mittaukset on tehty. Tämän jälkeen käydään läpi erilaiset protokollat joita on käytetty hyväksi tiedon mittaamisessa ja lopuksi suoritetaan vertailua käytössä olleiden laitteiden välillä.

4.1 Silmänliiketutkimuksen historiaa

Ranskalainen Louis Émile Javal (1839–1907) huomasi vuonna 1878 tutkimuksissaan, että silmänliikkeet tekstiä luettaessa eivät olekaan tasaista liikettä kuten tätä ennen oli yleisesti oletettu (Lehtinen, 2005). Javal huomasi silmien liikkeiden koostuvan lyhyistä pysähdyksistä (fiksaatioista) ja nopeista hypähdyksistä (sakkadit). Tämä nosti esiin uuden tutkimuskentän – katseenseurannan - jonka avulla haluttiin selvittää missä kohtaa tekstiä luettaessa katse pysähtyy, kuinka pitkäksi aikaa ja niin edelleen.

Silmänliiketutkimus on jaoteltu kolmeen aikakauteen Hyönä (1993), joista ensimmäinen oli 1900-luvun vaihteessa. Kyseinen aikakausi tunnetaan nykyisin varsin alkeellisista nauhoitustekniikoistaan, mutta jona aikana monet nykyään silmänliiketutkimuksessa peruskäsitteistöksi nousseet termit esiteltiin ensimmäistä kertaa. Näihin käsitteisiin kuuluvat esimerkiksi lukemisen aikana keskimääräinen fiksaation kesto, sakkadien pituudet ja kestot, sekä sakkadinen suppressio.

Toinen aikakausi rajataan 1900-luvun alusta saman vuosisadan puoleenväliin, jolloin silmänliiketutkimus oli hyvin läheisessä suhteessa psykologian ja erityisesti käyttäytymistieteen eli behaviorismin kanssa. Sinä aikana harjoitettiin sovellettua tutkimusta jossa pyrittiin esimerkiksi kehittämään tehokkaita silmänliikestrategioita, joiden avulla voitaisiin parantaa lukutaitoa.

Kolmas aikakausi silmänliiketutkimuksessa alkoi 1970-luvun alussa, jolloin apuun tulivat uudet nauhoitustekniikat ja tietokoneistettu tiedon näytteistäminen sekä analysointi. Tämän aikakauden ominaispiirteisiin kuuluu kiinnostus ihmisen tiedonkäsittelyyn joka taas on yhteydessä kognitiiviseen psykologiaan.

4.2 Katseenseurannan teoriaa

Silmänliiketutkimukseen kuuluu eräänä osana katseenseuranta. Joskus käsitteet silmänliiketutkimus ja katseenseuranta sotketaan keskenään, joten tehdään tässä vaiheessa selväksi niiden ero. Katseenseuranta on tapa jolla voidaan seurata, mitata ja kerätä tietoa ihmisen silmänliikkeistä, sekä katseen kulusta. Katseenseuranta on yksi osa silmänliiketutkimusta. Tärkein katseenseurannasta saatava hyöty on tieto ihmisen visuaalisen tarkkaavaisuuden kohteista. Eli kohteista joihin ihmisen tarkkaavaisuus on kohdistunut tietyllä hetkellä.

Tarkkaavaisuuden kiinnittyminen kohteeseen on kaksivaiheinen prosessi joka on jaettu *esi-tarkkaavaisuudeksi* (pre-attentive) ja *keskitetyksi tarkkaavaisuudeksi* (focal attention). Esitarkkaavaisuuden aikana ihminen havaitsee informaatiota koko visuaalisesta kentästä. Keskitetyn tarkkaavaisuuden aikana henkilö käsittelee yhtä tai korkeintaan muutamaa ärsykettä yhdessä. Katseenseurannan avulla voidaan mitata joko katseen huomiopistettä (*point of gaze*) tai silmänliikettä suhteessa päähän. (Lehtinen Merja 2005).

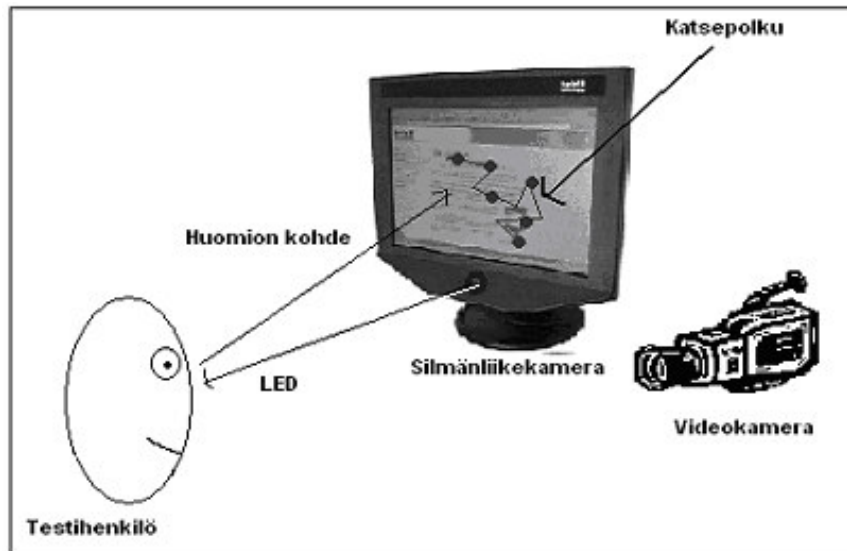
Tässä tutkielmassa kiinnostuksen kohteena ovat olleet fiksaatiot, sakkadit ja attention kohteet. Perehdymme myöhemmin tarkemmin näihin muuttujiin, kun teemme laitevertailua kahden eri silmänliikekameran välillä.

4.2.1 Katseenseuraja

Katseenseuraja (*eye tracker*) on laite, jolla voidaan tarkkailla testihenkilön katseen kohdetta ja liikettä. Katseenseurannassa käytetään nykyisin usein tähän tarkoitukseen varta vasten suunniteltuja silmänliikekameroita. Niitä löytyy erimallisia ja eri valmistajilta mutta käyttötarkoitus on niissä aina sama: tarkkailla miten ihmisen silmät liikkuvat ja mihin kohtaan silmien katse (etenkin attentio) kiinnittyy.

Toinen katseenseurannassa käytettävä apuväline on tavallinen videokamera, jonka avulla testitilanne voidaan kuvata. Videokamerassa olisi hyvä olla (niin kuin usein onkin) toiminto joka näyttää kellonajan kuvaushetkellä.

Kuvasta 6 käy ilmi millainen asetelma katseenseurantakokeessa yleensä on. Videokamera nauhoittaa tapahtumia testihenkilön katsoessa näytöllä olevia objekteja. Silmänliikekamera aistii silmänliikettä ja katseen sijaintia infrapunavalon heijastuman avulla.



Kuva 6. Asetelma katseenseurannassa

Silmänliikekamerasta on yhteys testihenkilön silmään ja silmänliikekamera rekisteröi silmänliikkeet. Samalla ohjelmisto tallentaa reaaliaikaisen POG:n (photo-oculography), josta katsepolku rekonstruoidaan myöhemmin.

4.2.2 Katseenseurannalla saatavat hyödyt

Katseenseurannan avulla saadaan tietoa tiedostamattomista kognitiivisista prosesseista, joita olisi muuten hankala selvittää. Katseenseurannan avulla saatuja tuloksia käytetään usein hyväksi käytettävyytutkimuksessa (Granka & al., 2004). Yleisesti on selvää että ihmisen ja tietokoneen välisessä kommunikoinnissa tietokoneen antamat tulosteet tulevat nopeammin, kuin mitä käyttäjän antamat syötteet erilaisten tiedonsyöttölaitteiden kuten näppäimistön, hiiren ja niin edelleen kautta. Käytettävyytutkimuksissa pyritään edistämään käyttäjäpuolen toimintaa, jotta kommunikointi tietokoneen ja ihmisen välillä nopeutuisi.

Etenkin käyttöliittymien suunnittelussa on katseenseurannasta huomattavaa apua. Tällöin kiinnostuksen kohteina ovat visuaalisen haun tehokkuus eli miten nopeasti henkilö löytää käyttöliittymästä tarvitsemansa asiat ja tarkkaavaisuuden kiinnittyminen eli havaitseeko käyttäjä olennaisemmat asiat vähemmän olennaisten joukosta. Saatujen tuloksien perusteella voidaan päätellä huomaako käyttäjä tietyn kohdan käyttöliittymässä ja saadaan tietoa siitä järjestyksestä jolla käyttäjä on hakenut ja prosessoanut käyttöliittymässä olevaa tietoa. (Lehtinen, 2005)

Katseenseurannan avulla saadaan myös selville tietoa siitä, miten nopeasti käyttäjä omaksuu ja oppii tietoa käyttöliittymästä. Tuloksien pohjalta käyttöliittymää voidaan parantaa niin, että käyttäjä löytää haluamansa tiedon helposti ja huomioi olennaiset asiat nopeasti. Itse katseenseurannassa optimaalinen tulos olisi sellainen, jossa katseella olisi suora linja ja mahdollisimman vähän sakkadeja, sekä katseen lyhyt kiinnittyminen kohteeseen. (Lehtinen, 2005)

4.2.3 Katseenseurannan haasteet

Katseenseurannalla on myös tiettyjä haasteita. Eräs niistä on se, ettei katseenseuranta anna vastausta kysymykseen miksi käyttäjä toimii tietyllä tavalla. Toinen ongelma on että vaikka käyttäjän katse onkin kiinnittynyt tiettyyn paikkaan käyttöliittymässä, ei se vielä takaa sitä että käyttäjä olisi ymmärtänyt tai muistanut kohteen sisällön. Myös koehenkilöiden valintakriteerit ja otoksen kattavuus saattaa vaikuttaa saatuun lopputulokseen. (Lehtinen, 2005)

Laitepuolelta ongelmaksi nousee yleensä se, että laitteistot ovat kalliita ja vaativat laboratoriotilat joissa niitä voidaan käyttää optimaalisimmin. Käytettävä laitteisto on yleensä jonkin verran kömpelöä, joten käyttäjän liikkumista joudutaan rajoittamaan. Tämä aiheuttaa sen että silmänliikkeiden tutkiminen käyttäjän luonnollisessa ympäristössä on vaikeaa. Lopuksi saadun datan analysointi voi olla aikaa vievää ja siten muodostua omalta osaltaan kalliiksi.

4.3 Videopohjainen tekniikka katseenseurannassa

Katseenseurannassa käytetään hyödyksi erilaisia tekniikoita, riippuen siitä millaisessa tutkimusympäristössä ollaan ja mitä katseenseurannalla mitataan. Yleisin ja tässäkin tutkielmassa suoritettujen mittausten taustalta löytyvä videopohjainen tekniikka on osoittautunut varsin hyväksi ja toimivaksi tekniseksi ratkaisuksi. Siinä silmänliikekamera fokusoidaan joko yhteen tai molempiin silmiin (tarkemmin silmän pupilliin ja sarveiskalvoon) ja sitten kamera nauhoittaa silmien liikettä. Apuna käytetään videokameraa, joka kuvaa tutkimustilannetta. Testeissä voi tapahtua niin monitorin videonauhoitusta, kuin videokameran nauhoitusta. Videokameran nauhoittaman videon perusteella voidaan myöhemmin tarkistaa millaisissa olosuhteissa testihenkilö on suorittanut annettua tehtävää ja mitkä asiat ovat voineet vaikuttaa saatuihin tuloksiin. Videokuvan perusteella on myös helppo tarkastaa myöhemmin milloin nauhoitus on alkanut ja loppunut, sekä millainen tehtävä on ollut kyseessä. Videolla näkyy usein myös kellonaika, josta voidaan seurata kokeen ajallista etenemistä ja tarkastella myöhemmin testitilannetta esimerkiksi tietyllä aikavälillä. Nykyisin testitilanteen nauhoittaminen voidaan hoitaa tietokoneen avulla, joka tallentaa testitilanteen videotiedostoksi, jota voidaan myöhemmin tutkia.

4.4 Silmänliikkeet lukemisen aikana

Seuraavaksi käsitellään silmänliikkeitä lukemisen aikana. Kuten aiemmin jo todettiin, pyritään katseenseurannan avulla saamaan tietoa koehenkilön silmänliikkeistä tietyn koetehtävän aikana. Tällaisia koetehtäviä voivat olla esimerkiksi staattisten pisteiden katsominen tai tekstin lukeminen. Koska tässä tutkielmassa koehenkilöille annettut

tehtävät muodostuvat pitkälti staattisten pisteiden katsomisesta, on hyvä tarkastella ensin sitä, miten silmät liikkuvat yleensä ottaen tekstiä luettaessa.

Katseenseurannassa on tutkittu jonkin verran sitä, miten katse käyttäytyy tekstiä luettaessa. Lukemisessa on yleensä kyseessä monimutkainen kognitiivinen prosessi, jossa katseen saamaa tietoa prosessoidaan aivoissa. Hyönä (1993) esittelee muutaman faktan silmänliikkeistä lukemisen aikana. Ensiksi hän toteaa että itse lukeminen tapahtuu pitkälti fiksaatioiden aikana, jolloin silmät pysyvät suhteellisen vakaina ja jolloin aivot prosessoivat tekstiä (tämä todettiin myös kohdassa 3.3.2., jossa tarkasteltiin lähemmin silmänliikkeitä). Sakkadien kesto on Hyönän mukaan noin kymmenen kertaa lyhyempää kuin fiksaatioiden kesto ja silloin silmät eivät omaksu tietoa samalla tavalla, vaan kuljettavat katsetta tekstissä eteenpäin ennakoiden tulevaa katseen kohtaa.

Hyönä on suorittanut kokeita joista käy ilmi että testihenkilöt ovat lukiessaan taipuvaisia tekemään fiksaatioita sanojen keskelle tavalla, jolla henkilö olettavasti tunnistaa parhaiten kyseisen sanan. Yhdyssanoissa mielenkiinto kohdistuu ensin niihin sanoihin, joiden informaatioarvo on lukijalle suurempi. Tämä myös sanoissa joissa informaatioarvo on suurempi lopussa olevassa sanassa. Ja tämä kaikki siitä huolimatta, että luetaan vasemmalta oikealle. Sanan sisäinen rakenne vaikuttaa Hyönän mukaan niin fiksaatioiden keston kuin niiden sijaintiinkin.

4.5 Huomioitavia asioita katseenseurannassa

Ennen kuin katseenseuranta voidaan aloittaa, on hyvä varmistua seuraavista (Goldberg, J.H. & Wichansky, A.M. 2003) seikoista jotka saattavat vaikuttaa mittauksissa saatuihin lopputuloksiin. Seuraavat asiat tulisi ottaa yleisesti huomioon laboratoriotestejä tehdessä. Myös tässä tutkielmassa on pyritty noudattamaan alla olevia sääntöjä.

- Pyri välttämään turhia valonlähteitä
Osallistujan silmään osuva kirkas valo saattaa aiheuttaa heijastuksia, jotka voivat sekoittaa katseenseurantalaitteen. Samoin voi käydä myös kirkkaiden ja heijastavien pintojen kanssa joita tulisi välttää laboratorio-olosuhteissa.
- Pyri välttämään kirkkaasti valaistuja kuvaruutuja
Kirkkaiden valojen ja näyttöjen käyttämisestä tulisi välttää, jotta testihenkilön pupilli olisi kooltaan mahdollisimman suuri. Valitettavasti videokameroiden hyvä kuvanlaatu vaatii usein kirkkaan valaistuksen. Tällöin joudutaan säätämään niin huoneen valaistusta, kuin myös infrapunavaloa niin, että yhteisvaikutus valaistukseen olisi mahdollisimman muuttumaton. Jos on käy niin että näyttöjen kirkkaus vaihtelee merkittävästi, voidaan huoneen valoisuutta kasvattaa pupillin koon muutosten tasaamista varten.
- Tarkista että kamera on asetettu oikein ja että siitä on esteetön näkymä testihenkilön silmään
Kamera asetetaan siten, että se kuvaa silmää kulmasta, joka on alempana kuin käyttäjän horisontaalinen näköpiiri. Kameraa ei kuitenkaan saa asettaa liian ylös tai alas.
- Sijoita kamera huolellisesti
Kamera asetetaan oikealle etäisyydelle siten, että siitä on esteetön näkymä testihenkilön silmän iirikseen ja pupilliin. Etäisyys kamerasta henkilöön tulee olla käytettävissä olevan silmänliikekameran ohjeen mukainen.
- Pyri minimoimaan päänliikkeitä
Käyttäjän pään tulee pysyä mahdollisimman hyvin paikoillaan niin kalibroinnin, kuin myös itse testin aikana. Käyttäjän on istuttava paikoillaan ja asennon tulee olla sellainen jossa käyttäjä tuntee olevansa luonnollisesti (niin ettei esimerkiksi selkä, hartiat tai niska rasitu). Joskus voidaan käyttää apuna myös leukatukea.
- Pyri välttämään silmälaseja tai piilolinsskejä käyttäviä testihenkilöitä

Yleensä ottaen voidaan todeta että mitä parempi näkökyky testihenkilöllä on, sitä paremmin kalibroinnin voidaan olettaa onnistuvan. Esimerkiksi piilolinssit voivat aiheuttaa valon heijastumista, mikä taas vaikuttaa valon heijastumiseen sarveiskalvolta.

4.5.1 Katseenseurannassa käytettävät laitteet

Katseenseurantaa varten on kehitetty laitteita aina 1800-luvun lopulta asti, jolloin Ranskalainen Louis Émile Javal aloitti silmänliiketutkimuksen. Laitteet kehittyivät vuosien saatossa ja niiden käytettävyys parani. Ensimmäiset katseenseurannassa käytetyt laitteet olivat nykyisiin verrattuina kömpelöitä ja vaativimpia testihenkilöiden kannalta. Alkuvaiheessa testihenkilöltä vaadittiin paljon (Oyekoya & Stentiford, 2006), sillä laitteista oli suora fyysinen kontakti silmään.

Silmänliiketutkimuksessa käytettyjen laitteiden kehitys kulki ja kulkee edelleen käsi kädessä muun teknisen kehityksen kanssa. Merkittäviä tekijöitä laitteiden kehityksessä olivat alussa kameratekniikan ja optisten laitteiden kehittyminen, myöhemmin tietokoneet, joihin voitiin kytkeä silmänliikekameroita ja joista saatua dataa voitiin käsitellä erilaisten ohjelmien avulla. (Oyekoya & Stentiford, 2006)

Nykyiset laitteet perustuvat pitkälti juuri silmänliikekameran ja tietokoneen, sekä ohjelmiston yhteisvaikutukseen. Laitteistopuolella on nykyään olemassa esimerkiksi päähän asetettavia silmänliikekameroita, jotka myötäilevät ihmisen pään liikettä tai pöydälle asettava kamera, joka seuraa katsetta jos siitä on esteetön kontakti ihmisen silmään. (Oyekoya & Stentiford, 2006)

Nykyisissä silmänliikekameroissa on yleensä infra-punavalo LED, josta heijastetaan valoa silmään. Silmänliikekamera on joskus kytkettynä niin sanottuun katseenseurantakontrolliyksikköön (*Eye tracker Control Unit*), joka tekee omasta

puolestaan laskentaa silmänliikekameronista saadun datan pohjalta. (Oyekoya & Stentiford, 2006)

Nykyinen kehitys pyrkii mahdollistamaan silmänliikekameronilta paremmin käyttäjän pään liikkeet (jolloin realisoituu käyttäjän toiminto oikeassa työskentely-ympäristössä) ja toisaalta olemaan riippumaton testihenkilöihin liittyvistä muuttuvista tekijöistä. Hyvän katseenseurantalaitteen ominaisuuksiin kuuluu, että laite ei ole riippuvainen testihenkilön silmien väristä tai muodosta, testihenkilön etnisestä taustasta, silmälaseista tai piilolinseistä ja niin edelleen. (Spakov Oleg, 2008)

Katseenseurajalaitteet jaetaan yleensä kolmeen pääluokkaan (Spakov Oleg, 2008) niiden fyysisen kosketuksen määrän mukaisesti. Jako on seuraavanlainen.

- Valon heijastumiseen perustuvat tekniikat
 - Silmänliikkeiden kuvaaminen videokameralla (video oculography)
 - Silmän reuna-alueen seuraaminen (Limbus tracking)
 - Pupillin liikkeenseuranta (Pupil tracking)
 - Keinotekoinen neuroverkko (Artificial neural network)
 - Silmänliikkeiden havaitseminen infrapunavaloa heijastamalla (Infrared oculography)
- Sähköpotentiaalia silmää ympäröivältä iholta mittaavat tekniikat
- Kontaktilinssin seuraamiseen perustuvat tekniikat

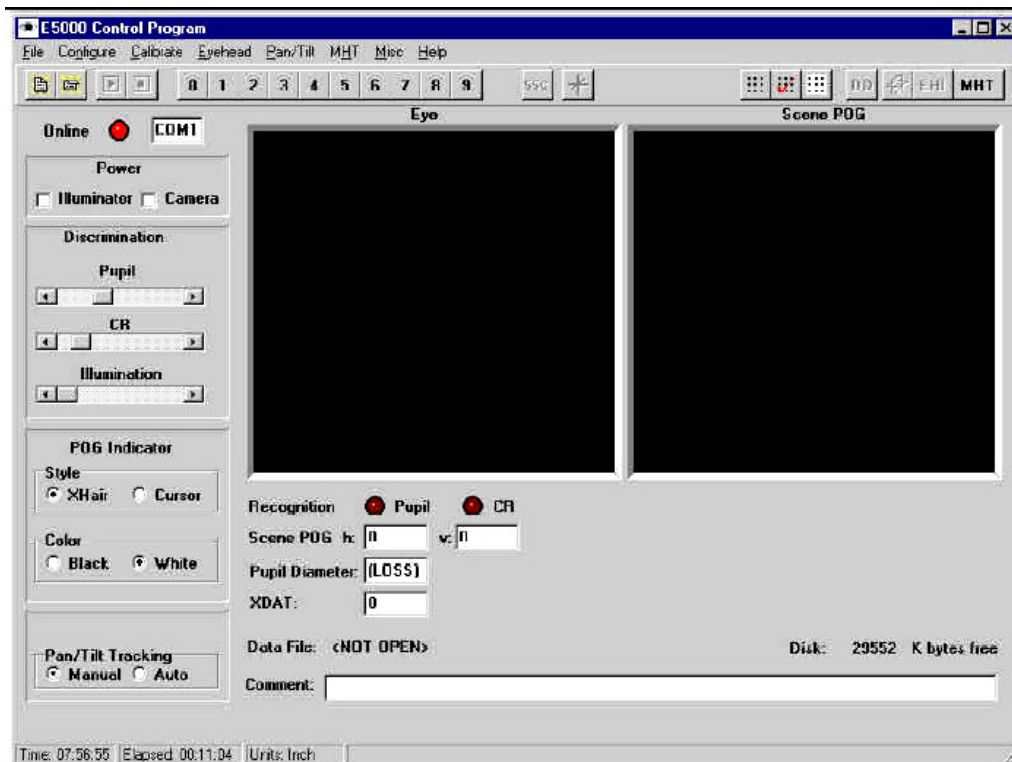
Ensimmäiseen luokkaan kuuluvat laitteet joilla seurataan valon – yleensä infrapunavalon – heijastumaa. Samainen luokka jakaantuu vielä pienempiin osiin, joissa on määritelty erilaisia tutkimustekniikoita. Toisen luokan muodostavat sähköpotentiaalia silmää ympäröivältä iholta mittaavat laitteet ja kolmannen luokan muodostavat silmään asetetun piilolinssin liikkeen seuraamiseen perustuvat laitteet. (Spakov Oleg, 2008) Tässä tutkielmassa keskitytään vain laitteisiin jotka seuraavat valon heijastumaa ihmisen silmästä.

4.5.2 Ohjelmistot

Silmänliikekameran avulla saatua raakadataa on pystyttävä käsittelemään ja analysoimaan jollakin tavalla. Tähän astuvat avuksi laitteiden mukana tulevat tai erikseen hankittavat ohjelmistot. Tässä tutkielmassa on käytetty Tobii - silmänliikekameran yhteydessä ClearView ohjelmaa ja ASL – silmänliikekameran yhteydessä Eyenal - ohjelmaa.

Molemmat ohjelmistot ovat kunkin silmänliikekameratyyppin tulostaman dataan erikoistuneita ja niiden avulla voidaan raakadatasta suodattaa haluttua tietoa tutkimuskäyttöön. EDTPlayer nimisen ohjelman avulla voidaan tarkastella videokuvaa ja laittaa sekä ASL:stä, että Tobii:sta saadut katsepolut näkymään yhtä aikaa videokuvan päällä.

Katseenseurantakontrolliyksiköllä on myös käytössään oma ohjelmisto. Kuvassa 7 nähdään millainen kyseinen ohjelmisto voi olla. Kuten kuvassa vasemmalla olevasta valikosta voidaan nähdä, pystytään ohjelman avulla säätämään pupillin (pupil) ja sarveiskalvon (CR) arvoja.



Kuva 7. E5000 Control Program (ASL, 2003)

Internetissä on myös saatavilla valmiita algoritmeja silmänliikkeiden eri ominaisuuksien laskemiseen, joita voidaan hyödyntää ohjelmistojen avulla. Yleensä silmänliikekameroiden ohjelmistojen mukana seuraa jonkinlainen kehitysympäristö (*development kit*), jonka avulla voidaan ohjelmoida uusia algoritmeja omaan tarpeeseen vanhojen tueksi. Ohjelmistoihin kuuluvat kehitysympäristöt tukeutuvat monesti suosittuihin ohjelmointikieliin, joten täysin uutta syntaksia ei tarvitse välttämättä alkaa opettelemaan, jos on aiemmin perehtynyt edes jonkin verran ohjelmointiin. Tässä tutkielmassa käytetyistä silmänliikekameroista saatu raakadata pystytään tuomaan Excel – taulukkolaskenta ohjelmaan, jossa dataa voidaan käsitellä. Raakadatiedostot sisältävät monesti suuren määrän tietoa, joten datan analysoinnissa voidaan avuksi ohjelmoida Excelissä Visual Basic – makroja.

4.5.3 Tutkimusympäristö

Kuten aikaisemmin tässä luvussa todettiin, on tutkimusympäristöllä suuri vaikutus testeissä saatuihin lopputuloksiin ja itse testien onnistumiseen. Tutkimusympäristön kuntoon asettaminen on aina omanlaisensa prosessi katseenseurannassa. Tärkeintä on ensin asentaa laitteisto paikoilleen, sen jälkeen hoitaa valaistus kuntoon ja lopuksi vielä säätää ja laittaa ohjelmisto päälle.

Katseenseurannan tutkimusympäristönä toimii tässä yhteydessä laboratorio, johon on asennettu tutkimuksia varten silmänliikekamerat, tietokoneet ja muu tarvittava laitteisto. Tämän lisäksi laboratoriossa on tehty tarvittavat laitteistokytkenät tutkimuskäyttöä varten. Tilan valaistusta pystytään säätämään tarvittaessa.

4.6 Katseenseurannan vaiheet

Seuraavaksi perehdytään katseenseurantaan käytännössä. Aluksi käydään läpi yleisimmät vaiheet, jotka on huomioitava ennen mittausten aloittamista. Sen jälkeen käydään läpi kohta kohdalta miten mittaustilanteessa kannattaa menetellä. Tutustutaan asioihin, jotka saattavat haitata mittaustilannetta ja vaikuttaa negatiivisessa mielessä saatuihin lopputuloksiin. Tämän luvun yhteydessä katseenseurannan päävaiheet on jaoteltu neljään osaan, käyttäen pitkälti pohjana Nevalaisen ja Sajaniemen (2004) mukaista jaottelua. Käydään läpi tärkeimmät katseenseurannan vaiheet.

4.6.1 Testihenkilön asettaminen paikoilleen

Aivan ensiksi, ennen kuin itse testi voidaan aloittaa, tulee testihenkilö ohjata paikalleen istumaan. Tällöin varmistetaan samalla että laitteet ja henkilö ovat oikeissa paikoissa suhteessa toisiinsa. Nyt tulee huomioida että silmänliikekamera on asetettu oikealle etäisyydelle ja että se on oikeassa kulmassa suhteessa testihenkilöön. Testihenkilön ja silmänliikekameran välillä tulee olla esteetön näkyvyys ja henkilön tulee istua paikoillaan kuitenkin niin, että hän pystyy istumaan vaivatta mutta liikkumatta paikoillaan. Ainakin koehenkilön pään tulisi pysyä paikoillaan sillä tässä yhteydessä suoritetuissa testeissä silmänliikekamerat ovat herkkiä pään liikkeille (tosin koehenkilö voi tehdä hienoisia päänliikkeitä ja korjaavia asennonmuutoksia).

Testihenkilön ja silmänliikekameran välinen etäisyys tulee olla optimaalinen (suositusetäisyys tarkistetaan tässä vaiheessa silmänliikekameran manuaalista). Lisäksi voidaan vielä tehdä hienosäätöä siirtämällä istuinta jossa testihenkilö istuu, lähemmäksi tai kauemmaksi silmänliikekamerasta. Tässä vaiheessa ohjelmisto on päällä ja sen avulla voidaan tarkastaa havaitseeko silmänliikekamera koehenkilön silmänliikkeet. Olisi erittäin suositeltavaa, että testihenkilön silmät näkyisivät mahdollisimman keskellä kameran näytössä.

4.6.2 Katseenseurantaohjelman säätäminen

Testitilanteessa toimii yleensä vähintään kaksi henkilö, joista toinen on varsinainen koehenkilö (jonka katsetta mitataan) ja toinen henkilö säätää laitteita, sekä huolehtii monitoroinnista. Seuraavaksi varmistetaan että katseenseurantaohjelmisto on valmiina käyttöä varten. Huomionarvoisia seikkoja ovat etenkin tarkistaa että silmänliikedatan tallennusta varten on avattu uusi tiedosto ja että ohjelman asetukset ovat kunnossa, sekä tunnistaako laitteisto varmasti silmänliikkeet. Jos käytössä on myös

katseenseurantakontrolliyksikkö, tulee varmistua että se on päällä ja että siihen on ladattu valmiiksi sen käyttämä ohjelmisto.

4.6.3 Kalibrointi

Kalibrointi suoritetaan aina ennen laitteiston varsinaista käyttöä. Näin varmistetaan että laitteisto toimii koetilanteessa riittävällä tarkkuudella. Ennen kalibrointia tarkistetaan löytääkö silmänliikekamera testihenkilön pupillin ja sarveiskalvon (tämä nähdään apumonitorista, johon testihenkilön silmä tarkennetaan). Henkilö joka vastaa monitoroinnin seurannasta voi tehdä vielä hienosäätöä ohjelman avulla ja muuttaa ohjelmiston avulla pupillin tai sarveiskalvon tarkkuusarvoja. Samalla ohjelmiston tulee antaa valideja arvoja (sellaisia arvoja jotka ovat ohjelman arvoasteikon sisällä) näytölle samalla kun testihenkilö katsoo eri kohtiin näyttöä. Näin varmistetaan että laitteisto rekisteröi varmasti testihenkilön silmänliikkeet eri kalibrointipisteissä. (Francesc Tintó Garcia-Moreno, 2001)

Varsinainen kalibrointi tapahtuu samoin eli näyttämällä käyttäjälle tietokoneen ruudulla olevaa kuvaa, jossa on tietty määrä staattisia pisteitä (eli kalibrointipisteitä). Pisteet sijaitsevat symmetrisesti eri puolilla näyttöä. Tällä taataan kalibroinnin tarkkuus myös ruudun eri äärialueilla. Käyttäjä katsoo pisteitä yksi kerrallaan ja samalla katseenseurantalaitteiston ohjelmisto analysoi silmän (tarkalleen sanoen pupillin keskikohdan ja sarveiskalvoheijasteen) asennon jokaisella kerralla. Katseen on osuttava tai käytävä hyvin lähellä pisteitä, että kalibroinnin voidaan sanoa onnistuneen. Ohjelman avulla kalibrointia voidaan kompensoida jonkin verran silloin, kun käyttäjä osuu suurimpaan osaan kalibrointipisteistä, mutta jos yksittäinen piste ei osu yhteen katseen kanssa.

Kalibroinnin onnistuminen määrittää myös pitkälle katseenseurannan onnistumisen, joten kalibrointi kannattaa tehdä huolella. Mitä lähemmäksi kalibrointipisteitä testihenkilön katse osuu, sitä tarkempia ovat lopussa saadut testitulokset. Jos

katseenseurantaan osallistuu useampi henkilö, jokaiselle suoritetaan kalibrointi erikseen. Joskus kalibrointi saatetaan toistaa tutkimuksen aikana. Syy tähän johtuu pitkälti teknologian rajoitteista, sillä esimerkiksi laitteiden tai koehenkilön asento voi muuttua ja tämä johtaa siihen että käytetyt algoritmit eivät enää toimi riittävän hyvin.

4.6.4 Monitorointi

Monitorointivaihe sisältää seurantalaitteen tilan tarkkailun ja mahdollisesti - jos tarvetta on - asetusten uudelleen asettamisen. Monitoroinnista huolehtii joku muu kuin testihenkilö. Monitoroinnin aikana tarkkaillaan tietokoneen ja silmänliikekameran näytölle tulostamaa syötettä ja tarkkaillaan ohjelmiston ruudulle tulostavia arvoja. Tärkeintä tässä vaiheessa on tarkistaa, että laitteen antamat arvot ovat valideja (vaihtelevat tietyn järkevän rajan sisällä ja että arvoja yleensä saadaan rekisteröityä laitteista). Jos jossain välissä käy niin, ettei valideja arvoja saada enää syntymään pitkällä aikavälillä (suhteessa testin kokonaiskestoon), joudutaan testi keskeyttämään ja aloittamaan alusta. Jos taas valideja arvoja ei saada tietyllä lyhyellä aikavälillä, tulee tämä huomioida kun myöhemmin tehdään datan analysointia (tämä nähdään suoraan raakadatasta).

4.7 Tutkimuksessa käytetyt silmänliikekamerat

Seuraavaksi tutustutaan tässä tutkielmassa käytettyihin silmänliikekameroihin. Tässä tutkielmassa tehdyissä tutkimuksissa on käytetty silmänliikekameroina Tobii- ja ASL-silmänliikekameroita. Malleina ovat olleet Tobii 1750 ja ASL 504.

ASL 504 on pöydälle asetettava, suhteellisen pienikokoinen ja helposti liikuteltava silmänliikekamera. Tobii 1750 on pöydälle asetettava, tavallista tietokoneen näyttöä

muistuttava silmänliikekamera, jossa itse kameraosa on upotettu näytön alapuoliseen osaan. Molemmissa kameroissa näytteenottotaajuutena on käytetty 20 millisekuntia.

4.7.1 ASL 504

ASL malli 504 on kuvan 8 mukainen pöydälle asetettava silmänliikekamera. Samasta kuvasta käy hyvin ilmi ASL 504:n rakenne. Kuten nähdään, on malli kooltaan suhteellisen pieni joten sitä voidaan liikutella vaivatta. Kamerassa on keskellä ympyrän muotoinen alue jossa sijaitsee infrapunaledi. Se aistii silmän liikettä heijastamalla infrapunavaloa koehenkilön silmään.

ASL 504:sta käytettäessä tulee testihenkilön silmän olla oikeassa kohtaa illuminaation aikaansaamiseksi. Kamera mahdollistaa 25 asteen kallistuskulman ja 100 asteen panorointikulman. Mallissa on alaosassa kallistuskulmansäätöruuvi, jonka avulla silmänliikekameran kuvakulmaa voidaan säätää. Kamerassa on myös automaattisesti fokusoituva linssi, sekä zoomaustoiminto laitteen minimi ja maksimitoimintosäteen sisällä. ASL 504:ssä on myös automaattinen silmän etsinnän toiminto. (ASL, 2003)



Kuva 8. ASL 504 (ASL, 2003)

ASL 504 käyttää apunaan katseenseurantakontrolliyksikköä (ASL Eye tracking System – model 5000 control unit). Se toimii omalla virtalähteellä ja on yhteydessä tietokoneeseen. Kontrolliyksikkö vastaa tiedon prosessoinnista johon kuuluu katseen

reitit laskeminen ja videon vastaanottaminen silmä- ja tapahtumakamerasta (scene camera). Kontrolliyksikkö on rajapinta tietokoneeseen, jonka avulla voidaan liikuttaa ASL-silmänliikekameraa pan/tilt – mekanismin avulla. ASL silmänliikekameran yhteydessä käytetään hyödyksi EyeNal – ohjelmistoa, jolla voidaan käsitellä ASL:stä saatua dataa. Toinen ASL- datan analysoinnissa käytettävä ohjelma on Eyenal – ohjelma. (ASL, 2003)

4.7.2 Tobii 1750

Tobii – 1750 silmänliikekamera, joka nähdään alla kuvassa 9, tarjoaa näyttöön upotetun kameran (kuvassa näytön alalaidassa). Laitteen yläosa toimii aivan kuin normaali tietokoneen näyttö.



Kuva 9. Tobii 1750 (Tobii Technology, 2003)

Laitemanuaalin mukaan Tobii 1750 pystyy 0,5 asteen tarkkuuteen (Tobii Technology AB, 2003) optimaalisessa mittausympäristössä ja asteen tarkkuuteen silloin kun otetaan huomioon erilaiset mittaustilanteisiin vaikuttavat muuttuvat tekijät, kuten päänliikkeestä johtuva kompensatio. Samoin Tobii 1750 osaa tutkia molempia silmiä yhtä aikaa ja päättää automaattisesti kumpi silmistä on oikea ja kumpi vasen. Tämä kaikki tapahtuu

välittämättä itse pään asennosta. Tällöin mahdollistuu pitempiaikainen kalibrointi, koska nyt monet huojunnasta johtuvat haitat pystytään symmetrisyyden ansiosta välttämään. Tobii 1750:n näkökenttänä on 20 x 15 x 20 cm (leveys, korkeus ja syvyys) oleva alue ja etäisyys näytöstä koehenkilöön tulee olla noin 50 senttimetriä.

Laitteistomanuaalin mukaan Tobii 1750 pystyy suoriutumaan erilaisissa valaistusolosuhteissa, poikkeuksena suora auringon paiste tai muu kirkas ulkopuolinen valo joka voisi sotkea laitteen oman valolähteen. Ohjelmistona toimii ClearView – ohjelma, jolla voidaan analysoida Tobii 1750:sta saatua videokuvaa.

5 KATSEENSEURANNAN MITTAAMINEN

Tässä luvussa perehdytään tutkielman sisältämään tutkimusaineistoon, joka saatiin laboratoriomittauksissa. Aivan luvun aluksi esitellään hieman taustoja miten mittaukset suoritettiin. Sen jälkeen tarkastellaan katseenseurannatatesteistä saatua dataa. Lopuksi käydään läpi datan analysointivaihe ja suoritetaan vielä vertailua Tobii- ja ASL – silmänliikekameroista saadun datan välillä. Luvun lopussa esitellään joitakin tutkimustuloksiin vaikuttavia tekijöitä, joilla on merkitystä kun analysoidaan dataa tulkitaan.

5.1 Johdatus katseenseurannan mittaukseen

Katseenseurannan avulla mitataan tietoa silmänliikkeistä tietyn, testihenkilöä varten määritellyn tehtävän aikana. Tällainen tehtävä voi olla esimerkiksi kuvan katsominen tai tekstin lukeminen. Kyseisen tehtävän aikana testihenkilön silmänliikkeitä tutkitaan laitteiston ja ohjelmiston avulla, jotka tuottavat mittauksissa dataa jota myöhemmin analysoidaan. Aivan aluksi saatua dataa kutsutaan raakadataksi. Se pitää sisällään kaiken mittauksissa saadun datan suoritettujen testien ajalta. Raakadata on otettu tietyllä näytteenottotaajuudella, joka tässä tutkielmassa on ollut 20 millisekuntia. Tämä tarkoittaa käytännössä sitä, että sekunnin nauhoitukseen mahtuu 50 näytettä ja minuutin nauhoitukseen 3000. Tässä on myös eräs syy miksi nauhoitusten kesto on rajattu vähän yli minuutin mittaiseksi.

Raakadata ei sinällään ole kovin havainnollista. Tämän takia sitä pyritään myöhemmin käsittelemään. Yleensä raakadatasta suodatetaan tietoa tietyltä aikaväliltä erilaisten

algoritmien avulla. Esimerkiksi fiksaatioiden suodattamiseen löytyy erilaisia algoritmeja, samoin kuin attention kohteiden määrittämiseen. Niistä puhutaan tässä tutkielmassa tarkemmin hieman myöhemmin.

Tutkimuksissa on pyritty simuloimaan samankaltaista tilannetta joka valitsee kun henkilö lukee tietokoneen näytöltä tekstiä, tosin pienin muutoksin. Siinä missä normaalisti luettaessa katse siirtyy vasemmalta oikealle, on tässä tutkielmassa tehdyissä testeissä katsottu kolmea näytön kesirivillä olevaa kalibrointipistettä vuorotellen. Katse liikkuu sekä vasemmalta oikealle, että oikealta vasemmalle. Attention kohde pyritään kiinnittämään aina keskelle kalibrointipisteitä.

5.2 Mitä mitattiin

Painopisteenä oli kahden eri silmänliikekameran välisen raakadatan vertailu. Tutkimuksissa haluttiin nähdä miten eri kameramallien välinen data poikkeaa toisistaan ja mitä yhteistä niillä on. Analysoinnin tuloksista haluttiin päätellä miten yhtenäistä raakadatat ovat keskenään samanlaisissa testinauhoitustapauksissa ja voidaanko niiden antamia tuloksia pitää keskenään luotettavina, sekä vertailukelpoisina.

Raakadatasta haluttiin selvittää ensin fiksaatioiden määrät, yksittäisten fiksaatioiden kestot, fiksaatioiden kokonaiskestot ja lisäksi fiksaatioiden minimi- ja maksimikestot. Tämän jälkeen tarkasteltiin lähemmin päällekkäisiä attentionkohteita, toisin sanoen katseen kohteiden päällekkäisyyksiä raakadatassa. Kun nämä oli saatu selvitetty niin ALS:stä (yksin- ja pariohjelmointiasetuksilla) kuin Tobii:sta, tutkittiin vielä miten lopputulokset vastaavat toisiaan.

5.3 Miten mitattiin

Mittauksissa käytettiin hyvin pitkälti tässä tutkielmassa esitettyjen vaiheiden (kohta 4.4) mukaista lähestymistapaa. Ensin koehenkilö ohjattiin istumaan pöydän ääreen. Pöydällä oli valmiiksi asetettu molemmat silmänliikekamerat ja tässä vaiheessa varmistettiin vielä, että näkyvyys molemmista kameroista testihenkilön silmiin oli esteetön. Tämän jälkeen koehenkilölle etsittiin sopiva asento jossa hän pystyisi katsomaan näytöllä olevia ja eri paikoissa sijaitsevia kalibrointipisteitä vaivatta. Nyt pyrittiin siihen, että testihenkilön ei tarvitsisi liikuttaa juurikaan päätään, vaan pelkästään katsetta. Samoin yritettiin muutenkin välttää koehenkilöllä kehon liikkeitä jotka olisivat haitanneet mittausta.

Kun koehenkilö oli saatu istumaan paikoilleen, aloitettiin asetusten säätäminen silmänliikekameran ohjelmiston avulla. Tobii-1750 saatiin vaivatta havaitsemaan koehenkilön silmänliikkeitä, koska kyseinen kameramalli toimii normaalin tietokoneen näytön tapaan. Tässä tapauksessa olikin tärkeää että koehenkilö istui oikealla etäisyydellä ja oikealla korkeudella kameran näytöstä ja että katseyhteys oli kameraan suora, siten ettei edessä ollut näköesteitä. Kuvassa 10 näkyy miten laitteet ovat sijoitettu suhteessa koehenkilöön.

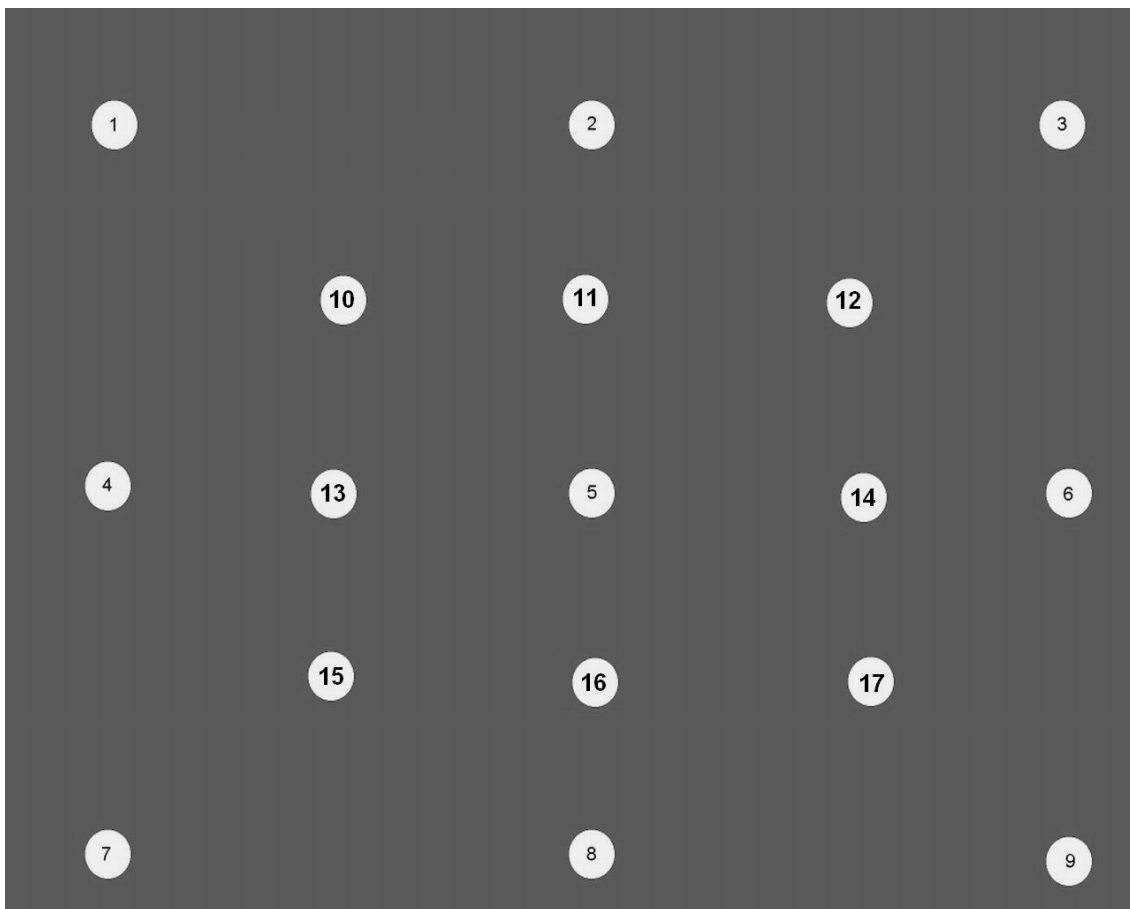


Kuva 10. Testitilanne

ASL 504 on kooltaan pienempi, joten se asetettiin Tobii 1750:en viereen alavasemmalle. Ensin ASL 504 laitettiin oikeaan asentoon siten, että siitä oli suora yhteys koehenkilön silmiin ja tämän jälkeen voitiin tehdä vielä tarkempaa hienosäätöä ohjelmiston avulla. Ohjelma päivittää reaaliajassa näytölle koehenkilön silmistä saatuja arvoja ja kameraa voidaan ohjata hienovaraisesti tietokoneella ohjelman avulla. Koehenkilön silmän tulisi näkyä apumonitorissa mahdollisimman keskellä, että saadut arvot olisivat tarkkoja.

Mittaaminen suoritettiin niin yksinohjelmoijan, kuin pariohjelmoijan näkökulmasta, koska haluttiin nähdä miten pariohjelmointia suoritettaessa silmänliikkeet eroavat yksinohjelmoinnista. Testinauhoitus aloitettiin yksinohjelmoijan asetuksilla. Tämä tapahtui siten, että testihenkilö sijoitettiin istumapaikalle josta hänen voitaisiin kuvitella tekevän työtä oikeassakin työympäristössä. Testihenkilön etäisyys (eli henkilön silmien etäisyys) näytön kalibrointipisteistä oli 65–70 senttimetrin sisällä riippuen kalibrointipisteen sijainnista näytöllä. Näillä asetuksilla tehtiin kolme nauhoitusta, joissa testihenkilö kävi yksi kerrallaan läpi näytöllä näkyviä kalibrointipisteitä. Ainoa oleellinen ero ”realistiseen” yksinohjelmointitilanteeseen oli se, että testihenkilö joutui pitämään päänsä asennon tarkemmin kohdallaan.

Itse testi pidettiin suhteellisen yksinkertaisena. Koehenkilön tehtävänä oli katsoa näytöllä olevan kuvan kalibrointipisteistä keskimmäisiä eli numeroja 13, 5 ja 14. Kuvasta 11 nähdään miten kalibrointipisteet ovat jakautuneet näytölle. Ne ovat jakautuneet näytölle symmetrisesti ja kattavat koko näytön alueen. Pisteiden välissä on vapaata tilaa sen verran, ettei pisteiden ulkopuolelle osuvaa katsetta sekoiteta kahden eri pisteen välillä.



Kuva 11. Näytön kalibrointipisteet

Seuraavaksi tehtiin kolme nauhoitusta pariohjelmoijan näkökulmasta. Nyt kameroiden paikkoja muutettiin suhteessa testihenkilöön seuraavalla tavalla. Testihenkilön katseen etäisyys Tobiiin näytöstä asetettiin 56 senttimetriin ja ASL:n etäisyys Tobii:sta vasemmalle 11 senttimetrin päähän ja alas 6 senttimetrin. Testihenkilö istui näin ollen sivummassa verrattuna yksinohjelmointitilanteen asetelmaan. Kuvasta 12 käy ilmi paremmin pariohjelmointia varten laadittu asetelma.



Kuva 12. Kameroiden ja testihenkilöiden sijainti simuloitaessa pariohjelmointia

Kun koehenkilö oli valmis ja alkuasetukset oli saatu kohdilleen, aloitettiin itse mittaaminen. Molempien silmänliikekameroiden antamaa dataa tallennettiin kovalevylle tiedostoon. Ensimmäisellä nauhoituskerralla koehenkilö katsoi näytöllä olevia kalibrintipisteitä ja toinen henkilö toimi apuna monitoroinnissa ja nauhoitusten päälle laitossa. Laboratoriossa tehtiin yhteensä kuusi lyhytkestoista nauhoitusta, joissa tarkoituksena oli saada vertailukelpoista dataa kahden eri silmänliikekameratyyppin välillä. Kun nauhoitukset lopetettiin, tallennettiin testissä saatu data tiedostoihin ja omiin alihakemistoihin. Tiedostot nimettiin päivän ja kellonajan mukaisesti siten, että niiden käsittely myöhemmin olisi selkeää. ASL:n antamasta datasta tuotiin Excel – taulukkolaskentaohjelmaan raakadata ja fiksaatiodata.

5.4 Datan analysointi

Seuraava vaihe on nauhoituksissa saadun datan analysoiminen. Kerrataan vielä mitä analysoinnin avulla halutaan tässä yhteydessä selvittää. Nyt käytössä on laaja määrä raakadataa, joka on saatu aiemmin esitellyistä nauhoitetuista testeistä Tobii 1750 – ja ASL 504 - silmänliikekameronilla. Raakadastasta halutaan nyt suodattaa tutkimuksen kannalta tarvittavaa tietoa. Haluamme tässä yhteydessä selvittää yksityiskohtaisesti raakadatan sisältämät fiksaatiot, suodattamalla ne ensin fiksaatioalgoritmin avulla. Toinen tarkastelun aihe on päällekkäiset attention kohteet, jotka nähdään suoraan raakadastasta tarkastelemalla attentionkohteiden X- ja Y-koordinaatteja, samalla ajanhetkellä ja tietyn marginaalin sisällä.

Kokeissa saatua dataa analysoitaessa on huomioitava, että kyseinen prosessi on katseenseurannassa usein varsin aikaa vievää. Toinen analysointiin vaikuttava seikka on käytössä oleva ohjelmisto ja sen rajallisuus. Ohjelmistoilla voidaan mitata vain tiettyjä arvoja, jotka ovat yleensä katsepolun kulku, fiksaatiot ja niihin liittyvät ominaisuudet ja jotain tietoa sakkadeista. Toisaalta kokeissa ei saada selville laitteiston ja ohjelmiston kautta esimerkiksi sitä tietoa, miksi testihenkilö toimii testitilanteessa tietyllä tavalla tai mitkä asiat vaikuttavat testihenkilön toimintaan.

Laitteiston ja ohjelmiston muodostamaa dataa kutsutaan aluksi raakadataksi. Raakadata sisältää kaiken sen informaation joka on saatu mittauksissa silmänliikekamerasta. Raakadata on joko ascii- tai binäärimuotoista, riippuen siitä mitä ohjelmistoa käytetään. Nyt suoritetuissa mittauksissa saatu raakadata tallennettiin testin jälkeen edellä esitellysti tiedostoihin ja alihakemistoihin. Koska raakadata sisältää kaiken testissä saadun datan, on se tästä syystä määrällisesti laajaa. Sen takia raakadastasta suodatetaan yleensä tietyillä algoritmeilla haluttua tietoa, jolloin datasta tulee havainnollisempaa analysointia varten.

Raakadatan analysointi aloitetaan erottamalla silmänräpäykset ja sakkadien aikana talletettu aineisto fiksaatioista. Itse katseenseurantadatasta voidaan määrittää fiksaatiot, sakkadit, ja katsepolut. Niiden käyttäytymistä voidaan sen jälkeen tutkia. Saatua dataa voidaan helpoiten analysoida kunkin laitteen dataan erikoistuneella ohjelmistolla. Ohjelmiston avulla dataa voidaan suodattaa tietyltä aikaväliltä, erottaa sieltä haluttu tieto ja tehdä lopuksi saatujen arvojen perusteella johtopäätöksiä.

5.4.1 Alkuasetukset yksinohjelmointia simuloitaessa

Ensin katsotaan lähemmin yksinohjelmointitavalla saatuja tuloksia. Seuraavassa tarkastelussa käydään läpi ensin tutkimustilanteeseen liittyvät yksityiskohdat. Etäisyydet muutetaan tässä vaiheessa senttimetreistä tuumiksi koska jatkossa käytettävissä sovellusohjelmissa käytetään pitkälti tuuma-arvoja.

Testitilanteessa henkilö istuu verrattain keskellä suhteessa tietokoneen näyttöä. Hän pääsee vapaasti sopivalle etäisyydelle, joka on tässä yhteydessä mitattuna noin 67,5 senttimetriä. Näytön koko, jota testihenkilö katsoo, on 17 tuumaa. Joudumme laskemaan seuraavaksi näkökentän laajuuden, johon vaikuttavat testihenkilön silmien etäisyys näytöstä ja näytöllä olevien kalibrointipisteiden etäisyys (käytetään pisteitä 1 ja 9).

Näkökentän laajuus saadaan laskemalla x kaavasta:

$$d = s * \tan x$$

Kaava 1. Näkökentän laajuus

Missä d = kahden kalibrointipisteen (1 ja 9) välinen etäisyys näytöllä, s = testihenkilön katseen etäisyys näytöstä ja x = näkökentän laajuus.

Seuraavaksi joudutaan vielä laskemaan sekä pysty-, että leveysuunnassa olevien eyetracker yksiköiden määrä per aste. Siihen käytetään apuna alla olevia kaavoja.

$$\theta_h = \tan^{-1}(dh/s)$$

Kaava 2. Leveysuunnassa olevien eyetracker yksiköiden määrä per aste.

$$\theta_v = \tan^{-1}(dv/s)$$

Kaava 3. Korkeussuunnassa olevien eyetracker yksiköiden määrä per aste.

Missä s = katseen etäisyys näytön keskikohdasta, dh = leveysuunnassa kahden kalibrintipisteen etäisyys ja dv = pystysuunnassa kahden kalibrintipisteen etäisyys. Tuloksena molemmista saadaan kulma asteina jossa testihenkilö katsoo näytön pisteitä.

Sen jälkeen joudutaan vielä jakamaan tällä välillä olevien eyetracker-pisteiden määrä kulman suuruudella, jolloin saadaan eyetracker-unittien määrä / aste niin pysty- kuin vaakasuunnassakin.

Asetukset simuloitaessa pariohjelmointia	
Katseen etäisyys näytöstä	67,5 cm = n. 27"
Näytön koko	17"
Näkökentän laajuus	34,95 astetta
Katseenseurantakoordinaatit kalibrintipisteelle 1	X = 31 Y = 36
Katseenseurantakoordinaatit kalibrintipisteelle 9	X = 229 Y = 227
Etäisyys vaakasuunnassa kalibrintipisteestä 1 pisteeseen 9	26,3 cm = 10,35"
Etäisyys pystysuunnassa kalibrintipisteestä 1 pisteeseen 9	20 cm = 7,9"
Leveysuunnassa olevien eyetracker - yksiköiden määrä (yksikkö / aste)	9,7 yksikköä/aste
Pystysuunnassa olevien eyetracker - yksiköiden määrä (yksikkö / aste)	13,1 yksikköä/aste

Taulukko 2. Yksinohjelmointia simuloivan testitilanteen asetukset

Kun ASL:stä saatua raakadataa analysoidaan, avataan ensiksi Eyenal – ohjelmistolla aiemmin syntynyt tiedosto. Tiedosto on tässä tapauksessa eyd – päätteinen. Kun tiedosto aukaistaan Eyenal – ohjelmalla, saadaan näkyviin kuvan 13 mukainen ikkuna, josta nähdään jo tiettyjä arvoja. Iso ikkuna on jaettuna kahteen osaan, jossa vasemmalla näkyvät tiedostot jotka ovat auki ja oikealla puolella raakadataa sisältävät sarakkeet. Sarakkeista voidaan nähdä mm.

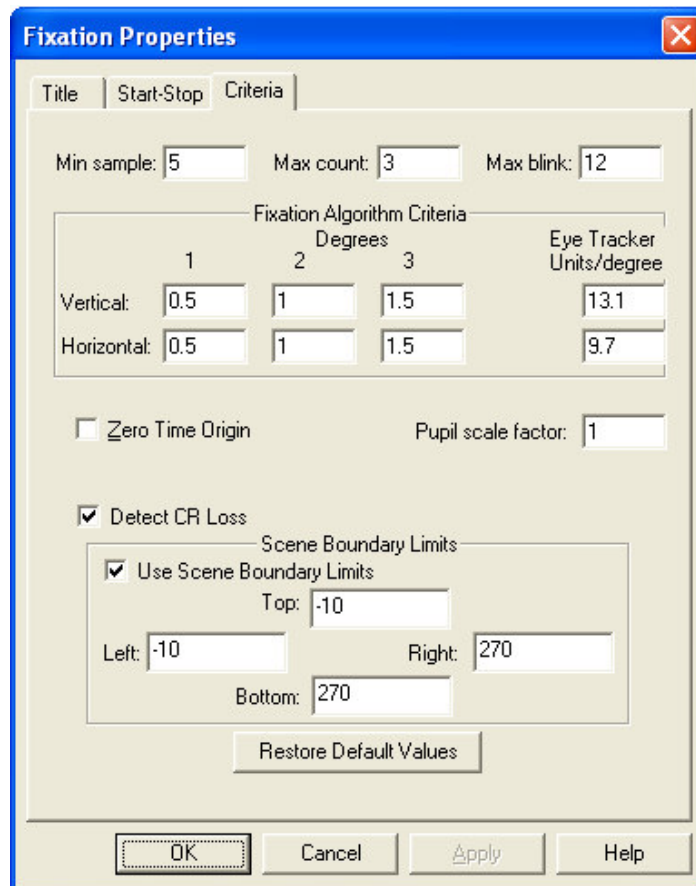
Field No	Seg No	Time	Total secs	H-pos	V-pos	Diam	XDAT	Mark	FC
0	1	11:30:28.640	41428.640	165	224	51	0		44829
1	1	11:30:28.660	41428.660	165	209	51	0		44830
2	1	11:30:28.680	41428.680	164	192	51	0		44831
3	1	11:30:28.700	41428.700	170	181	51	0		44832
4	1	11:30:28.720	41428.720	173	172	51	0		44833
5	1	11:30:28.740	41428.740	177	163	51	0		44834
6	1	11:30:28.760	41428.760	181	155	51	0		44835
7	1	11:30:28.780	41428.780	180	146	51	0		44836
8	1	11:30:28.800	41428.800	180	141	50	0		44837
9	1	11:30:28.820	41428.820	182	139	49	0		44838
10	1	11:30:28.840	41428.840	184	135	48	0		44839
11	1	11:30:28.860	41428.860	185	135	47	0		44840
12	1	11:30:28.880	41428.880	188	133	46	0		44841
13	1	11:30:28.900	41428.900	189	131	45	0		44842
14	1	11:30:28.920	41428.920	190	130	43	0		44843
15	1	11:30:28.940	41428.940	191	129	43	0		44844
16	1	11:30:28.960	41428.960	189	129	42	0		44845
17	1	11:30:28.980	41428.980	190	128	41	0		44846
18	1	11:30:29.000	41429.000	188	129	40	0		44847
19	1	11:30:29.020	41429.020	188	129	40	0		44848
20	1	11:30:29.040	41429.040	189	132	39	0		44849
21	1	11:30:29.060	41429.060	189	132	0	0		44850
22	1	11:30:29.080	41429.080	182	142	37	0		44851
23	1	11:30:29.100	41429.100	190	173	32	0		44852
24	1	11:30:29.120	41429.120	190	173	0	0		44853
25	1	11:30:29.140	41429.140	190	173	0	0		44854
26	1	11:30:29.160	41429.160	190	173	0	0		44855
27	1	11:30:29.180	41429.180	190	173	0	0		44856
28	1	11:30:29.200	41429.200	190	173	0	0		44857
29	1	11:30:29.220	41429.220	190	173	0	0		44858
30	1	11:30:29.240	41429.240	190	173	0	0		44859
31	1	11:30:29.260	41429.260	190	173	0	0		44860
32	1	11:30:29.280	41429.280	190	173	0	0		44861
33	1	11:30:29.300	41429.300	190	173	0	0		44862
34	1	11:30:29.320	41429.320	190	173	0	0		44863
35	1	11:30:29.340	41429.340	190	173	0	0		44864
36	1	11:30:29.360	41429.360	0	0	0	0		44865
37	1	11:30:29.380	41429.380	0	0	0	0		44866
38	1	11:30:29.400	41429.400	0	0	0	0		44867
39	1	11:30:29.420	41429.420	0	0	0	0		44868
40	1	11:30:29.440	41429.440	0	0	0	0		44869

Kuva 13. Näkymä raakadatatista Eyenal-ohjelmalla aukaistuna.

ASL – silmänliikekameran yhteydessä tulevalla Eyenal – ohjelmalla voidaan säätää raja-arvoja, joiden avulla raakadatatista voidaan suodattaa haluttua tietoa. Ohjelmasta

löytyy oma kohta fiksaatioasetuksille, joiden avulla käyttäjä voi asettaa parametrejä algoritmile joka laskee fiksaatioita.

Tässä tutkielmassa yksinohjelmointia simuloitaessa Eyenal – ohjelman avulla muutettiin fiksaatioasetuksia seuraavalla tavalla (kts. Kuva 14). Fiksaatioasetusten (*Fixation properties*) välilehdestä kriteerit (*criteria*) asetettiin vähimmäisnäytteenottoväliksi (*min. sample*) arvoksi 5, joka vastaa 50 Hz ja maksiminäytteenottoväliksi (*max. count*) arvoksi asetettiin 3 ja silmän räpäytyksen maksimikestoksi (*max blink*) arvoksi 12. Sen jälkeen laitettiin edellä lasketut eye-tracker units / degree arvot niin pysty- kuin vaakasuunnassa. Tämän lisäksi laitettiin rasti kohtaan use scene boundary limits.



Kuva 14. Eyenal-ohjelmassa käytetyt fiksaatioasetukset (*Fixation properties*).

Kun arvot on asetettu, painetaan OK – painiketta ja ohjelma muodostaa raakadatan annettujen parametrien pohjalta fiksaatiodatan. Fiksaatiodatan tiedostopäätte on .fix ja se

sisältää kaikki ne fiksaatiot, jotka on suodatettu edellä olevasta raakadatatista annettujen parametrien avulla. Sarakkeiden alapuolella olevassa harmaassa palkissa nähdään joitakin yksityiskohtia liittyen saatuun fiksaatiotietoon. Näitä tietoja ovat mm. fiksaatioiden määrä, keskimääräinen fiksaation kesto ja niin edelleen.

subject_id=
start_flag=
stop_flag=
min_fix_sample= 5

Seg_No	Fix_No	Ph_No	Strt_Time	Fix_Dur	InterFix_Dur	InterFix_Deg	Hor_Pos	Ver_Pos	Pupil_Diam	Eye/Scn_Dist	No_of_flags	Fix_Loss	InterFix_Loss	Fix
1	1	0	11:30:28.780	0.300	0.000	0.000	186.333	133.867	44.067	0.000	1	0.020	0.000	
1	2	0	11:30:30.160	0.480	1.080	0.440	189.520	130.040	40.040	0.000	0	0.000	1.020	
1	3	0	11:30:30.660	1.920	0.020	0.359	186.608	127.464	42.536	0.000	0	0.000	0.000	
1	4	0	11:30:32.680	0.120	0.100	10.991	80.000	127.857	44.714	0.000	0	0.000	0.000	
1	5	0	11:30:32.860	2.860	0.060	5.055	129.021	129.528	45.750	0.000	0	0.000	0.000	
1	6	0	11:30:35.800	2.000	0.080	5.233	78.257	129.653	46.089	0.000	0	0.000	0.000	
1	7	0	11:30:40.400	0.120	2.600	1.175	73.857	143.857	48.857	0.000	0	0.000	2.520	
1	8	0	11:30:40.540	0.200	0.020	1.384	74.091	125.727	45.000	0.000	0	0.000	0.000	
1	9	0	11:30:41.880	0.080	1.140	5.314	112.600	172.000	36.200	0.000	0	0.000	0.920	
1	10	0	11:30:41.980	0.120	0.020	2.974	94.714	141.429	39.000	0.000	0	0.000	0.000	
1	11	0	11:30:42.120	0.440	0.020	2.097	78.652	124.565	42.087	0.000	0	0.000	0.000	
1	12	0	11:30:42.620	2.460	0.060	5.397	130.944	128.065	45.008	0.000	0	0.000	0.000	
1	13	0	11:30:45.140	2.420	0.060	5.773	186.852	132.336	44.164	0.000	0	0.000	0.000	
1	14	0	11:30:47.660	2.200	0.100	11.269	77.550	130.306	45.297	0.000	0	0.000	0.000	
1	15	0	11:30:49.920	1.980	0.060	5.038	126.380	127.840	47.060	0.000	0	0.000	0.000	
1	16	0	11:30:51.980	1.740	0.080	6.232	186.625	134.557	45.955	0.000	0	0.000	0.000	
1	17	0	11:30:53.820	1.740	0.100	11.228	77.761	130.170	45.398	0.000	0	0.000	0.000	
1	18	0	11:30:55.640	2.100	0.080	5.205	128.245	131.160	45.547	0.000	0	0.000	0.000	
1	19	0	11:30:57.820	1.860	0.080	5.865	185.138	131.479	46.457	0.000	0	0.000	0.000	
1	20	0	11:30:59.760	1.820	0.080	10.993	78.511	131.043	45.152	0.000	0	0.000	0.000	
1	21	0	11:31:01.660	2.180	0.080	5.046	127.400	134.273	45.691	0.000	0	0.000	0.000	
1	22	0	11:31:03.920	1.880	0.080	5.945	185.053	132.779	46.842	0.000	0	0.000	0.000	
1	23	0	11:31:05.880	1.720	0.080	10.914	79.195	131.368	45.506	0.000	0	0.000	0.000	
1	24	0	11:31:07.660	2.400	0.060	4.992	127.504	135.851	47.512	0.000	0	0.000	0.000	
1	25	0	11:31:10.140	1.600	0.080	5.968	185.383	134.630	47.407	0.000	0	0.000	0.000	
1	26	0	11:31:11.820	1.560	0.080	10.723	81.405	130.747	45.430	0.000	0	0.000	0.000	
1	27	0	11:31:13.460	2.060	0.080	4.976	129.587	134.683	45.125	0.000	0	0.000	0.000	
1	28	0	11:31:15.580	1.460	0.060	5.951	187.284	132.243	46.135	0.000	0	0.000	0.000	
1	29	0	11:31:17.120	1.540	0.080	10.985	80.731	130.923	43.449	0.000	0	0.000	0.000	
1	30	0	11:31:18.740	2.120	0.080	4.860	127.850	132.888	47.654	0.000	0	0.000	0.000	
1	31	0	11:31:20.940	1.480	0.060	6.027	186.307	132.093	46.040	0.000	0	0.000	0.000	
1	32	0	11:31:22.500	0.240	0.080	10.421	85.231	130.615	47.692	0.000	0	0.000	0.000	
1	33	0	11:31:22.760	1.560	0.020	0.751	77.949	130.308	45.359	0.000	0	0.020	0.000	
1	34	0	11:31:24.400	2.660	0.080	5.170	128.075	132.179	46.776	0.000	0	0.000	0.000	
1	35	0	11:31:27.120	1.400	0.060	5.711	183.465	133.127	47.789	0.000	0	0.000	0.000	
1	36	0	11:31:28.600	1.360	0.080	10.111	85.391	131.594	42.638	0.000	0	0.000	0.000	
1	37	0	11:31:30.020	2.320	0.060	4.288	126.983	132.282	47.444	0.000	0	0.000	0.000	
1	38	0	11:31:32.400	1.420	0.060	5.874	183.958	132.778	47.569	0.000	0	0.000	0.000	
1	39	0	11:31:33.900	0.540	0.080	10.358	83.536	128.464	46.357	0.000	0	0.000	0.000	
1	40	0	11:31:34.460	0.920	0.020	0.533	78.787	131.234	45.532	0.000	0	0.000	0.000	

segment_start_time= 11:30:28.640
segment_end_time= 11:31:37.400
segment_duration_time= 00:01:08.760
number_of_fixations= 41

Kuva 15. Raakadatatista suodatettu fiksaatiotietoa Eyenal-ohjelmassa

Fiksaatiotiedostot tallennetaan oletuksen mukaisesti samaan kansioon raakadatatiedostojen (.eyd) kanssa. Tässä vaiheessa fiksaatiotietoa ja raakadatatia voidaan viedä Excel – taulukkolaskentaohjelmaan painamalla hiiren oikealla painikkeella eyd- tai fix-pääteisten tiedoston nimeä Eyenal-ohjelmassa ja valitsemalla kohta ”Export to EXCEL”. Näin voidaan siirtää tekstimuotoinen data Excel – tiedostoon, josta sitä on myöhemmin helppo käsitellä.

Taulukosta 3 nähdään ASL-nauhoituksessa saadut fiksaatiotiedot yksinohjelmointaessa.

ASL – nauhoitus 3 (yksinohjelmointi)	
Nauhoituksen kesto	1 min. 8 sekuntia
Fiksaatioiden määrä	41 kpl
Pisin fiksaation kesto	2,859375 s
Keskimääräinen fiksaation kesto	1.496098 s
Fiksaatioiden kokonaiskesto	61,01171875 s

Taulukko 3. ASL-nauhoituksen tiedot (yksinohjelmointi)

Taulukossa 4 nähdään Tobiiilla nahoitetusta datasta saadut fiksaatiotiedot.

Tobii nauhoitus	
Nauhoituksen kesto	1 min. 8 sekuntia
Fiksaatioiden määrä	88 kpl
Pisin fiksaation kesto	1,774 s
Keskimääräinen fiksaation kesto	0,48 s
Fiksaatioiden kokonaiskesto	42,525 s

Taulukko 4. Tobii-nauhoituksen tiedot

Nauhoituksen kesto oli molemmissa tapauksissa 68 sekuntia. Fiksaatioiden määrä jäi Tobiiilla puoleen verrattuna ASL:stä saatuun dataan. Keskimääräinen fiksaation kesto oli ASL:stä saadussa datasta pitempi, samoin myös fiksaatioiden kokonaiskesto.

5.4.2 Alkuasetukset pariohjelmointia simuloitaessa

Pariohjelmointia simuloivan testitilanteen erot yksinohjelmointiin verrattuna liittyvät pitkälti testihenkilön sijoittumiseen suhteessa silmänliikekameraan ja näyttöön. Nyt testihenkilö asetettiin istumaan niin, että hänen vierelleen olisi periaatteessa mahtunut toinenkin henkilö. Testihenkilö joutuu nyt istumaan hieman lähempänä näyttöä, koska ei voi tilan puutteen vuoksi perääntyä kauemmas.

Muuten toimitaan täysin samalla tavalla kuin yksinohjelmointia simuloitaessa. Ensin lasketaan näkökentän laajuus, sekä eyetracker-unittien määrä astetta kohti niin pysty- kuin vaakasuunnassa. Lopuksi Eyenal-ohjelman avulla luodaan fiksaatiodata. Taulukosta 5 nähdään asetukset Eyenal-ohjelmassa simuloitaessa pariohjelmointia.

Asetukset simuloitaessa pariohjelmointia	
Katseen etäisyys näytöstä	56 cm = n. 22,5”
Näytön koko	17”
Näkökentän laajuus	42,25 astetta
Katseenseurantakoordinaatit kalibrointipisteelle 1	X = 31 Y = 36
Katseenseurantakoordinaatit kalibrointipisteelle 9	X = 229 Y = 227
Etäisyys vaakasuunnassa kalibrointipisteestä 1 pisteeseen 9	26,3 cm = 10,35”
Etäisyys pystysuunnassa kalibrointipisteestä 1 pisteeseen 9	20 cm = 7,9”
Leveysuunnassa olevien eyetracker-yksiköiden määrää (yksikkö / aste)	8,1 yksikkö / aste
Pystysuunnassa olevien eyetracker-yksiköiden määrää (yksikkö / aste)	10,8 yksikköä / aste

Taulukko 5. Pariohjelmointia simuloivan testitilanteen asetukset

Taulukossa 5 nähdään kuinka etäisyys katsojan ja näytön välillä on 56 senttimetriä. Kalibrointipisteiden sijaintikoordinaatit ja niiden väliset etäisyydet ovat myös merkitty taulukkoon, samoin kuin eyetracker-yksiköiden määrä niin pysty- kuin leveys suunnassa.

ASL – nauhoitus 4 (pariohjelmointi)	
Nauhoituksen kesto	1 min. 8 sek.
Fiksaatioiden määrä	52 kpl
Keskimääräinen fiksaation kesto	1,211538 s
Pisin fiksaation kesto	2,640625 s
Fiksaatioiden kokonaiskesto	61,73438 s

Taulukko 6. ASL-nauhoituksen tiedot

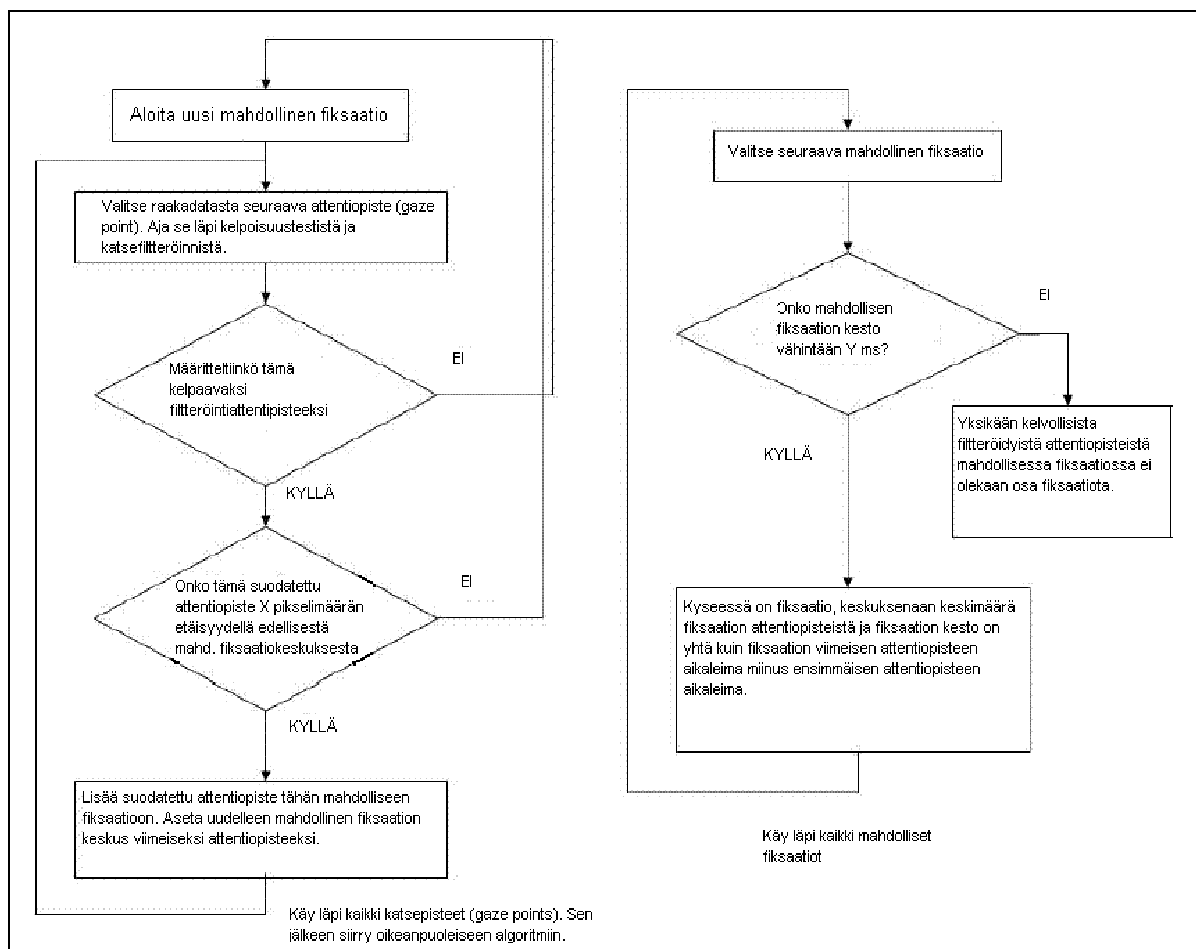
Taulukosta 6 voidaan nähdä ASL – nauhoituksen tiedot pariohjelmointiasetuksilla. Fiksaatioiden määräksi saatiin ASL:llä 52 kappaletta, keskimääräiseksi fiksaation kestoksi n. 1,21 sekuntia.

5.4.3 ClearView ja Eyenal - fiksaatioalgoritmit

Seuraavaksi esitellään Tobiiin ClearView – ohjelmiston ja ASL:n Eyenal – ohjelmiston käyttämät fiksaatioalgoritmit. Fiksaatioalgoritmit löytyvät yleensä ohjelmistojen omista manuaaleista.

ClearView

Kuvassa 16 on esitettynä Tobiiin käyttämä ClearView – ohjelmiston fiksaatioalgoritmi vuokaaviona.



Kuva 16. ClearView – ohjelman fiksaatioalgoritmi vuokaaviona. (Tobii Technology AB., 2006)

Kuvassa 16 on esitetty vuokaaviona ClearView-ohjelman käyttämä fiksaatioalgoritmi. Fiksaatioalgoritmi koostuu oikeastaan kahdesta eri algoritmista, joista toisen avulla suodatetaan attentiopisteitä ja toisella tarkastetaan onko kyseessä oikea fiksaatio (eli onko fiksaation kesto vähintään minimifiksaatiokeston mittainen). Ensin käydään läpi kuvan 16 vasemmanpuoleista algoritmia. Algoritmin avulla suodatetaan mahdollisen fiksaation attentiopisteitä. Ensin valitaan raakadatasta seuraava attentiopiste ja ajetaan se läpi kelpoisuustestistä, sekä katsefilteröinnistä. Mikäli attentiopiste menee läpi kummastakin tarkistuksesta, lisätään suodatettu attentiopiste tähän mahdolliseen fiksaatioon. Sen jälkeen asetetaan mahdollinen fiksaation keskus viimeiseksi attentiopisteeksi. Kaikki katsepisteet käydään läpi edellä mainitulla algoritmilla (kuvassa 16. vasen puoli). Sen jälkeen siirrytään kuvan 16 oikeanpuoleiseen algoritmiin jonka avulla suodatetaan mahdollisia fiksaatioita keston mukaan. Mikäli mahdollisen

fiksaation kesto on vähintään Y millisekuntia on kyseessä aito fiksaatio. Jo ei, niin yksikään suodatetuista attentipisteistä ei olekaan osa fiksaatiota.

EyeNal

EyeNal käyttää hyväkseen kolmen raja-arvon suodatusta (ASL, 2001) määritellesään fiksaatioita. Kolme suositeltavaa arvoa filteröintiin ovat 0,5 ; 1,0 ja 1,5 astetta. Ensimmäisen tutkitaan minimi näytteenottoväli (*min. sample*) arvoksi määritelty näyteväli, joka yleensä vastaa noin 100 millisekuntia. Jos keskimääräinen vaihteluväli X- ja Y-tasossa on vähemmän kuin ensimmäinen filteröintiväli (0,5 astetta), käytetään näitä X- ja Y-arvoja väliaikaisina fiksaatiokoordinaatteina (merkitään XT ja YT). Jos keskimääräinen vaihteluväli oli suurempi kuin kohdan yksi kriteeri, siirrytään eteenpäin yksi väli ja laskemista toistetaan kunnes löydetään kohdan 1 kriteerit täyttävä kohta.

Kun fiksaation alkukohta on määritelty, laskee ohjelma korkeus- ja leveysuuntaisen etäisyyden (DX, DY) seuraavaan näytepalaan väliaikaisesta sijainnista (XT, XY). Jos (DX, DY) on suurempi kuin kriteeri 2 (1,0 astetta), näyte sisällytetään fiksaatioon. Jos se ei mene läpi kakkoskriteeristä, sitten seuraavaa näytettä testataan kakkoskriteeriä vastaan.

Tämä jatkuu niin kauan kunnes mittausnäyte läpäisee kakkoskriteerin tai kunnes maksimi arvon (*max count*) mukaiset jaksolliset näytteet on testattu. Jos yksikin näistä arvoista jää kakkoskriteeriin, niin edelliset näytteet jotka eivät tehneet näin, testataan kolmoskriteeriä vastaan. Kaikki ne näytteet jotka läpäisivät kriteerit 2 tai 3 sisällytetään fiksaatioon. Tämä tarkoittaa sitä että niitä käytetään lopullisen fiksaation sijainnin määrityksessä.

Jos DX ja/tai DY maksimi arvosta (max count) jaksolliset näytteet ylittävät kakkoskriteerin niin sitten X ja Y arvot lasketaan mukaan. Jos arvojen sisältö ei eroa XT- ja XY-arvoista enempää kuin kakkoskriteerin arvon verran, sisällytetään ne kaikki

fiksaatioon. Muutoin fiksaatio suljetaan viimeisimmällä ehdot hyväksyttävällä datanäytteellä.

Silmän räpytykset määritellään maksimi kiinnioloaika (*max blink*) – arvon mukaiseksi pupilliarvon häviöksi tai niitä ei oteta huomioon. Ne eivät myöskään katkaise fiksaatiota. Huomion arvoista on todeta että maksimi kiinnioloaika eli silmänräpäyksen maksimikesto on syytä asettaa noin 200 millisekuntiin. Pupilliarvon katko, joka on kestoltaan max blink arvoa suurempi, johtaa fiksaation katkeamiseen viimeisellä hyväksytyllä näytedatalla.

5.5 Datan vertailu Tobii:n ja ASL:n välillä

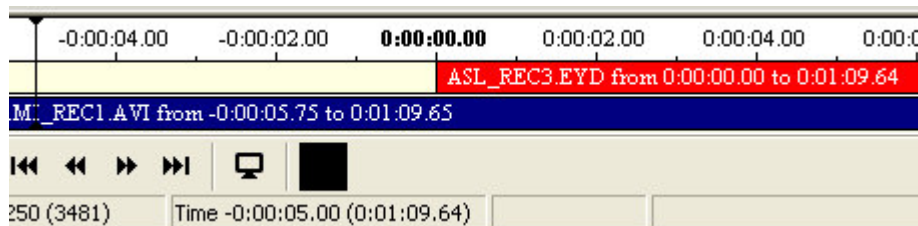
Tässä tutkielmassa saatujen mittaustulosten on määrä toimia pohjatietona suoritettaessa data-analyysiä kahden eri silmänliikekameran välillä. Mittauksissa käytetyistä silmänliikekameroista ASL 504:sta ja Tobii 1750:sta saatu raakadata ei ole suoraan vertailukelpoista. Tämän takia on aivan ensiksi suhteutettava saadut arvot toisiinsa niin, että niistä tulee vertailukelpoisia. Toinen seikka mikä tulee ottaa huomioon, että kumpikin silmänliikekameramalli käyttää eri sovellusohjelmia datojensa lukuun ja muodostaa hieman erityyppistä raakadataa käsiteltäväksi.

Seuraavaksi esitellään ne asiat jotka tulee ottaa huomioon tehtäessä vertailua kyseisten kameramallien välillä.

<u>Laitevertailu</u>	ASL 504	Tobii 1750
Resoluutio	260 x 240 (pikseliä)	1280 x 1024 (pikseliä)
Analysointiohjelmisto	Eyenal	ClearView
Fiksaatioalgoritmi	Kts. kohta 5.4.3	Kts. kohta 5.4.3

Taulukko 7. ASL 504 ja Tobii 1750 ominaisuuksien vertailu.

Aluksi huomataan, että molemmilla kameroilla on hieman erilainen X- ja Y-koordinaatisto jossa katseenkulkua seurataan. Tästä syystä täytyy ensin tehdä suhteutus että molempien kameramallien arvoja voidaan pitää vertailukelpoisina. Tobii 1750:n näytön resoluutiona oli mittauksissa 1280 x 1024 pikseliä ja ASL 504:n näytön resoluutiona oli 260 x 240 pikseliä. Suhteutetaan ASL:n näytön resoluutio Tobiiin vastaavaan laskemalla $1280 / 260$ ja $1024 / 240$, joista saadaan suhdeluvuiksi X-asteikolle 4,923 ja Y-asteikolle 4,267. Nämä saadut luvut toimivat kertoimina kun vertaillaan fiksaatioiden osumapisteitä (koordinaatioston X- ja Y-arvoja) ASL:n ja Tobiiin välillä. Lisäksi joudumme myös vielä tarkastamaan mittauksista saaduista nauhoituksista ovatko ne synkronoituja keskenään. ETDPlayer paljasti, että Tobii nauhoitus (Kuva 17) oli 5 sekuntia 75 sekunnin kymmenesosaa liian aikaisessa vaiheessa, kun nauhoitukset oli synkronoitu keskenään.



Kuva 17. ETDPlayeristä nähdään suoraan kuinka paljon Tobii nauhoitus on jäljessä.

Silmänliikekameroiden analysointiohjelmisto eroaa toisistaan myös jonkin verran. Tämä huomataan dataa analysoitaessa. Onneksi käytössä on ETD-player videontoisto-ohjelma, johon voidaan tuoda samaan aikaan näkymään niin Tobiiista, kuin ASL:stä nauhoitettua dataa. Videokuva paljastaa, että ASL:n kalibrointi ei ole onnistunut niin hyvin kuin Tobiiin kalibrointi. Vaikka ASL pystyy käytännössä parempaan tarkkuuteen jää se nyt tarkkuudeltaan muutamia millejä kalibrointipisteistä.

Fiksaatioita suodatettaessa tulokseksi saatiin Eyenal - ohjelmalla, että ASL 504 rekisteröi mittauksen aikana 41 fiksaatiota. Tobii 1750 sen sijaan sai samalla nauhoituskerralla rekisteröityä 88 fiksaatiota.

5.5.1 Fiksaatioalgoritmi

Joudumme käyttämään omaa fiksaatioalgoritmia selvittäessämme fiksaatioiden ominaisuuksia raakadatasta, koska valmiit algoritmit näyttivät antavan liian paljon toisistaan poikkeavaa fiksaatiodataa kameramallien välille. Tämä ei voi kuitenkaan pitää täysin paikkaansa, sillä jo videokuva EDTPlayerissä kertoo, etteivät katseen attentiokohteet eroa merkittävästi toisistaan. On siis syytä käyttää omaa algoritmia fiksaatioiden suodatukseen. Olen valinnut erään yksinkertaisen algoritmin, jonka avulla pitäisi vertailuun saatava fiksaatioiden suodatus olla mahdollisimman yhdenmukainen Tobiiista ja ALS:stä saatujen raakadatojen kesken. Algoritmi sisältää tässä yhteydessä seuraavat ominaisuudet:

- Fiksaation alkuaika lasketaan siitä hetkestä, kun katsekohta pysyy puolen asteen (puoli astetta vastaa n. 16 pikseliä näytöllä) säteellä yli 10 ms.
- Fiksaation katsotaan jatkuvan niin kauan kuin katsekohta pysyy yhden asteen (yksi aste vastaa n. 32 pikseliä näytöllä) sisällä.
- Jos data saa miinusarvoja, luetaan ne katkoksiksi ja näitä ei lasketa mukaan.

Tätä algoritmia käytetään hyväksi raakadatan analysoinnissa. Aivan ensin raakadata muutetaan Excel-taulukkolaskentaohjelman hyväksymään muotoon, josta sitä on helppoa käsitellä. Sen jälkeen ASL:n raakadata skaalataan samalle asteikolle Tobii:sta saadun datan kanssa. Tämä tapahtuu suhteuttamalla ASL:n raakadatan H- ja V-position arvot skaalauskerroimilla (X-tasossa kerroin on n. 4,9231 ja Y-tasossa 4,2667).

Kun ASL:n raakadata on saatu skaalattua Tobii:n kanssa vertailukelpoiseksi, on aika ohjelmoida fiksaatioalgoritmi Visual Basic makrokseksi Excel taulukkoon. Näin sitä on helppo kutsua ja käyttää. Samoin sen saamia parametreja on nyt helppo muuttaa. Kun Exceliin ohjelmoitu Visual Basic makro ajettiin läpi, saatiin fiksaatioiden määräksi ALS:lle yksinohjelmointiasetuksilla 43 fiksaatiota ja pariohjelmoinnilla X fiksaatiota.

Tobii:n raakadatasta suodattui 78 fiksaatiota, eli vähän alle kaksinkertainen määrä verrattuna ASL:stä saatuun määrään.

Seuraavaksi oli aika tutkia tarkemmin fiksaatioiden laatua. Ensin laskettiin molemmista raakadatoista fiksaatioiden yhteiskesto ja sitten fiksaation keskimääräinen kesto. Ensin ASL – nauhoitus yksinohjelmointiasetuksilla. Fiksaatioalgoritmin ja Exceliin ohjelmoidun Visual Basic makron avulla saatiin seuraava, taulukon 8 mukainen tulos:

Nauhoitus ASL (yksinohjelmointiasetukset)	
Nauhoituksen kokonaiskesto	1 min. 8 sek.
Fiksaatioiden kokonaismäärä	53 kpl
Fiksaatioiden yhteenlaskettu kokonaiskesto	60,65 s.
Keskimääräinen fiksaation kesto	1,144 s. / fiksaatio

Taulukko 8. ASL (yksinohjelmointiasetukset) fiksaatiot

Ja sitten vielä sama ASL nauhoitukselle pariohjelmointiasetuksilla, jolloin saatiin taulukon 9 mukaisia tuloksia.

Nauhoitus ASL (pariohjelmointiasetukset)	
Nauhoituksen kokonaiskesto	1 min. 8 sek.
Fiksaatioiden kokonaismäärä	60 kpl
Fiksaatioiden yhteenlaskettu kokonaiskesto	60,44 s
Keskimääräinen fiksaation kesto	1,007 s. / fiksaatio

Taulukko 9. ASL (pariohjelmointiasetukset) fiksaatiot

Tobiilla nauhoitetusta raakadatasta saadaan samalla fiksaatioalgoritmilla selville seuraavaa:

Nauhoitus Tobii	
Nauhoituksen kokonaiskesto	1 min. 15 sek.
Fiksaatioiden kokonaismäärä	78 kpl
Fiksaatioiden yhteenlaskettu kokonaiskesto	21,42 s
Keskimääräinen fiksaation kesto	0,275 s / fiksaatio

Taulukko 10. Tobii fiksaatiot

Omaa fiksaatioalgoritmia käytettäessä havaitaan miten fiksaatioiden kokonaismäärä on nyt realistisempi eli vastaa paremmin EDTPlayerin videokuvaa.

5.6 Attention päällekkäisyys

Eräs olennainen vertailukohde silmänliikekamaroista saaduista raakadatoista on attention päällekkäisyys eli tieto siitä, kohdistuuko katse näytöllä samaan kohtaan tietyllä ajanjaksolla molempien silmänliikekameroiden raakadatassa. Tätä voidaan tarkastella suoraan silmänliikekameroiden raakadatoista ja toisaalta asia voidaan myös varmistaa videolta käynnistämällä EdtPlayerissä samaan aikaan Tobii:sta ja ASL:stä saatu nauhoite.

Alla kuvassa 18 nähdään miten vasemmalla olevan kalibrointipisteen päällä ovat sekä ASL:stä, että Tobii:sta saatu katseen kiinnittyminen tietyllä ajanhetkellä. Kuten jo kuvasta näemme, eivät katseen kohteet osu täysin toistensa päälle. Tämä johtuu useasta tekijästä, joista tärkeimmät ovat kalibroinnin tarkkuus, videoiden synkronointi (että videot alkavat varmasti samalla ajanhetkellä).



Kuva 18. Attention kohteet kalibrointipisteen kohdalla (Vasen piste Tobiin ylin ja ASL alin).

Raakadatoista nähdään katseen attention sijainti tietyllä ajanhetkellä. Sijainti ilmoitetaan pikseleinä (X- ja Y-koordinaatit). Tämän johdosta on kätevää käyttää Excel-taulukkolaskentaohjelmaa ja tuoda attentiokohteiden sijaintikoordinaatit molemmista raakadatoista vierekkäin Excel-taulukkoon. Näin tehtiin myös tässä tutkielmassa ja saatiin kuvan mukainen näkymä:

Microsoft Excel - Vertailu_ASL_ja_Tobii						
File Edit View Insert Format Tools Data Window Help						
A23 925,5428						
	A	B	C	D	E	F
1	ASL (Yksinohjelmointi)		ASL (Pariohjelmointi)		Tobii	
2						
3						
4						
5						
6						
7						
8						
9	SkaalattuH-pos	SkaalattuV-pos	SkaalattuH-pos	SkaalattuV-pos	GazepointX (L)	GazepointY (L)
86	1048,6203	1062,4083	920,6197	529,0708	933	511
87	945,2352	618,6715	915,6966	529,0708	936	480
88	940,3121	614,4048	900,9273	524,8041	950	480
89	935,389	601,6047	896,0042	524,8041	936	499
90	930,4659	584,5379	896,0042	524,8041	938	500
91	930,4659	567,4711	896,0042	524,8041	931	480
92	930,4659	550,4043	900,9273	524,8041	936	504
93	930,4659	546,1376	910,7735	529,0708	936	501
94	925,5428	546,1376	910,7735	524,8041	930	503
95	925,5428	546,1376	910,7735	524,8041	945	485
96	930,4659	541,8709	910,7735	524,8041	940	540
97	930,4659	546,1376	896,0042	520,5374	944	493
98	935,389	550,4043	900,9273	520,5374	952	496
99	925,5428	546,1376	900,9273	516,2707	937	490
100	930,4659	541,8709	900,9273	516,2707	934	503
101	935,389	537,6042	905,8504	520,5374	954	476
102	935,389	541,8709	910,7735	520,5374	942	491
103	940,3121	546,1376	905,8504	524,8041	944	494
104	935,389	546,1376	915,6966	529,0708	936	502
105	930,4659	546,1376	915,6966	524,8041	933	489
106	930,4659	541,8709	915,6966	524,8041	951	510
107	935,389	541,8709	915,6966	529,0708	919	521
108	935,389	541,8709	900,9273	524,8041	922	518
109	940,3121	541,8709	905,8504	520,5374	934	486
110	930,4659	541,8709	905,8504	520,5374	933	530

Kuva 19. Attention sijainti (X- ja Y-koordinaatistossa) raakadatojen mukaisesti

Kuvasta 19 nähdään vierekkäin ASL:n yksin- ja pariohjelmoinnista, sekä Tobii:sta saadut attention sijainnit. Eri raakadatoista saadut arvot on eroteltu erivärisille pohjille selkiyttämään näkymää. Kuten kuvasta X nähdään, on ASL arvot skaalattu (kohdassa 5.5 esitellysti) Tobii:n koordinaatiston mukaisesti. Kun arvot ovat siististi taulukossa, on niistä helppo laskea attention päällekkäisyys tietyllä ajanhetkellä.

Koska näytteenottoväli on 20 millisekuntia, on se myös järkevintä säilyttää aikavälinä attention sijainnin laskemisessa. Tämän jälkeen lasketaan koordinaattipisteiden sijainnin erotus 20 millisekunnin välein niin X- kuin Y-asteikossa ASL:n ja Tobiiin raakadatojen välillä. Tarkoitus on ensin laskea miten kaukana katseen attention sijainti on kullakin hetkellä toisistaan ja tämän jälkeen tutkia miten suuren osan videon kokonaisuudesta katseet ovat päällekkäin. Attentiokohteiden koordinaattien välisen etäisyyden laskemiseen on jälleen järkevintä käyttää Excelin tarjoamaa mahdollisuutta ohjelmoida Visual Basic makroja. Näin ollen varsinainen laskeminen automatisoituu makron ansiosta täydellisesti. Makroon ohjelmoidaan silmukka, joka käy läpi kaikki datarivit ja suorittaa algoritmin mukaisen laskennan. Ohjelmoitu makro laskee siis jokaisen attention-pisteen sijainnin välisen etäisyyden molemmilla kameramalleilla. Olen antanut kompensatiota 75 pikselin verran (joka vastaa hiukan yli kalibrointipisteen sädettä), joka tarkoittaa että jos katseen attentioniot ovat kameroissa saaduissa raakadatoissa tämän alueen sisällä, lasketaan attentionkohteiden silloin olevan täysin samat.

Edellä esitellyllä tavalla saatiin aikaan seuraavat tulokset:

- ASL (yksinohjelmoitiasetuksilla) ja Tobii
Videoiden synkronoitu kokonaiskesto 69,02 sekuntia, joista katseen attentionkohteet olivat päällekkäin 37,4 sekuntia, joka on prosentuaalisesti n. 54,2%.
- ASL (pariohjelmoitiasetuksilla) ja Tobii
Videoiden synkronoitu kokonaiskesto 68,82 sekuntia, joista katseen attentionkohteet olivat päällekkäin 48,6 sekuntia, joka on prosentuaalisesti n. 70,6%.

Tarkastellaan vielä attention päällekkäisyyttä vaihteluvälillä 50 – 100 pikseliä viiden pikselin välein. Itse vaihteluväli 50-100 pikseliä vastaa 1 – 2 kalibrointipisteen sädettä (eli 1,47 – 2,94 senttimetriä).

ASL vs. Tobii päällekkäiset attention kohteet (molemmissa tapauksissa Tobiiin raakadata vastaan)		
Tarkasteluväli (pikseleinä)	Päällekkäisiä attentiokohteita ASL (yksinohjelmointiasetuksilla)	Päällekkäisiä attentiokohteita ASL (pariohjelmointiasetuksilla)
50	41,44 %	54,93 %
55	45,06 %	59,58 %
60	47,81 %	63,21 %
65	50,28 %	66,41 %
70	52,30 %	69,02 %
75	54,19 %	70,62 %
80	56,07 %	72,65 %
85	57,52 %	73,53 %
90	58,68 %	74,25 %
95	59,69 %	74,98 %
100	60,27 %	75,56 %

Taulukko 11. Päällekkäiset attentiokohteet raakadatas

Taulukosta 11 havaitaan kuinka pariohjelmointiasetuksilla on saatu aikaan enemmän päällekkäisiä attention kohteita kuin yksinohjelmointiasetuksilla.

5.7 Testituloksiin vaikuttavat tekijät

Tarkastellaan vielä luvun lopuksi joitakin testituloksiin vaikuttavia tekijöitä. Lähinnä sellaisia seikkoja jotka voivat osaltaan vaikuttaa testitulosten oikeellisuuteen. Ensiksikin eräs merkittävä tekijä tulosten luotettavuudessa on kalibroinnin onnistuminen. Nyt

suoritetuissa testeissä kalibroinnissa pyrittiin mahdollisimman hyvään tarkkuuteen. Aina on kuitenkin mahdollista että kalibroinnin tarkkuus voi olla näytön eri pisteiden kohdalla hieman vaihtelevaa. Toisin sanoen tietyissä pisteissä kalibrointi voi olla tarkempaa kuin toisissa. Tämä johtuu usein siitä, että kun kalibroinnin alkuasetuksia tehdään, pidetään silmä (ja samalla koko pää) paikoillaan niin, että silmä näkyy keskellä apumonitorin näyttöä. Myöhemmin, varsinaisen testin aikana, saattaa testihenkilö joutua vahingossa liikuttamaan myös päätään pelkän katseen sijaan. Näin voi syntyä hienoista epätarkkuutta, joka pääsee monitoroinnista läpi. Pahimmillaan tämä voi katkaista hetkellisesti datan saannin, mikäli silmä siirtyy kameran kantaman ulkopuolelle. Tällöin raakadataan tulee attention kohteiden sijaintiin yleensä miinusmerkkisiä arvoja.

Toinen mittaustuloksiin vaikuttava tekijä on pyöritykset, joita on jouduttu tekemään kun kameramallien näytön resoluutioiden suhteutus on suoritettu. Samalla myös Tobii - silmänliikekameran nauhoituksesta syntynyt video on synkronoitu ASL:n kanssa samalla viivalle. Tämäkin voi aiheuttaa hienoista vääristymää raakadatan vertailuun.

Kolmas testituloksiin vaikuttava, ja varsin merkittävä tekijä on käytetty fiksaatioalgoritmi. Molemmista silmänliikekameroista saadusta raakadatasta suodatettiin fiksaatioalgoritmin avulla fiksaatiot algoritmin täyttämisen ehdoin. Ensin käytin silmänliikekameroiden mukana tulevia omia ohjelmistoja ja niiden tarjoamia algoritmeja (kts. kohta 5.4.3 ClearView ja Eyenal – fiksaatioalgoritmit) mutta koska algoritmit antoivat niin paljon toisistaan poikkeavaa fiksaatiodataa, siirryin käyttämään omaa algoritmia.

Koska algoritmi on sama kaikille raakadatoille, ei sen pitäisi vääristää vertailtavuutta mutta sen sijaan on otettava huomioon että kameroiden tarkkuuksissa on eroja. Tarkkuuden vaihtelut vaikuttavat myös fiksaatioalgoritmin suodatukseen. Jos toisen kameran tarkkuus on suurempi, voidaan siitä johtuen saada fiksaatioiden määräksi saada isompi tai pienempi arvo kuin toisella kameralla.

Lopuksi mainittakoon vielä erääksi tuloksiin vaikuttavaksi tekijäksi katsepolkujen päällekkäisyyttä tarkasteltaessa annettu tarkasteluväli (eli 75 pikseliä). Tämä arvo toimii

kompensaationa ja jos molempien silmänliikekameroiden raakadatoissa attention kohteen sijaintikoordinaatit ovat samalla ajanhetkellä tämän alueen sisällä, lasketaan attention kohteen olevan sama. Jos tarkasteluväliä olisi suurennettu, oltaisi saatu lisää päällekkäisyyttä attention kohteille (tämä todetaan kohdassa 5.6 attention päällekkäisyys) ja vastaavasti tarkasteluväliä pienentämällä attentio kohteiden päällekkäisyys olisi suhteessa pienentynyt. Tässä yhteydessä käytetty 75 pikselin tarkasteluväli on arvioitu videokuvasta, jossa se vastaa vähän yli kalibrointipisteen sädettä.

6 YHTEENVETO

Suoritetaan lopuksi yhteenveto suoritetuista mittauksista ja niissä saaduista tuloksista. Tuloksiin vaikuttavia tekijöitä on olemassa jonkin verran ja ne täytyy ottaa huomioon tuloksia tarkisteltaessa. Oman fiksaatioalgoritmin käyttö on välttämätöntä, koska siten voidaan fiksaatioita suodattaa helpommin halutuilla parametreilla ja datasta saadaan helpommin vertailukelpoisempaa. Kun algoritmi on kerran ohjelmoitu esimerkiksi Visual Basic - makrosi, on sitä jälkeempään helppo muokata vastaamaan omia tarpeita. Samoin makroja voidaan ajaa useilla eri parametreilla.

ClearView rekisteröi useampia lyhytkestoisia fiksaatioita, kun taas Eyenal rekisteröi fiksaatioiksi pitempikestoisia kohdistumia. Erikoisinta oli että Tobiista saatiin fiksaatioiden kokonaiskestoksi huomattavasti lyhyempi aika kuin ASL:stä. Tämä näytti johtuvan pitkälti siitä, että Tobii rekisteröi enemmän lyhytkestoisia, (alle 150 millisekuntia) mutta ei fiksaatioksi luokiteltavia katseen kohdistumisia. Tähän vaikutti osaltaan käytetty fiksaatioalgoritmi mutta vaikka fiksaation kestoja pienennettiin 150:stä millisekunnista 120:een, ei tulos muuttunut merkittävästi. Eräs tapa jolla fiksaatioiden määrää olisi Tobiissa saatu kasvatettua realistisesti, olisi ollut lyhytkestoisten ja melkein fiksaatioita muistuttavien katseen keskittymien yhdistäminen, mikäli välissä olisi ollut vain lyhytkestoinen katko. Toisaalta silloin olisi jouduttu käyttämään samaa algoritmia myös ASL:n raakadatan suodatuksessa ja tulos olisi todennäköisesti ollut suhteessa samansuuntainen nykyisen kanssa.

Katseen päällekkäisiä attention kohteita laskettaessa päädyttiin videokuvaa vastaaviin tuloksiin. Päällekkäisyyksiä oli oltava yli 50 prosenttia, mielellään enemmän, koska tätä määrää myös videokuva omalta osaltaan tuki. Tutkimus osoitti että jos tarkasteluväli oli 75 pikseliä eli kaksi senttimetriä, oli myös päällekkäisten attention kohteiden määrä yli 50 prosenttia. Tarkasteluvälinä 75 pikseliä vastaan noin 1,5 kertaa kalibrointipisteen kokoa.

VIITELUETTELO

ASL (2001) *Eyenal (Eye-Analysis) Software Manual Windows version version 1.3a*. Applied Science Group, Inc.

ASL (2003) *Eye Tracking System Instruction Manual – Model 504 Pan/Tilt Optics*. Applied Science Laboratories.

Ditchburn, R. W. (1973) *Eye-movements and visual perception*, Glarendon Press, Oxford.

Duchowski, A.T. (2003) *Eye tracking methodology : Theory and practice*, Springer, cop. Lontoo.

Garcia-Moreno, F. T. (2001) *Installation and setup of ASL eye tracking system model 504 with Pan/Tilt Optics*, Queen's University of Belfast, Belfast.

Goldberg, J.H. Wichansky, A.M. (2003) Eye tracking in usability evaluation: A practitioner's guide. In Hyönä, J., Radach, R. & Deubel, H. (Eds.) *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*. Amsterdam: Elsevier Science, 493–516.

Granka, L. Joachims, T. Gay, G. (2004) *Eye-Tracking Analysis of User Behavior in WWW search*. ACM, New York, USA.

Jacob, Robert J.K. (1993) *Eye-gaze Computer Interfaces: What You Look At is What You Get*, IEEE Computer, vol. 26, no. 7, 65-67 (July 1993).

Jacob, Robert J.K. (1998) *The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get*, ACM Transactions on Information Systems, vol. 9, no. 3, pp. 152-169 (April 1991)

Hulkko, H. Abrahamsson, P. (2005) *A Multiple Case Study on the Impact of Pair Programming on Product Quality* International Conference on Software engineering St. Louis, MO, USA May 15-21 2005, 495-504.

Hyönä, J. (1993) *Eye Movements During Reading And Discourse processing*, Psychological research reports / University of Turku 65, Painosalama Oy, Turku.

Johansen, S.A., Hansen, J.P. (2006) Do we need eye trackers to tell where people look?, *Conference on a Human Factors in Computing Systems* (toim.), ACM, 923-928.

Kanski, J.J. (2003) *Clinical Ophthalmology: A Systematic Approach, fifth edition*. Butterworth-Heinemann, Boston.

Lehtinen, M. (2005) Katseenseuranta. Ovaska, S., Aula, A. & Majaranta, P. (toim.) *Käytettävyytutkimuksen menetelmät*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos B-2005-1, 223-236.

Lui, K.M., Chan, K.C.C. (2006) *Pair programming productivity : Novice-novice vs. expert-expert*. *Int. J. Human-computer Studies* 64 (2006), 915-925.

Martinez-Conde, S., Macknik, S.L., Hubel, D. H. (2004) *The Role Of Fixational Eye Movements in Visual Perceptio*. *Nature Reviews Neuroscience* 5(3), 229-240.

McDowell, C. Werner, L. Bullock, H.E. Fernald, J. (2006) *Pair programming improves student retention, confidence, and program quality* Communications of the ACM August 2006/vol. 49, No.8., 90-95.

Muller, M.M. (2006) *A preliminary study on the impact of a pair design phase on pair programming and solo programming*. Information and Software Technology 48, 335-344.

Neisser, U. (1982) *Kognitio ja todellisuus*. Weiling + Göö, Espoo.

Nevalainen, S., Sajaniemi J. (2004) *Comparasion of three eye tracking devices in Psychology of programming Research*, 151-158.

Oyekoya, O.K., Stentiford, F.W.M. (2006) *Eye tracking – a new interface for visual exploration*, BT Technology Journal, vol. 24, no. 3. (July 2006)

Roorda, A. (2002) Human Visual System - Image Formation, *Encyclopedia of Imaging Science and Technology*, **1**, 539-557.

Špakov, O. (2008) *iComponent – Device-Independent Platform for Analysing Eye Movement Data and Developing Eye-Based Applications* Väitöskirja Tietojenkäsittelytieteenlaitos, Tampere.

Succi, G. Marchese, M (2001) *Extreme Programming Examined*. Addison Wesley, Boston, 221-320.

Tobii Technology AB. (2003) *Product Description Tobii 1750 Eye-tracker Release B*. Release date Nov 2003, Tobii Technology AB.

Tobii Technology AB. (2006) *User Manual : Tobii Eye Tracker ClearView analysis software*. Tobii Technology AB.

Wikipedia, the free encyclopedia (2007a) *Wikipedia, the free encyclopedia*.
<http://en.wikipedia.org/wiki/Image:ReadingFixationsSaccades.jpg>, (15.5.2008)

Wikipedia, the free encyclopedia (2007b) *Wikipedia, the free encyclopedia*.

http://en.wikipedia.org/wiki/Image:Schematic_diagram_of_the_human_eye_en.svg,
(15.5.2008)

Williams L, Kessler, R.R., Cunningham, W., Jeffries, K. (2000) *Strengthening the case for pair programming*. IEEE Software July/August 2000.

Wolfe J. (2000) *Visual attention*. In: De Valois KK, editor. Seeing. 2nd ed. San Diego, CA:Academic Press; 2000, 335-386.

Yarbus, A. L. (1967) *Eye Movements and Vision*. (Translated from Russian by Basil Haigh. Original Russian edition published in Moscow in 1965.). Plenum Press, New York, 1967.

Zhu, Z., Qiang, L. (2004) Eye and gaze tracking for interactive graphic display, *Machine Vision and Applications*, **15**(3), 139-148.